

RELAY LOCATION IN TELECOMMUNICATIONS AND TRANSPORTATION NETWORKS

A DISSERTATION SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
INDUSTRIAL ENGINEERING

By

Barış Yıldız

09 March 2016

Relay Location in Telecommunications and Transportation Networks

By Barış Yıldız

09 March 2016

We certify that we have read this dissertation and that in our opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Oya Kardeş(Advisor)

Selim Aktürk

Sinan Gürel

Hande Yaman Paternotte

Haldun Sürat

Approved for the Graduate School of Engineering and Science:

Levent Onural
Director of the Graduate School

ABSTRACT

RELAY LOCATION IN TELECOMMUNICATIONS AND TRANSPORTATION NETWORKS

Barış Yıldız

Ph.D. in Industrial Engineering

Advisor: Oya Karaşan

09 March 2016

With different names and functions, relays play a crucial role in the design of telecommunications and transportation networks and finding optimal relay locations is an important concern in various applications. We investigate several relay location problems from the literature, propose new ones and design efficient solution methods to obtain managerial insights. The basic problem we investigate in this dissertation is the Regenerator Location Problem (RLP). We revisit RLP from the hub location perspective and introduce two new dimensions involving the challenges of survivability. Considering the flexible optical network architecture, we relax all pairs connectivity, infinite capacity links and single modulation level assumptions of RLP and introduce the regenerator location problem in flexible optical networks (RLP-FON). RLP-FON solves regenerator location, routing, bandwidth allocation and modulation selection problems jointly to better exploit the opportunities offered by this novel network architecture. For various problems arising in telecommunications and transportation it is beneficial to consider edge design and relay locations together. We add the edge design aspect to RLP and extend our research to Network Design Problem with Relays. Different than telecommunications networks, the total length of a route is an important issue in transportation. So in the final part we include circuitry constraints to the routing decisions and study the Refueling Station Location Problem for Alternative Fuel Vehicles.

We approach relay location problems from different angles: network topologies, capacities, costs, and demands and provide significant theoretical results. For all relay location problems, the reach limitations for the related entities pose the main challenge and we propose novel path-segment based formulation approaches to incorporate these constraints in an efficient way. Extensive numerical experiments with realistic problem instances attest to the efficacy of the proposed approach. *Keywords:* Relay, Regenerator, Refueling Station, Network Design, Location, Integer Programming, Branch-and-Cut, Branch-and-Price.

ÖZET

HABERLEŞME VE ULAŞIM AĞLARINDA RÖLE YERLEŞTİRME PROBLEMLERİ

Barış Yıldız
Endüstri Mühendisliği, Doktora
Tez Danışmanı: Oya Karaşan
09 Mart 2016

Haberleşme ağlarında optik sinyallerin menzillerini artıran yenileyiciler, ulaşım ağlarında yüklerin, sürücülerin yada ulaşıma vasıtalarının değiştirildiği stratejik olarak seçilmiş noktalar, alternatif yakıt ile çalışan araçların uzun mesafeli yolculuklarını tamamlayabilmeleri için yol ağı üzerine yerleştirilen dolum istasyonları rölelerin değişik uygulama alanlarındaki kullanımlarına birer örnek teşkil etmektedir. Röleler değişik isimler ve fonksiyonlarla birçok haberleşme ve ulaşım ağının temel birer parçasıdır ve rölelerin ağlarda en iyi yerlerinin bulunması birçok uygulama alanında önemli bir problem olarak ortaya çıkmaktadır. Bu çalışmada, yazında yer alan değişik röle yerleştirme problemleri ele alınmış, yeni problemler tanıtılmış ve bu problemlerin çözümü için etkin kesin çözüm yöntemleri önerilmiştir. Röle yerleştirme problemlerine değişik ağ topolojileri, kapasiteleri, maliyetleri ve talep durumları perspektiflerinden yaklaşılarak önemli teorik sonuçlar elde edilmiştir.

Röle yerleştirme problemlerini temel olarak diğer problemlerden ayıran ve çözümünü zorlaştıran ana husus ele alınan varlıkların ağdaki hareketini sınırlayan menzil kısıtlarıdır. Önerdiğimiz yol-parçacığı tabanlı formülasyonlar ile söz konusu menzil kısıtlarının ve oluşturulacak rotalar için ön görülecek daha genel sınırlandırmaların doğrudan ve etkin bir şekilde modellenmesi mümkün olmaktadır. Yol-parçacığı tabanlı formülasyonların esnek yapıları kullanılarak, değişik uygulama alanlarında ortaya çıkan ve bir birinden farklı özellikleri barındıran röle yerleştirme problemleri için etkin çözüm yöntemleri sunulmuştur. Gerçekçi problem örnekleri üzerinde yapılan kapsamlı hesapsal çalışmalar, önerdiğimiz bu özgün yaklaşım ile geliştirdiğimiz kesin çözüm yöntemlerinin hesapsal performanslarının literatürde var olan en iyi sezgisel algoritmalar ile yarışabilecek seviyelerde olduğunu göstermektedir.

Anahtar sözcükler: Röle, Yenileyici, Dolum İstasyonu, Ağ Tasarımı, Yer Seçimi, Tamsayılı Programlama, Dal Kesi, Dal Fiyat.

Acknowledgement

I would like to express my deepest gratitude to those people and institutions who made this dissertation possible with their invaluable contributions.

Foremost, I wish to thank to my advisor Prof.Dr. Oya Karasan. These pages are essentially a brief summary of her unwavering academic rigor, vast patience and never ending support. She is the one who turned this arduous task to a wonderful learning experience that I will cherish to the end of my life. There is not enough space here to list fortunes of having her as an advisor but there is certainly one serious disadvantage. Her exceptional example puts the bar so high that, it is always disappointing to compare yourself to those unreachable standards. Nevertheless, having a north-star is always good to find your way when you get lost even if you know you can never reach to it.

Besides my advisor, I would like to thank my thesis committee members: Prof. Dr. Hande Yaman and Prof. Dr. Haldun Sural for their commitment to ensuring that this thesis pushed the boundaries of academic research.

I want to express special thanks to my wife Özlem, for her unconditional love and endless support, on which I stood in comfort to complete this long and challenging journey. She did not only have to take care of the family business when I got lost under the pile of papers, she also encouraged me to take one more step forward when I felt like I will not be able to. I am also obligated to express my warmest thanks to my kids Elif and Onur Alp, who wiped away all the worries and the exhaustion from the long working hours with the warmest little hugs in the world. Deleting useless files from my computer, putting some color and shape on my pale research papers and teaching me how to type with four hands on the keyboard, they put some fun to this sometimes monotonous academic endeavor.

I will always feel indebted to Turkish Armed Forces for supporting all my educational ventures starting with the high school graduation to end with a Ph.D. degree.

Last but not least, I want to express my thanks to TUBITAK for supporting this research with grants 214M211 and 2211-A.

Contents

1	Introduction	1
1.1	Regenerator Location Problem	2
1.2	Hub Location Perspective and Transportation Applications	4
1.3	Regenerator Location Problem in Flexible Optical Networks	6
1.4	Network Design Problem with Relays	9
1.5	Refueling Station Location Problem	12
1.6	Conspectus	14
2	Nomenclature	16
3	Regenerator Location Problem	19
3.1	Mathematical Models	23
3.1.1	Definitions and Notation	23
3.1.2	Problem Complexity	23
3.1.3	Regenerator Location Problem: Compact Formulation	25
3.1.4	Regenerator Location Problem: Cut Formulation	28
3.1.5	Regenerator Location Problem under Regenerator Failures: Compact Formulation	29
3.1.6	Regenerator Location Problem under Regenerator Failures: Cut Formulation	33
3.1.7	Regenerator Location Problem under Node Failures: Com- pact Formulation	34
3.1.8	Regenerator Location Problem under Node Failures: Cut Formulation	35
3.2	Path Recovery	36

3.3	Separation Algorithms	40
3.3.1	Separation Algorithm for the RLP	40
3.3.2	Connectivity Separation (CS)	41
3.3.3	Global Minimum Cut Separation (GMCS)	41
3.3.4	Minimum Cut Separation (MCS)	42
3.3.5	Separation Algorithm for the RLPRF	43
3.3.6	Two-Connectivity Separation (TCS)	43
3.3.7	Connectivity Separation (CS_q)	45
3.3.8	Global Minimum Cut Separation ($GMCS_q$)	45
3.3.9	Minimum Cut Separation (MCS_q)	45
3.3.10	Separation Algorithm for the RLPNF	46
3.3.11	Branch-and-cut Implementation Details	46
3.4	Numerical Experiments	48
3.4.1	Performances of the Models and Algorithms	48
3.4.2	Spatial Insights	52
3.5	Summary	60
4	Regenerator Location Problem in Flexible Optical Networks	61
4.1	Mathematical Model	63
4.1.1	Definitions and Notation	63
4.1.2	RLP-FON Path-Segment Formulation (PS)	66
4.1.3	Solution Approach:	68
4.1.4	Heuristic Solutions	74
4.2	Insights to Problem Complexity	75
4.2.1	Uncapacitated Edges	79
4.2.2	Computational Performance of the Branch-and-Price Algorithm for Special Problem Instances	85
4.3	Numerical Experiments	89
4.3.1	Network and Traffic	89
4.3.2	Fiber and Modulation Level Parameters	91
4.3.3	Implementation Details	93
4.3.4	Performance of the Branch-and-Price Algorithm	94
4.3.5	Managerial Insights	100

4.4	Summary	103
5	Network Design Problem with Relays	105
5.1	Definitions and Notation	108
5.2	Solving NDR	110
5.2.1	Properties of NDR Solutions	110
5.2.2	Multi-commodity Flow Formulation (MCF)	111
5.2.3	Solving MCF	113
5.3	Solving NDR-S	117
5.3.1	Properties of NDR-S Solutions	117
5.3.2	Tree Formulation (TF)	119
5.3.3	Strengthening TF	121
5.3.4	Solving STF	123
5.3.5	Enhancing MCF for NDR-S problems	127
5.4	Numerical Experiments	127
5.4.1	Grid Networks	128
5.4.2	Steiner Problems from OR-Library	133
5.5	Summary	135
6	Refueling Station Location Problem for Alternative Fuel Vehicles	137
6.1	Mathematical Model	141
6.1.1	Definitions and Notation	141
6.1.2	Path-Segment Formulation for RSLP-R (<i>PSF</i>)	143
6.1.3	LP Solution	145
6.1.4	IP solution	147
6.2	Numerical Experiments	149
6.2.1	AZ Network	149
6.2.2	CA Network	154
6.3	Summary	156
7	Conclusion	158
7.1	Remarks	158
7.2	Future Study	159

List of Figures

1.1	Relay location problems studied in this dissertation	14
3.1	Non-simple paths	24
3.2	Working with G_q^c rather than G^c	30
3.3	32-Node Network	53
3.4	32-Node Network: Number of Regenerators versus d_{max}	54
3.5	Hub Locations (marked nodes) for $d_{max} = 1090$	55
4.1	A simple example depicting the fact that initial columns matter	72
4.2	Depiction of a small transformation example with three knapsack items 1, 2 and 3.	77
4.3	Depiction of a small transformation example for $S = \{3, 9, 12\}$.	79
4.4	Network topologies; link lengths in km. [55]	90
5.1	Examples of non-simple paths in directed and undirected graphs with threshold set to 9	106
5.2	Virtual network example	109
5.3	Pricing graph corresponding to the detection of negative reduced cost variables. The artificial arcs are depicted with dashed lines.	115
5.4	Grid graph solution illustrations	130
6.1	Non-simple path example	140
6.2	Network instances	150
6.3	Number of paths generated vs tolerance Level	155

List of Tables

3.1	Cut Formulation Results	51
3.2	Flow vs Cut formulations	52
3.3	Computational Results for the 32-Node Network Example	56
3.4	Analysis of Alternative Solutions	57
3.5	Computational Results For the Euclidean Networks	58
3.6	Euclidean Network Statistics	60
4.1	Topological parameters	91
4.2	Modulation level parameters	93
4.3	Results for COST-266 network with uniform demand distribution	96
4.4	Results for COST-266 network with traffic density demand distribution	97
4.5	Results for NSF network with uniform demand distribution	98
4.6	Results for NSF network with traffic density demand distribution	99
4.7	Exact solution results for COST-266 and NSF networks	101
4.8	Number of required regenerators for COST-266 network with traffic density demand distribution.	101
4.9	Percentage of nodes appearing as regeneration points in the optimal solutions for the problem instances with COST-266 network and traffic density demand distribution.	102
5.1	Solution time comparisons for the proposed algorithms	132
5.2	Results summary for STF solutions on larger size problem instances	133
5.3	TSF Algorithm results for OR library Steiner tree instances	135
6.1	Network and related O-D pair parameters	149

6.2	Solution time comparisons of DFRLM and B&P algorithms.	151
6.3	B&P solutions for the AZ network	152
6.4	B&P Solutions for the CA network	157



Chapter 1

Introduction

Relays are regenerators extending the reach of optical signals in telecommunications networks; they may be strategic locations where exchange of drivers, trucks or mode of transportation takes place in transportation networks; they may become refueling/recharging stations extending the reach of alternative fuel vehicles in green transportation. With different names and properties, relays are significant parts of several telecommunications and transportation network architectures.

The main functions of the relays are to extend the reach of some entities traveling in a network. In telecommunications networks, signal quality degrades with the distance and relays function as repeaters, regenerators, amplifiers, etc. to enhance the reach of the signal [152]. For example in an optical network, the reach of an optical signal is the maximum distance it can traverse before its quality degrades and regenerators are deployed on the network to restore the signal quality on its route and hence extend the optical reach. As an example from transportation, alternative fuel vehicles with limited ranges use refueling/recharging stations located on the road networks as relays to extend their reach and complete long distance trips.

In this dissertation we consider different practical settings and problems for

which the main concern is to find optimal relay placements. We investigate several relay location problems from the literature, propose new ones and design efficient solution methods to obtain managerial insights. Employing a hub location perspective we propose a new approach to solve relay location problems and build a unifying framework to relate relay location problems studied independently in the literature.

1.1 Regenerator Location Problem

One basic relay location problem from telecommunication is the regenerator location problem (RLP) [27, 152]. The practical motivation for this problem originates from the signal degradation problem in the optical networks. Despite their immense capacity to transmit digital data, fiber optic networks suffer from transmission impairments that limit transmission ranges. Optical signals cannot travel more than some given distance before their quality degrades below some threshold level. One key technology to extend optical reach and overcome this issue is opto-electro-optical (OEO) regeneration. Processed by an OEO regenerator, the optical signal is rejuvenated and after this renewal it can travel up to its optical reach before it arrives to a new regenerator or its final destination. So, with the expense of more capital investment and operating cost (such as energy and maintenance), it is possible to enhance the optical reach of a signal by employing regenerator equipments. As such, regenerators are essential for the optical networks and there is a rich literature that investigates regenerator locations from different technical and managerial perspectives [27, 28, 58, 67, 150, 152–154]. Because regeneration costs make up a significant portion of a network's setup and operating costs [150], there is great motivation to design optical networks with few regeneration points. With this practical motivation RLP seeks to place the minimum number of regenerators in an optical network so as to facilitate the communication of a signal between any node pair.

Although RLP is mostly known as a telecommunications problem, it is first introduced with a quite different application in mind. Campbell [19] introduces

and formulates a variant of the hub covering location problem under an interesting coverage criterion, namely, that demand nodes are considered covered when served via “close” hubs mutually interconnected in a complete fashion of “close” diameter. Although this specific covering criterion did not receive much attention in the succeeding hub location literature, it is of close kinship to the RLP. Rejuvenating its hub location roots, we approach the RLP from the hub location perspective in Chapter 3 and develop quite efficient solution algorithms to solve it with the help of this new perspective.

Although they are less costly, sparse networks in general and telecommunications networks in particular are vulnerable to damage and equipment failure. Therefore, survivability is also a big concern for optical networks designers [41, 66, 100]. Although the literature on survivable network design and the RLP is extensive, few studies consider the network survivability and regenerator placements together. We address the network survivability issue by introducing two new dimensions to RLP: First, we examine the survivability of a network that hedges against a single failure in the regeneration equipment. In this case, we assume that only nodes with regeneration capabilities (hubs) are susceptible to failure or that the high setup and operating costs of regeneration equipment prohibit employing redundancy strategies as freely as in the case of nodes with no regenerators (non-hub nodes). We thus define the RLP with resilience against regenerator failures (RLPRF) as the problem of finding the minimum number of regenerators (and their locations) that can facilitate transportation of the commodity between any two nodes even if an arbitrary regeneration point fails. Second, we extend the previous survivability notion to all nodes in the network. We define the RLP with resilience against node failures (RLPNF) as the problem of finding the minimum number of regenerators (and their locations) that can facilitate transportation of the commodity between any two nodes even if any node (hub or not) in the network fails. Regarding the RLP, our contributions are twofold.

1. Incorporating the hub location perspective to the problem, we relate RLP to various transportation settings and introduce the problems RLPRF and

RLPNF to the literature.

2. We study all three versions of the RLP under a unifying solution methodology, where for each problem we provide a compact flow formulation and a cut formulation suitable for efficiently utilizing a branch-and-cut procedure. Indeed, the performances of our exact algorithms challenge the best heuristic algorithms available in the literature.

1.2 Hub Location Perspective and Transportation Applications

As a significant gap in the literature, the relation between the relay and hub location problems is not well established. In our dissertation, we draw attention to this relation and employ a hub location perspective to develop efficient solution methods for the relay location problems. Similarly, the transportation applications of relay location problems are not as common as their telecommunications counterparts. In their insightful analysis of the last twenty-five years of hub location research, Campbell and O’Kelly [21] discuss the differences of transportation and telecommunications networks. These entities though seemingly similar in the abstract level of mathematical models could be quite different in their operations, service measures, costs and constraints due to the inherent differences of the transported objects. The lack of costs in pure service-oriented hub center and hub covering models have found more reasonable applications in telecom than in transportation. Confirming this analysis of Campbell and O’Kelly [21], most of the literature on relay location problems is confined to the field of telecommunications [17, 67, 85]. However, there are various transportation applications for relay location problems. Relays with their potential of extending the transport range via rejuvenation mainly adopt the connectivity feature (in the terminology of Campbell and O’Kelly [21]) as hubs.

Relay points in transportation networks are physical locations where various activities like the exchange of drivers, trucks and trailers can take place. Drivers

can rest or switch transportation means or simply leave the consignment at relay points to be picked up by other drivers. The relay network design problem in freight transportation systems as studied by Ali et al. [3] is almost identical with RLP. The hub-and-spoke aspect of relay networks has originated with the work of Taha and Taylor [131] and has been adopted as an alternative dispatching method for TL transportation in order to alleviate the common driver retention problem in several studies [3, 77, 132, 136, 137, 139]. Though relay networks and hub-and-spoke networks (not necessarily complete hub networks) are very similar, there are operational limitations such as circuitry, distances between relay points, load balance at relay points, and number of handlings from origin to destination that are specific to TL transportation in relay networks. Incorporating some of these limitations within an exact solution methodology for relay network designs has been the challenge in works of Üster and Kewcharoenwong [136], Üster and Maheshwari [137] and Vergara and Root [139] who manage to achieve optimal or near-optimal results for moderate sized problem instances.

Optimal deployment of combat logistics units in a military campaign and air-to-air refueling for unmanned aerial vehicles (or refueling for unmanned underwater vehicles) are just a few other examples where relay location problems can arise in transportation settings. In these applications, the objective is to facilitate the constant movement of combat units or sophisticated surveillance equipment that needs some kind of replenishment to continue functioning, while keeping the number of logistics support units at a minimum. Due to the either tactical or physical restrictions, the set of candidate locations to place such logistic units is usually a finite one which sets a context very similar to the one assumed by the relay location problems in the literature. In the presence of an intelligent adversary who actively targets the logistics network, survivability is even more important, and the need to extend relay location problems by including this aspect is more apparent.

A futuristic transportation application for relay location problems could be prime air delivery settings utilizing unmanned aerial vehicles like the drones of Amazon [109]. Drones have limited reach capabilities beyond which they have to be recalibrated/recharged. One can envision a hub-and-spoke network with a

covering aspect when designing a network including depots, recharging locations of drones and customers.

1.3 Regenerator Location Problem in Flexible Optical Networks

In its basic form, RLP considers all pairs communication through single modulation transmissions on links with infinite data carrying capacity. Although these assumptions are not restrictive for most applications, exponential growth in Internet traffic and recent advances in the fiber optic technologies introduces new problems in which they have to be relaxed. Wider availability Internet access, introduction of the mobile communication devices (smart phones, tablets, etc.) and booming sector of mobile applications have taken Internet age to a new stage [1]. In 2011, global mobile data traffic was eight times the size of the whole Internet in 2000, and is expected to increase 18-fold by 2016 [57]. As the growth of Internet surpasses even the highest estimates, utilized bandwidth of the optical fibers rapidly approaches its theoretical limits [39, 133]. Just worsening the problem, the rigid nature of the current optical networks cannot efficiently use available optical bandwidth to support this increasing traffic. The energy consumption of telecommunications networks is also adversely affected by wasteful resource utilization. Such inefficiencies unnecessarily increase the number of required active network equipments ultimately increasing the total energy consumption of the network. This is an issue of increasing importance since the power consumption of Internet is estimated to reach to the 10% of the worldwide energy consumption in a very short notice [114]. In the US alone, a 1% saving in the energy consumption of Internet due to the adoption of energy efficient network management strategies is estimated to result in savings of \$5 billion per year given that the price of electricity is 17 cents per kWh [126]. The concern over climate change and heavy carbon footprint of energy generation only increases the importance of energy efficiency of telecommunications networks.

Motivated by this urgent practical problem, researchers developed the flexible optical network (FON) architecture that can flexibly choose its transmission parameters according to the varying traffic conditions and significantly increase the resource utilization efficiency [39, 133]. The major sources of these inefficiencies, remedies offered by FON architecture and the algorithmic challenges raised by the adoption of this novel technology can be summarized as follows.

In the current optical network architecture, the available bandwidth is divided into a set of fixed bandwidth channels each serving a single transmission demand. However, due to the increasing variability of the offered online services, the capacity demands of connections come from a much broader range with granularities of several gigabits per second to 100 Gb/s or more. Due to the granularity mismatch between the widths of these channels and demand sizes, the already drained fiber bandwidth cannot be fully utilized [58]. On the other hand, with FON, the optical spectrum is divided into fine bandwidths called *slots* and custom-size bandwidths are created by the contiguous concatenation of those slots. Such custom-size transmission channels can significantly reduce the bandwidth waste and increase the amount of available fiber bandwidth.

The data transfer capacity of backbone networks is not solely dependent on the range of the available bandwidth. Indeed, this capacity is jointly determined by the available bandwidth range of the fiber and the modulation level that induces the amount of data that could be transferred on a fixed bandwidth. Modulation levels with higher bit rates can carry more data on a given bandwidth. But the down side of using higher bit rate modulations is the shorter optical reach. As a significant source of inefficiency, current optical networks use fixed modulation levels and waste bandwidth by using the same modulation level for both the short and long distance transmissions [59]. FON has been designed to increase the data transmission capacity of optical networks by smartly managing routing and modulation level selection in coordination and in particular, by utilizing high bit rate modulation schemas to increase bandwidth efficiency. However, implementing such an approach is quite challenging due to the optical reach limitations. Optical reach is a decreasing function of the bit rate and optical reach limitations can significantly restrict the potential gains of flexible modulation selection. One

key capability envisioned by the FON is being able to use different modulation levels at each segment of a light-path that connects the source of a demand to its destination possibly passing through several regenerators to maintain a certain level of signal quality. This requires single modulation assumption of RLP to be abolished. So with this new architecture, it is possible to use regenerators to switch modulation formats on a light-path such that the spectrum allocation is minimum while the signal quality is within the predefined limits. In short, FON can flexibly choose its transmission parameters according to the varying traffic conditions and significantly increase the resource utilization efficiency [39, 133].

It is clear that the better exploitation of the opportunities offered by the FON architecture requires the solution of routing, bandwidth allocation, modulation level selection and regenerator location problems jointly but the problem of solving all these problems concurrently is a challenging one. Indeed, researchers indicate that lack of such an efficient algorithm constitutes a major barrier for the adoption of this novel technology [133]. Since FON architecture is quite new, despite its immense practical importance, this theoretically challenging problem is not well studied in the literature. As a first attempt to close this gap, we consider FON architecture in Chapter 4 and extend our studies on RLP by considering multiple modulation levels, capacitated edges and communication demands as origin destination (OD) pairs with different volumes of data to be exchanged that generalizes all-pairs-connectivity requirement of RLP. More formally, we introduce the regenerator location problem in flexible optical networks (RLP-FON) that seeks the best routing, modulation level and regenerator location combination that minimizes the regenerator deployment costs while not using more than a predetermined portion of the fiber bandwidth. Adding two new facets to RLP; modulation level selection and bandwidth capacity limitations of the fiber links, RLP-FON finds the minimum amount of network resources needed to satisfy a given set of transmission demands.

RLP-FON arises both in the design and network management layers. In the design phase, RLP-FON determines the minimum amount of network equipment (regenerators, router chassis, optic line cards etc.) required to satisfy the targeted demand whereas in the network operation layer RLP-FON can help reduce the

operating cost of the network by identifying the network elements that could be put into sleep when the actual demand is less than the maximum supported demand size. The significance of the latter can be better understood considering the fact that optical backbone networks are designed somewhat to support the worst case demand scenarios and peak demand is more than two times larger than the minimum observed on the same day [119]. Indeed, motivated by such an opportunity, hardware developers intensified their research and development efforts on manufacturing network devices with capabilities to go to sleep mode to save energy.

As a major contribution we present a path-segment formulation for RLP-FON and develop a branch-and-price algorithm to solve it. As the results of our numerical experiments show the resulting algorithm is a quite efficient one that can solve realistic problem instances in reasonable time limits. To the best of our knowledge this dissertation is the first study in which path-segments instead of paths are used as the variables in a column generation framework.

The path-segment based formulation approach we introduce in Chapter 4 for RLP-FON is quite efficient to handle reach limitations and flexible to include various side constraints on the routes that connect OD pairs. As such, path-segment based formulations bring a new perspective to model relay location problems and developing solution algorithms. Indeed, we extend our initial path-segment formulation to address network design problem with relays and refueling station location problems studied in Chapters 5 and 6.

1.4 Network Design Problem with Relays

Practical settings for the RLP and RLP-FON problems do not require considering edge design together with the relay locations. However, for many applications arising in telecommunications and transportation it is beneficial to consider edge design and relay locations together. One such example is the network design problem with relays (NDR).

Motivated by a telecommunications network design project in Alberta, the network design problem with relays (NDR) is introduced by Cabral et al. [17]. A classical network design instance is represented by an undirected graph corresponding to the physical transportation or the telecommunications network. A given set of origin-destination (OD) pairs should be routed through this network. There is an associated establishment cost for every link/edge in this network and once an edge is chosen in the design, it can be used in the routing of any number of OD pairs. The aim is to choose a subset of edges in the most cost-effective manner so as to enable the routing of every OD pair. The network design problem with relays shares all these characteristics with a classical network design problem. However, the commodities have also reach limitations and while traversing an edge the reach diminishes by an amount proportional to the length of this edge. Ultimately, relays should be located in order to extend the reach and there is an associated cost to enhancing a node in the graph with the relay capability. The route for each OD pair should be such that two consecutive nodes from the set of terminal and relay nodes on this route should be within the reach limitation. The aim is to choose a subset of edges and a subset of nodes so as to enable the proper communication of all the OD pairs in the most cost-effective manner. Variations of this general form are also studied in the literature. In the directed network design problem with relays (DNDR), the underlying graph is directed. In the single-source network design problem with relays (NDR-S), the origin is the same for each OD pair.

Extending our studies on RLP by considering edge design and relay location decisions jointly, we study the Network Design Problem with Relays (NDR) in Chapter 5. Various network design problems are closely related with NDR. One such example is the Steiner tree problem (STP) [56]. When all pairwise communications are required for a given subset of nodes and the threshold value is arbitrarily large, NDR reduces to STP. NDR also generalizes the node-weighted Steiner tree problem (NSTP) [124] and the Steiner tree problem with hop constraints (STPH) [140]. Note that, with the same demand set defined above, the threshold and edge lengths equal to one, NDR becomes the same problem as NSTP. Similarly, NDR where the threshold is the allowed number of hops and

edge lengths are all equal to one, solves STPH. Obviously, RLP is also a special case of NDR where each node pairing in the graph is an OD pair and all edge costs are assumed to be zero.

In Chapter 5 we modify our path-segment based multi-commodity flow formulation for the RLP-FON to include edge design decisions and present a branch-and-price algorithm to solve the NDR. We first present some useful properties of an optimal solution of NDR and using these properties we generate an optimality cut to strengthen our initial formulation. We pay a special attention to the special case NDR-S which has also been the scope of studies such as Cabral et al. [17], Li et al. [90] and Leitner et al. [89]. The special attention for NDR-S problem in the literature stems from both practical and theoretical reasons. On the practical side, for many telecommunications and transportation applications there is a special source from which signals or some commodities are disseminated to other terminal nodes as we have in server-client network architectures considered by Cabral et al. [17]. In multicasting applications, which have gained recent popularity in optical networks due to the growing demand on video streaming services, a point-to-multipoint connection from a source node to multiple destination nodes is required. Sometimes these special nodes are hubs where signals/commodities are congregated or exchanged. In a transportation setting the common source could be a depot from which some goods are distributed to their final destinations; it could be the center where entities from terminal nodes are collected and mode of transportation is switched. When the number of OD pairs grow, NDR becomes harder to solve. In solving these large problems, NDR can be decomposed into several NDR-S problems. For instance, one can relax NDR in a Lagrangian manner to obtain single source problems. As such, having an efficient algorithm to solve NDR-S is of interest to devise efficient solution algorithms for the general NDR problem. On the theoretical side, NDR-S also has some quite interesting properties. We can show that there is an optimal design which satisfies certain conditions and exploiting these properties we propose an improved formulation that uses the cut characterization of Steiner trees. We derive valid inequalities and optimality cuts and implement a branch-and-price-and-cut algorithm to solve our tree formulation that contains an exponential number of constraints involving

an exponential number of variables. The resulting algorithm can exactly solve problem instances up to 500 nodes and 250 OD pairs that are not attempted by even the heuristic solution approaches in the literature.

1.5 Refueling Station Location Problem

Perhaps the closest transportation related application setting to RLP is the refueling station locations for alternative-fuel vehicles (AFVs). Vehicle range is the key factor for determining the locations and number of refueling stations for completing a journey, especially a long-distance one. The location of switching/recharging stations for accommodating electric/hybrid vehicles has been the scope of recent research [51, 80, 99, 104, 143, 144, 146]. It is not hard to see that the refueling station placement problem with its inherent coverage and range limitations is closely related with the RLP. However, the former problem has some dimensions which RLP lacks in its basic form. For example in RLP establishing a feasible (regeneration-enabled) route between two nodes is considered sufficient when meeting the connection demand. However, the physical length of some regeneration-enabled-routes might be unacceptable for a typical AFV driver. Considering this issue, we extend our studies on RLP in Chapter 6 and work on refueling station location problem for the alternative fuel vehicles.

Alternative fuel vehicle (AFV) technologies aim at reducing the greenhouse gas emissions, the cost of transportation and the dependence on export oil. Due to economic, security and environmental concerns associated with fossil fuels, the penetration of alternative fuel vehicles (AFV) into the transportation network is on the rise. Penetration of these game-changing technologies bring about several opportunities for different players of transportation sector. However, a widespread adoption of vehicles by the community is contingent upon the availability of refueling stations for alternative fuels. Lack of these stations is identified as one of the foremost barriers by several researchers [9, 80, 95, 96, 120]. On the other hand, establishing new refueling stations by the private sector necessitates

a large number of vehicles on the road [80, 96, 97]. This so called ‘chicken-egg’ problem [80, 96, 143] led to several studies flourish in the recent literature. Commonly assuming a government participation in the initial phase of refueling station establishment, the major concern has been to locate a given number of stations in a road network refueling station location problem for the alternative fuel vehicles.[9, 25, 71, 72, 80–82, 95, 96, 99, 120, 143–145, 155].

As we have discussed in Section 1.2, because of the different physical properties of the commodities transferred we need to revise our approach for RLP to address refueling station location problem and consider the total length of the route for each OD pair. In particular we focus on the Deviation Flow Refueling Location model DFRLM by Kim and Kuby [71] that aims to locate a limited number of refueling stations in order to maximize the flow volume that can be refueled respecting the range limitations of the alternative fuel vehicles and the shortest path deviation tolerances of the drivers. Adding routing decisions of AFV drivers to this model we introduce the Refueling Station Location Problem (RSLP). Different than RLP and RLP-FON, the total length of the route for a traffic demand is a concern for RLSP as a result of drivers’ intolerance for deviating from the shortest/fastest route to complete their trip. We modify our path-segment formulation of RLP-FON to include circuitry constraints and present a branch-and-price algorithm that relaxes the simple path assumption for the first time in the literature and implicitly takes into account deviation tolerances without the pregeneration of the routes. Our unifying solution approach can also handle multiple vehicle types and include AFV driver preferences. We conduct extensive numerical experiments to solve this theoretically challenging and practically important problem. Regarding to refueling station location problem, the contributions of this dissertation to the existing literature can be summarized as follows:

- We introduce the refueling location station problem with routing (RSLP-R) that generalizes the DFRLM to handle the non-simple path deviations from the shortest path and present its complexity status.
- We propose a branch-and-price algorithm for solving the RSLP-R. Similar

to enhanced DFRLM, the solution time decrease is significant with respect to the original DFRLM model. Moreover, because the algorithm does not require the explicit enumeration of paths, it scales very well to higher network dimensions.

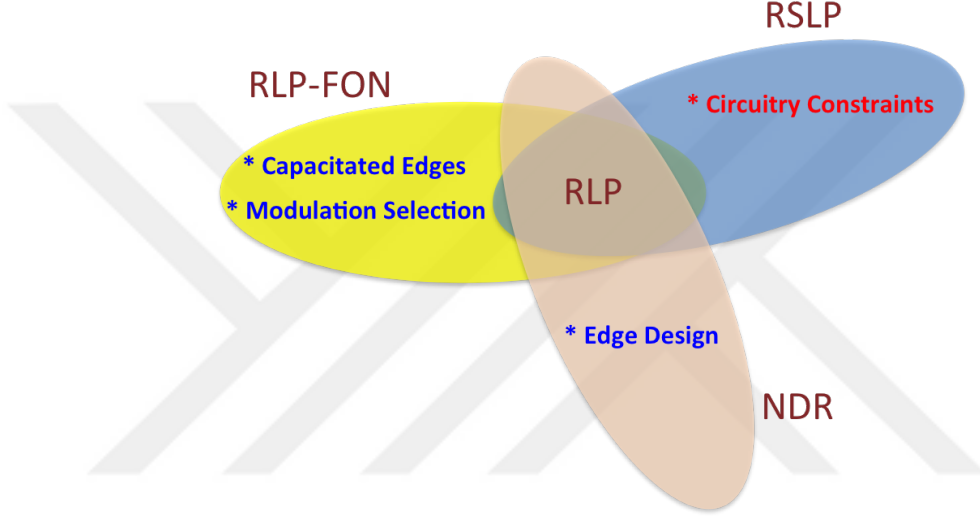


Figure 1.1: Relay location problems studied in this dissertation

1.6 Conspectus

In this study we consider different practical settings and problems arising in telecommunications and transportation for which the main concern is to find optimal relay placements. The relay location problems studied in this dissertation and their relation with each other are depicted in Figure 1.1. We analyze relay location problems with the hub location perspective and point out relevant transportation applications. In each chapter we investigate a different relay placement problem in a self contained manner and propose novel formulation approaches and efficient solution algorithms. Although each problem has its own specific properties they all share some common structure due to the relay notion and we utilize this common base to present a general framework to approach these problems.

We approach relay location problems from different angles: network topologies, capacities, costs, and demands and provide significant theoretical results. For the relay location problems, the reach limitations for the related entities pose the main challenge and we propose novel path-segment based formulation approaches to incorporate these constraints in an efficient way. Path-segment based formulations do not only provide an efficient way to include reach constraints but they are quite flexible to incorporate different requirements coming from different practical limitations. So we are able to extend our solution approaches to several significant relay location problems in the literature and present exact solution methods that could compete with state of the art heuristic algorithms in the literature.

Chapter 2

Nomenclature

In each of the following chapters we investigate a different relay placement problem in a self contained manner and propose novel formulation approaches and solution algorithms. Although each problem has its own special aspects and our formulations and solution approaches change accordingly they repeatedly use some common notions and structures. So in this chapter we present the common notation and definitions in general and state their specific intended use in each chapter if needed.

Throughout the text, $G = (V, E, l)$ will be our input weighted graph with node set $V = \{1, 2, \dots, n\}$, edge set E and $l_e > 0$ the length of edge $e \in E$. We shall assume that $d_{max} > 0$ is the given threshold of relay-free transportation/communication, i.e., the transport range or the optical reach. In other words, two nodes of distance at most d_{max} in G can exchange entities without any need for a relay. When practical setting of the problem requires to have more than one threshold level we state it explicitly and define the needed change in the notation. We assume without loss of generality that $l_e \leq d_{max}$ for every $e \in E$ since any edge violating this condition can simply be deleted from G . Induced by the edge set E we define the arc set A which contains two arcs $\bar{e} = (i, j)$ and $\underline{e} = (j, i)$ for each edge $e = \{i, j\} \in E$ with $l(\bar{e}) = l(\underline{e}) = l(e)$. The cost for relay placement in node $i \in V$ is denoted by h_i , the relay usage cost for some entity/commodity is

given by η and the cost of building/activating an edge in the network is annotated by $c_e \geq 0$ for $e \in E$. The capacity of an edge $e \in E$ is denoted by ω_e .

The set of origin destination (OD) pairs between which a commodity/entity is to be exchanged is depicted as $K = \{1, 2, \dots, \kappa\}$. For each $k \in K$, we denote $\mathcal{O}(k)$ as the origin node, $\mathcal{D}(k)$ as the destination node.

One key concept we use in our solution approaches is the closure graph transformation. Let d_{ij} be the length of the shortest path from i to j in G . After solving an all pairs shortest path problem in G , we create the unweighted *closure graph* (V, E^c) , where the edges correspond to pairs of nodes that can communicate with no need for regeneration, i.e., $E^c = \{\{i, j\} : i, j \in V, i \neq j, d_{ij} \leq d_{max}\}$. Let $A^c = \{(i, j) \cup (j, i) : \{i, j\} \in E^c\}$ be the set of arcs induced by the edges in the closure graph. A similar construction, under the name of “communication graph”, is introduced to the literature by Chen et al. [27]. Throughout the text the notation $[G]^c$ will correspond to the operation of “taking the closure” of an input graph $G = (V, E)$. In other words, $[G]^c = (V, E^c)$. For notational convenience we also define $V_q = V \setminus \{q\}$, $G_q = G(V_q)$, $G^c = [G]^c = (V, E^c)$ and $G_q^c = [G_q]^c = (V, E_q^c)$. Note that $E_q^c \subseteq E^c(V_q)$ as the closure edges of E^c induced by node q are not present in E_q^c .

We call a pair of nodes $(i, j) \in V \times V$ a *plausible pair* if $(i, j) \in A^c$. For different formulations and solution approaches we present in the following chapters, we need to impose more conditions to classify a node pair as plausible. So in each chapter we give the specific definition of a *plausible pair* when needed.

A directed path is a sequence of arcs $(a_1, a_2, \dots, a_\beta)$ with $a_i = (n_{i-1}, n_i) \in A$, for $i = 1, \dots, \beta$ and $n_i \in V$ for $i = 0, \dots, \beta$. The directed path is non-simple if it repeats nodes and simple otherwise. Our formulations heavily depend on the notion of *path-segments*. A *path-segment* p is a directed simple path with some problem specific meaning such as a modulation level. We denote the source and destination nodes of a path-segment p as $o(p)$ and $d(p)$, respectively. For each *path-segment* p we denote $\bar{p}$ as the set of edges $e \in E$ such that p passes through $\bar{e}$ or $\underline{e}$ and define the indicator function $I(e, p)$ that returns 1 if $e \in \bar{p}$

and 0 otherwise. The length of a path-segment $l(p) = \sum_{e \in \bar{p}} l(e)$ is the sum of the lengths of the edges contained in $\bar{p}$. In each problem setting we define which path segments are considered to be feasible and denote $\mathcal{P}$ as the set of all those feasible path-segments.

For $S \subseteq V$, $\delta(S)$ denotes the set of edges in E that have one endpoint in S and the other in $V \setminus S$. Similarly, the set of edges with both end points in S are denoted by $E(S)$. We define $G(S)$ as the subgraph induced by S , i.e., $G(S) = (S, E(S))$. We adopt the same notation for the path-segments and for a given $S \subseteq V$, define $\mathcal{P}(S) = \{p \in \mathcal{P} : o(p), d(p) \in S\}$ as the set of all path-segments with both endpoints in set S . Similarly, we define $\delta^+(i)$ and $\delta^-(i)$ as the sets of path-segments that start and end at node i , respectively.

A route $P = (p^1 \dots, p^q)$ is an ordered union of path-segments $p^i, i = 1, \dots, q$ where $d(p^i) = o(p^{i+1})$ for $i = 1, \dots, q - 1$. We call a route feasible for a demand $k \in K$ if $o(p^1) = \mathcal{O}(k), d(p^q) = \mathcal{D}(k)$, $l(p^i) \leq d_{max}, i \in 1, \dots, q$, and $d(p^i)$ is a regenerator node for $i = 1, \dots, q - 1$.

Chapter 3

Regenerator Location Problem

In Chapter 1 we draw attention to the relation between the relay and hub location problems and suggest exploiting this relation to develop efficient solution methods. Now in this chapter we study RLP with a hub location perspective and present efficient methods to solve RLP and its extensions RLP-RF and RLP-NF. The results of our studies in this chapter have been contributed to the literature by Yildiz and Karasan [154]

For the broad literature on hub location research we refer the reader to the seminal works [105–107]. Moreover, Campbell [20], and more recently, Alumur and Kara [4] provide a comprehensive survey of this literature. Campbell [19] studies the hub covering problem that is closely related to the RLP, defines three coverage criteria for hubs and provides the first mixed integer programming (MIP) formulations for the problem. Kara and Tansel [63] prove that the single allocation hub covering problem is *NP*-hard and provide a linearization for the original quadratic model, which performs better than its previous counterparts. Wagner [141] provides high-performance preprocessing techniques to reduce the number of the variables and improve the problem formulations. Hamacher and Meyer [49] delineate facet-defining valid inequalities for the hub covering problem. In all these studies, the underlying hub network is assumed to be complete. Campbell et al. [22] relax the fundamental complete hub network assumption and instill a

network design perspective to the hub location problems. In a companion study Campbell et al. [23] provide integer programming formulations and optimal solution algorithms. Calik et al. [18] and Alumur et al. [5] also address the more general incomplete but connected hub network topologies. We refer the interested reader to [5, 24, 31, 98, 116, 147, 148] for the hub location studies that directly deal with the transportation networks. None of these studies consider system survivability in the case of hub failure or destruction as a significant aspect of the problem. Our study also relaxes the fundamental complete hub network assumption and builds hub networks resilient to hub or node failures.

Kim and O’Kelly [69] introduce the reliable q -hub location problems in hub-and-spoke networks. Using the probabilities of successful edge or hub flow transmissions as reliabilities, the reliability of the network performance can be measured. Kim and O’Kelly [69] formulate and solve two hub location models namely the q -hub maximum reliability and the q -hub mandatory dispersion models focusing on maximizing the network performance in terms of reliability based on empirical traffic loss rates among origin destination pairs. Both single and multiple allocation versions of the problem are addressed. Considering hub unavailability and alternative routes in air transportation systems, Zeng et al. [157] propose different versions of reliable hub location models. Davari et al. [33] and Zarandi and Davari [156] design reliable hub networks using fuzzy goal programming. Lei [88] and Hamidi et al. [50] utilize a hub interdiction viewpoint and present hub protection and preventive reliable hub location problems, respectively, to the literature. An et al. [6], Kim [68], Kim and O’Kelly [70] and Azizi et al. [8] bring the survivable network design perspective into the hub-and-spoke networks. With this perspective, backup hubs and alternative routes are designed to provide a continuum of service with a typical objective of minimizing the transportation cost. In our study, the fundamental complete hub network assumption, which is present in the mentioned relevant studies in survivable hub location literature, is relaxed. Moreover, the designs should obey the transport range (optical reach) limitations respecting the edge lengths of an input transportation (optical) network. In particular, for RLP, given an underlying network (typically a sparse

one), hubs (regenerators) should be located in such a way that between any origin destination pair, there exists a path visiting perhaps more than two hubs such that each segment of this path is within the transport range. With RLPRF, the hub network design should respect RLP connectivity requirements even if an arbitrary hub node fails. Finally, with RLPNF, the hub network design should respect RLP connectivity even if any node in the original network fails. In other words, for the survivable versions, connectivity induced by the original network after the particular failure should be considered. On the other hand, our designs as per the application requirements are sought solely from the viewpoint of enabling communication under failures and transportation costs are overlooked.

There is a rich literature on network survivability from a topological design viewpoint. Parallel to the resilience criteria we use in this chapter, Grotschel et al. [48] argue that two-connected networks are able to provide a sufficient level of survivability because in most cases, the probability of two nodes failing at the same time is significantly small. This resilience criterion is widely accepted throughout the operations research literature; the reader can refer to [41, 100, 129], and [86] for relevant studies and detailed surveys on this topic. Similar to our approach, Chimani et al. [29] show that supported by a strong MIP formulation, the branch-and-cut procedure can efficiently solve two-node-connected Steiner network problems of realistic sizes. Kerivin and Mahjoub [66] investigate techniques used to solve survivable network design problems where the measure of robustness is modeled in terms of node or link connectivity. In our context we find it more relevant to build resilience against node failures rather than on link (edge) failures because in applications where nodes represent complex processes (regeneration, charging, refueling, amplification, encryption, etc.), nodes can be much more prone to malfunction, expensive to build redundancy into or more likely to be targets of deliberate attacks.

Regarding the RLP literature, Yetginer and Karasan [152] are the first to introduce the sparse regenerator placement in a static routing environment. The authors take network survivability into account and provide heuristics to place the regenerators, giving one working and one restoration path between any two nodes. However, they establish survivability by edge disjointness, which is not

adequate when the context dictates node disjointness as a survivability criterion. The paper works under the simple path assumption which unnecessarily limits the solution space and hence may result in suboptimal solutions. Konak [79] is a recent work dealing with the same design problem and providing heuristic solution methodologies.

Taking the geographical aspect of the RLP into account, Chen et al. [27] introduce it to the operations research literature, proving its NP -completeness and showing that it can be modeled as a special Steiner arborescence problem (SAP). Using this fact, they formulate a MIP model that can provide optimal solutions for relatively small instances of the problem. They adapt a branch-and-cut algorithm tailored for the SAP as a solution methodology, however, because the solution times are prohibitive for large instances of the RLP, they offer three heuristics and test the performances with an extensive experimental study. Similar to this study, we formulate the RLP as a MIP and solve it with a branch-and-cut procedure. However, Chen et al. [27] first transform the RLP to a SAP and then apply branch-and-cut. This transformation necessitates duplicating the nodes in the original input graph and hence prolongs the solution times. This does not occur in our formulation because we apply the branch-and-cut procedure directly to the original graph. Moreover, with the hub viewpoint, we are able to apply the separation algorithms in the restricted hub subgraph which leads to further efficiency. The maximum leaf spanning tree problem (MLSTP) and the minimum connected dominating set problem (MCDSP) are equivalent problems to the RLP. Using this equivalence, Sen et al. [125] provide another heuristic for the RLP. Lucena et al. [93] provide branch-and-cut solution methodologies for the MLSTP and MCDSP and ultimately for the RLP. In a recent study, Gendron et al. [46] propose efficient exact solution methodologies built on Benders decomposition and branch-and-cut techniques. Our experimental results prove that none of these exact solution methodologies can compete with the branch-and-cut algorithm developed in this chapter which has the advantage of working only in the hub network due to the hub location perspective we adopt.

We take the geographical aspect of the RLP as the focal point of our study in this chapter. The interested reader can refer to [108, 122, 123, 149] for studies that

investigate the RLP with a focus on different aspects such as blocking probabilities in dynamic optical networks with wavelength division multiplexing.

3.1 Mathematical Models

In this section, we provide some background information on the RLP, establish the NP -completeness results of the two new problems introduced and provide the notation that will be adopted throughout this chapter. For each of the three problems under the scope of this chapter a compact formulation and a cut formulation are proposed. We shall use hubs and regenerators interchangeably.

3.1.1 Definitions and Notation

In this chapter we mainly use the common notation defined in 2. For the formulations and solution methods specific to RLP, RLP-RF and RLP-NF we define the following additional notation

For two disjoint subsets S and T of V , we denote by $[S, T]$ the set of edges with one endpoint in S and the other in T . A given regenerator or hub set $R \subseteq V$ and d_{max} jointly induce a restricted form of connectivity in G , namely, $s, t \in V$ are *R -connected* if there exists a walk $\mathcal{W}$ from s to t on which any two consecutive nodes in $\{s, t\} \cup R$ are at most d_{max} apart. Such a walk in G corresponds to a path in G^c where every internal node is in set R .

3.1.2 Problem Complexity

The decision versions of the three problems studied in this chapter can be formalized as follows.

Given $G = (V, E, l)$, d_{max} value, and $k \leq n$, does there exist a regenerator set

$R \subseteq V$, $|R| \leq k$ such that:

- RLP *Regenerator Location Problem*: Any node pair is R -connected in G ?
- RLPRF *Regenerator Location Problem under Regenerator Failures*: Any node pair is R -connected in G_q for every q in set R ?
- RLPNF *Regenerator Location Problem under Node Failures*: Any node pair is R -connected in G_q for every q in set V ?

With the notion of R -connectivity non-simple paths might be more advantageous. In the following example (Figure-3.1a) with $d_{max} = 5$, the RLP has an optimal solution of one regenerator located at node 4 (Figure-3.1b). In particular, the connection between nodes 1 and 3 can be established by using the walk 1-2-4-2-3 and any solution requiring simple path connectivity results in a higher number of regenerators.

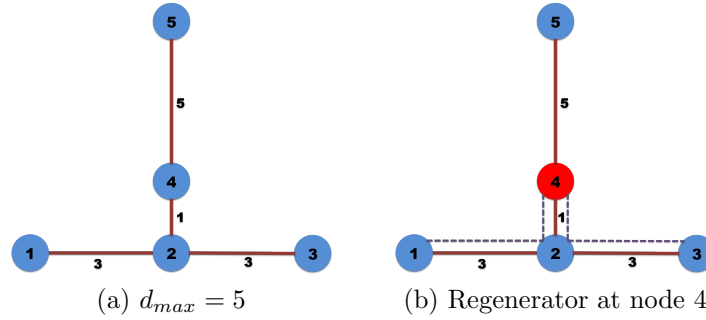


Figure 3.1: Non-simple paths

RLP is shown to be NP -complete in Chen et al. [27]. This result can also be derived from the NP -completeness proof in Kara and Tansel [63]. The following theorem is an immediate consequence of this result.

Theorem 1. *RLPRF and RLPNF are NP -complete.*

Proof. Both problems are in NP because checking for the feasibility of a given regenerator subset R for a specific origin-destination pair, and ultimately for all pairs, can be accomplished in polynomial time.

Let $G = (V, E, l)$ with $V = \{1, \dots, n\}$, d_{max} and k be an instance of RLP. Let $V' = \{i_1 \cup i_2 : i \in V\}$ and $E' = \{\{i_1, j_1\} \cup \{i_2, j_2\} : \{i, j\} \in E\} \cup \{\{i_1, i_2\} : i \in V\}$. Furthermore, let l' be the cost vector associated with E' edges. $l'_{\{i_1, j_1\}} = l'_{\{i_2, j_2\}} = l_{\{i, j\}}$ for all $\{i, j\} \in E$ and $l'_{\{i_1, i_2\}} = 0$ for all $i \in V$. Let $G' = (V', E, l')$, d_{max} and $2k$ be an instance of RLPRF(RLPNF). It is easy to note that the RLP has a yes answer if and only if the RLPRF(RLPNF) has a yes answer. $\square$

We assume that the input graph G is two-connected. To eliminate trivial cases, in our formulations we shall also assume that the input networks necessitate the use of at least three regenerators. This is not a binding assumption because for each of our problem definitions, it is easy to enumerate all possible hub sets of size at most two and check for feasibility in polynomial time.

3.1.3 Regenerator Location Problem: Compact Formulation

In this subsection, we provide new compact and cut formulations for the RLP. Within the framework of hub covering, given the closure graph $G^c = (V, E^c)$, we seek a (hub) set $R \subseteq V$ of minimum cardinality such that $G^c(R)$ is connected (not necessarily in a complete fashion) and that for every node $i \in V \setminus R$, there exists a node $j \in R$ such that $\{i, j\} \in E^c$, i.e., every demand node is covered by at least one hub, i.e., multi-allocation is allowed in hub location terminology. In this fashion, we guarantee that any two nodes in V are R -connected. In other words, solving RLP in G^c is equivalent to solving the hub covering location problem with the “closeness” criterion of Campbell [19] in G^c with the relaxation of the complete hub network assumption. Note that the design is only pertinent to the choice of hub locations. Once such locations are fixed, the union of closure edges interconnecting these locations, i.e., $E^c(R)$ make up the hub network.

In the compact formulation for the RLP, we shall require a tree rooted at a chosen regenerator node visiting all other regenerator nodes using only the edges in E^c to guarantee the connectivity of the hub network. We define the following

decision variables.

$$s_i = \begin{cases} 1, & \text{if node } i \in V \text{ is the chosen root regenerator} \\ 0, & \text{otherwise,} \end{cases}$$

$$r_i = \begin{cases} 1, & \text{if node } i \in V \text{ is a regeneration point} \\ 0, & \text{otherwise,} \end{cases}$$

$$x_{ij} = \text{amount of flow on arc } (i, j) \in A^c \text{ in the designed tree.}$$

We would like the flow variables to satisfy the following balance equations.

$$\sum_{j:(i,j) \in A^c} x_{ij} - \sum_{j:(j,i) \in A^c} x_{ji} = \begin{cases} \sum_j r_j - 1, & \text{if node } i \in V \text{ is the chosen root regenerator} \\ -1, & \text{if node } i \in V \text{ is a non-root regenerator node} \\ 0, & \text{if node } i \in V \text{ is not a regenerator node.} \end{cases}$$

These relations can be induced by the following nonlinear form of the flow balance equations.

$$\sum_{j:(i,j) \in A^c} x_{ij} - \sum_{j:(j,i) \in A^c} x_{ji} = (\sum_j r_j - 1)s_i - r_i(1 - s_i) \quad \forall i \in V.$$

The formulation for the RLP is as follows.

$$\min \sum_{i \in V} r_i \quad (3.1)$$

$$\text{s.t. } \sum_{i \in V} s_i = 1 \quad (3.2)$$

$$s_i \leq r_i \quad \forall i \in V, \quad (3.3)$$

$$\sum_{j: \{i,j\} \in E^c} r_j \geq 1 \quad \forall i \in V, \quad (3.4)$$

$$\sum_{j: (i,j) \in A^c} x_{ij} - \sum_{j: (j,i) \in A^c} x_{ji} \geq \sum_j r_j - 1 - (n-1)(1-s_i) \quad \forall i \in V, \quad (3.5)$$

$$\sum_{j: (i,j) \in A^c} x_{ij} - \sum_{j: (j,i) \in A^c} x_{ji} \geq -r_i \quad \forall i \in V, \quad (3.6)$$

$$\sum_{i: (i,j) \in A^c} x_{ij} \leq (n-1)r_j \quad \forall j \in V, \quad (3.7)$$

$$r_i, s_i \in \{0, 1\} \quad \forall i \in V, \quad (3.8)$$

$$x_{ij} \geq 0 \quad \forall (i,j) \in A^c. \quad (3.9)$$

The objective function (3.1) is the number of nodes that need to be equipped with the regeneration property. Constraints (3.2) and (3.3) force one of the regenerator nodes to be chosen as the root node. Due to constraints (3.4), any node (demand or hub) will be connected to the hub network in the closure graph. Because we assume the hub network consists of at least three hubs, this constraint is valid for hub nodes as well. Constraint set (3.5)-(3.6) provides a linearization for the flow balance equations. It forces the chosen source regenerator to have a net outgoing flow amount equal to the number of remaining regenerator nodes and any other regenerator node to have a net incoming flow value of one unit. The constraints become redundant if node i is not chosen as a regenerator. The x variables may take on positive values only in the designed hub network. In this fashion, it is guaranteed that a path will initiate from the root hub to every other hub in the closure graph. If an arc of the closure graph is utilized in any one of these paths, constraints (3.7) force the head node to become a regenerator point. Constraints (3.8)-(3.9) are variable restrictions.

3.1.4 Regenerator Location Problem: Cut Formulation

To facilitate the use of a branch-and-cut algorithm in the solution methodology, we provide an equivalent cut formulation for the RLP. The following decision variables will be used.

$$x_e = \begin{cases} 1, & \text{if edge } e \in E^c \text{ is present in the (designed) regenerator subgraph} \\ 0, & \text{otherwise,} \end{cases}$$

$$r_i = \begin{cases} 1, & \text{if node } i \in V \text{ is a regeneration point} \\ 0, & \text{otherwise.} \end{cases}$$

Note that there is no design with respect to the x variables; the choice of the hub nodes defines these variables uniquely. In other words, we have the following nonlinear relationship.

$$x_{ij} = r_i r_j \quad \forall \{i, j\} \in E^c. \quad (3.10)$$

Similar to the compact formulation, the cut formulation will force the hub network to be connected and every demand node to be covered by some hub in this subgraph.

$$\min \sum_{i \in V} r_i \quad (3.11)$$

$$\text{s.t.} \quad \sum_{j: \{i, j\} \in E^c} r_j \geq 1 \quad \forall i \in V, \quad (3.12)$$

$$x_{ij} \leq r_i \quad \forall \{i, j\} \in E^c, \quad (3.13)$$

$$x_{ij} \leq r_j \quad \forall \{i, j\} \in E^c, \quad (3.14)$$

$$x_{ij} \geq r_i + r_j - 1 \quad \forall \{i, j\} \in E^c, \quad (3.15)$$

$$x(\delta(S)) \geq r_i + r_j - 1 \quad \forall S \subset V, i \in S, j \notin S, \quad (3.16)$$

$$x_e \in \{0, 1\} \quad \forall e \in E^c, \quad (3.17)$$

$$r_i \in \{0, 1\} \quad \forall i \in V. \quad (3.18)$$

The number of regenerator nodes is minimized through (3.11). Every node i should be able to reach the regenerator subgraph via a closure edge (valid due to our assumption of having more than one regenerator in the resulting design), and this is forced with constraints (3.12). Constraints (3.13)-(3.15) linearize the relationship (3.10). Constraints (3.16) are the cut constraints. If S and $V \setminus S$ induce a partition of V , each having a regenerator node, then there should be at least one edge crossing this partition in the regenerator subgraph for connectivity. Finally, constraints (3.17)-(3.18) force the required variable restrictions. It is enough to force the integrality of the r variables only since they imply the integrality of the x variables.

We would like to remark that both the compact and the cut formulations provided above are novel in the RLP as well as the hub location literature. In particular, our single-commodity flow formulation is different from its multi-commodity counterpart in Alumur et al. [5] since the hub network is induced by the closure edges and there is no need to keep the connectivity information explicitly. Similarly, the cut formulation is quite different from its counterpart in Chen et al. [27] where a rooted Steiner tree is sought on a transformed graph rather than a rooted tree on the closure graph. Indeed, we believe that the differences in the performances of the algorithms are merely due to the differences in the dimensions of the underlying input graphs.

3.1.5 Regenerator Location Problem under Regenerator Failures: Compact Formulation

Let $R \subseteq V$ be the set of nodes selected for the regeneration property. Under the RLPRF, we need to guarantee that the removal of any node in R will not destroy connectivity. Unfortunately, the closure graph utilization for this problem is not as direct as it is for the RLP. Namely, simply forcing the resulting hub network design to be two-connected and every demand node to be covered by at least two hubs will not be sufficient, due to the information loss that will occur in the transfer from the original graph to the closure graph. The following example

is included to clarify this issue. For the original graph depicted in Figure 3.2a, consider the closure graph induced by $d_{max} = 5$ and nodes 2, 3 and 6 as hubs. The resulting hub network is two-connected and every demand node is connected to at least two hub nodes as seen in Figure 3.2b. However, in G_3^c , demand node 4 is disconnected from the hub network and hence $\{2, 3, 6\}$ is not a feasible solution for the RLPRF. Indeed, the optimal design shown in Figure 3.2d has 4 hub nodes.

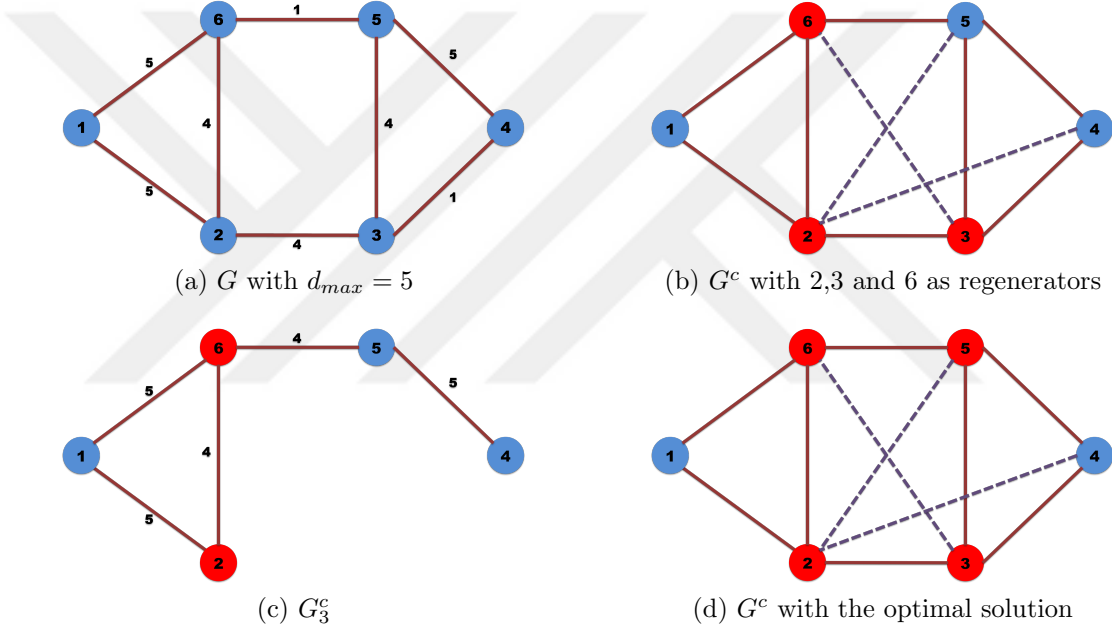


Figure 3.2: Working with G_q^c rather than G^c

We first present a compact formulation for the RLPRF. Let s^1 and s^2 be two distinct hub nodes, to be chosen from V in the resulting design. For any node $q \in V$ chosen as a regenerator node in the design, our model will force s^1 to reach every $t \in V_q \setminus \{s^1\}$ using edges in E_q^c . In other words, the model will build a rooted tree at s^1 using only edges in E_q^c , visiting all the remaining regenerator nodes in V_q . Similarly, since node s^1 is a regeneration point, its removal should not destroy connectivity. This is forced by rooting a tree at s^2 to every regenerator node in $V_{s^1} \setminus \{s^2\}$ using only $E_{s^1}^c$ edges. In this fashion, we shall guarantee that even if a regenerator node is deleted from the graph, the closure graph induced by the remaining edges is connected, and hence there is resilience against failures. Let

$$s_i^1 = \begin{cases} 1, & \text{if node } i \in V \text{ is the chosen root regenerator} \\ 0, & \text{otherwise,} \end{cases}$$

$$s_i^2 = \begin{cases} 1, & \text{if node } i \in V \text{ is the chosen root regenerator when node } s^1 \text{ is deleted} \\ 0, & \text{otherwise,} \end{cases}$$

$$r_i = \begin{cases} 1, & \text{if node } i \in V \text{ is a regeneration point} \\ 0, & \text{otherwise,} \end{cases}$$

x_{ij}^q = amount of flow on arc $(i, j) \in A_q^c$ in the designed tree rooted at s^1 or s^2 (when $q = s^1$).

In particular, we would like the flow variables to satisfy the following balance equations for all $q \in V, i \in V_q$.

$$\sum_{j:(i,j) \in A_q^c} x_{ij}^q - \sum_{j:(j,i) \in A_q^c} x_{ji}^q = \begin{cases} \sum_j r_j - 2, & \text{if } s_i^1 = 1 \text{ and } r_q = 1, \\ \sum_j r_j - 2, & \text{if } s_i^2 = 1 \text{ and } s_q^1 = 1, \\ -1, & \text{if } s_i^1 = 0, s_i^2 = 0, r_i = 1, \text{ and } r_q = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Our compact formulation for the RLPRF after proper linearization of the above

relations is as follows.

$$\min \sum_{i \in V} r_i \quad (3.19)$$

$$\text{s.t. } \sum_{i \in V} s_i^1 = 1 \quad (3.20)$$

$$\sum_{i \in V} s_i^2 = 1 \quad (3.21)$$

$$s_i^1 + s_i^2 \leq r_i \quad \forall i \in V, \quad (3.22)$$

$$\sum_{j: \{i,j\} \in E^c} r_j \geq 2 \quad \forall i \in V, \quad (3.23)$$

$$\sum_{j: \{i,j\} \in E_q^c} r_j \geq r_q \quad \forall q \in V, i \in V_q, \quad (3.24)$$

$$\sum_{j: (i,j) \in A_q^c} x_{ij}^q - \sum_{j: (j,i) \in A_q^c} x_{ji}^q \geq \sum_j r_j - 2 - M(1 - s_i^1) - M(1 - r_q) \quad \forall q \in V, i \in V_q, \quad (3.25)$$

$$\sum_{j: (i,j) \in A_q^c} x_{ij}^q - \sum_{j: (j,i) \in A_q^c} x_{ji}^q \geq \sum_j r_j - 2 - M(1 - s_i^2) - M(1 - s_q^1) \quad \forall q \in V, i \in V_q, \quad (3.26)$$

$$\sum_{j: (i,j) \in A_q^c} x_{ij}^q - \sum_{j: (j,i) \in A_q^c} x_{ji}^q \geq -r_i - M(1 - r_q) \quad \forall q \in V, i \in V_q, \quad (3.27)$$

$$\sum_{i: (i,j) \in A_q^c} x_{ij}^q \leq M r_j \quad \forall q \in V, j \in V_q, \quad (3.28)$$

$$r_i, s_i^1, s_i^2 \in \{0, 1\} \quad \forall i \in V, \quad (3.29)$$

$$x_{ij}^q \geq 0 \quad \forall q \in V, (i, j) \in A_q^c. \quad (3.30)$$

The objective function (3.19) minimizes the cardinality of the regenerator set. Constraints (3.20)-(3.22) ensure that two distinct regenerator points will be chosen as roots s^1 and s^2 . Constraints (3.23)-(3.24) are the cover constraints for resilience against regenerator failures. In particular, every node (demand or hub) will be reachable from two different hubs in the closure graph G^c , and when a regenerator node q is deleted, every node (demand or hub) will be reachable from another hub node in the resulting closure graph G_q^c . Adapted flow balance constraints (3.25)-(3.27) guarantee that when an arbitrary node q (different from s_1) selected as a regeneration point is deleted from G , there will be a directed

path from regenerator s_1 to every other regenerator node in the closure graph G_q^c . Similarly, when the deleted node is the chosen root node s^1 , there will be another tree rooted at s^2 reaching every other regenerator node in the resulting closure graph. A node visited by any of these rooted tree arcs should be a regenerator node, which is forced via constraints (3.28). $M = n - 1$ is a large enough number to ensure that constraints (3.25) through (3.28) are proper linearizations. Finally, constraints (3.29)-(3.30) are the required variable restrictions.

3.1.6 Regenerator Location Problem under Regenerator Failures: Cut Formulation

For the resulting design to be survivable against regenerator failures, the removal of any regenerator node should not spoil connectivity. This in turn will imply two-connectivity of the resulting design. The following cut formulation is based on these facts, and we use the same decision variables as those in the cut formulation of the RLP.

$$\min \sum_{i \in V} r_i \quad (3.31)$$

$$\text{s.t.} \quad \sum_{j: \{i,j\} \in E^c} r_j \geq 2 \quad \forall i \in V, \quad (3.32)$$

$$\sum_{j: \{i,j\} \in E_q^c} r_j \geq r_q \quad \forall q \in V, i \in V_q, \quad (3.33)$$

$$x_{ij} \leq r_i \quad \forall \{i,j\} \in E^c, \quad (3.34)$$

$$x_{ij} \leq r_j \quad \forall \{i,j\} \in E^c, \quad (3.35)$$

$$x_{ij} \geq r_i + r_j - 1 \quad \forall \{i,j\} \in E^c, \quad (3.36)$$

$$\sum_{\{k,l\} \in E_q^c: k \in S, l \notin S} x_{kl} \geq r_i + r_j + r_q - 2 \quad \forall q \in V, S \subset V_q, i \in S, j \notin S, \quad (3.37)$$

$$x_e \in \{0, 1\} \quad \forall e \in E^c, \quad (3.38)$$

$$r_i \in \{0, 1\} \quad \forall i \in V. \quad (3.39)$$

As usual, the objective given in (3.31) is to minimize the number of regenerator nodes. Assuming at least three regenerator nodes in the design, any node should reach two regenerator nodes in the closure graph G^c through constraints (3.32). Once a regenerator node q is deleted from the graph, any node should be able to reach a regeneration point in the resulting closure subgraph G_q^c for resilient connectivity. This is forced by constraints (3.33). Since the x variables are only defined in the regenerator subgraph of the design, using constraints (3.34)-(3.36), we linearize relationship (3.10). Constraints (3.37) are the adapted form of cut inequalities. Once a regenerator node q is deleted from G , if there exists a partition of the remaining subgraph such that both sets in the partition include regenerator nodes, then a closure edge in G_q^c should cross the cut in the design for connectivity. Finally, (3.38)-(3.39) are the variable restrictions and (3.38) can be relaxed due to (3.34)-(3.36).

3.1.7 Regenerator Location Problem under Node Failures: Compact Formulation

For the RLPNF, we first present a compact formulation: Let s^1 and s^2 be two distinct nodes in V . For any node $q \in V_{s^1}$, independent of whether chosen as a regenerator point or not, we shall guarantee that s^1 can reach every other node in V using edges in E_q^c . In this fashion, we shall ensure that even if node q is deleted from the graph, the closure graph induced by the edges in E_q^c is connected. We shall ensure the same is true when s^1 is deleted from G by forcing s^2 to reach every other node in $V_{s^2} \setminus \{s^1\}$ through the edges induced by $E_{s^1}^c$.

The compact formulation for the RLPNF below parallels that of the RLPRF except that there is no conditioning on the node removed from the graph to be a regeneration point. $M = n - 1$ is a large enough value for this formulation to

work properly.

$$\min \sum_{i \in V} r_i \quad (3.40)$$

$$\text{s.t. } \sum_{i \in V} s_i^1 = 1 \quad (3.41)$$

$$\sum_{i \in V} s_i^2 = 1 \quad (3.42)$$

$$s_i^1 + s_i^2 \leq r_i \quad \forall i \in V, \quad (3.43)$$

$$\sum_{j: \{i,j\} \in E^c} r_j \geq 2 \quad \forall i \in V, \quad (3.44)$$

$$\sum_{j: \{i,j\} \in E_q^c} r_j \geq 1 \quad \forall q \in V, i \in V_q, \quad (3.45)$$

$$\sum_{j: (i,j) \in A_q^c} x_{ij}^q - \sum_{j: (j,i) \in A_q^c} x_{ji}^q \geq \sum_j r_j - 1 - r_q - M(1 - s_i^1) \quad \forall q \in V, i \in V_q, \quad (3.46)$$

$$\sum_{j: (i,j) \in A_q^c} x_{ij}^q - \sum_{j: (j,i) \in A_q^c} x_{ji}^q \geq \sum_j r_j - 2 - M(1 - s_i^2) - M(1 - s_q^1) \quad \forall q \in V, i \in V_q, \quad (3.47)$$

$$\sum_{j: (i,j) \in A_q^c} x_{ij}^q - \sum_{j: (j,i) \in A_q^c} x_{ji}^q \geq -r_i \quad \forall q \in V, i \in V_q, \quad (3.48)$$

$$\sum_{i: (i,j) \in A_q^c} x_{ij}^q \leq Mr_j \quad \forall q \in V, j \in V_q, \quad (3.49)$$

$$r_i, s_i^1, s_i^2 \in \{0, 1\} \quad \forall i \in V, \quad (3.50)$$

$$x_{ij}^q \geq 0 \quad \forall q \in V, (i, j) \in A_q^c. \quad (3.51)$$

3.1.8 Regenerator Location Problem under Node Failures: Cut Formulation

For the resulting design to be two-connected, the removal of any node should not spoil connectivity. The following cut formulation is based on this fact: For any node q , G_q^c should be connected.

$$\min \sum_{i \in V} r_i \quad (3.52)$$

$$\text{s.t.} \quad \sum_{j: \{i,j\} \in E^c} r_j \geq 2 \quad \forall i \in V, \quad (3.53)$$

$$\sum_{j: \{i,j\} \in E_q^c} r_j \geq 1 \quad \forall q \in V, i \in V_q, \quad (3.54)$$

$$x_{ij} \leq r_i \quad \forall \{i,j\} \in E^c, \quad (3.55)$$

$$x_{ij} \leq r_j \quad \forall \{i,j\} \in E^c, \quad (3.56)$$

$$x_{ij} \geq r_i + r_j - 1 \quad \forall \{i,j\} \in E^c, \quad (3.57)$$

$$\sum_{\{k,l\} \in E_q^c: k \in S, l \notin S} x_{kl} \geq r_i + r_j - 1 \quad S \subset V, i \in S, j \notin S, q \in V \setminus \{i,j\}, \quad (3.58)$$

$$x_e \in \{0, 1\} \quad \forall e \in E^c, \quad (3.59)$$

$$r_i \in \{0, 1\} \quad \forall i \in V. \quad (3.60)$$

Similar to the compact formulation, the cut formulation of the RLPNF closely follows that of the RLPRF; hence the explanations are not repeated here.

3.2 Path Recovery

In all of the envisioned applications of our problems, it is imperative to know the underlying transportation paths between every source destination pair. For the first two problems recovery of such paths can be done efficiently once the design is attained, but, this task turns out to be more challenging for our third problem.

Consider a specific source destination pair s and t . Let R^* be a solution in the resulting design and let $H = (R^* \cup \{s, t\}, E_H)$ be such that E_H is a subset of $E^c(R^* \cup \{s, t\})$, including only the closure edges $\{i, j\}$ for which there exists a path from i to j in G of length at most d_{max} , such that none of the intermediate nodes on this path belongs to $R^* \cup \{s, t\}$. In other words, the closure edges in $E^c(R^* \cup \{s, t\})$ induced by using nodes in $R^* \cup \{s, t\}$ as intermediate nodes are

omitted in E_H . Let A_H be the arc set induced by the edges in E_H . Note that H is connected for the RLP and two-connected for the RLPRF and RLPNF.

Finding a walk from s to t for R^* -connectivity is easy. Let P be a directed $s - t$ path in H . The required walk can be constructed simply by concatenating the original paths in G from i to j for every induced arc $(i, j) \in P$.

To satisfy the connectivity requirements of the RLPRF, for every source destination pair s and t we need to construct two $s - t$ walks in G that are R^* connected and that do not share a node from R^* . This can still be done in polynomial time by simply constructing two node disjoint paths from s to t in H and simply expanding each closure arc appearing in a path by one of its corresponding paths in the original graph and concatenating these paths.

The situation is somewhat different when trying to find two walks from s to t in G which are R^* -connected and node disjoint, i.e., satisfying the connectivity requirements of the RLPNF. A special case of this problem when $|R^*| = 0$ is equivalent to finding two node disjoint paths between s and t such that the longer one is the smallest which is an *NP*-complete problem (see Garey and Johnson [44]).

The following model will construct two paths from s to t in H , indicated by variables $P1$ and $P2$, respectively. For each closure arc induced by these variables, variables $p1$ and $p2$ will indicate the expanded path segments of length at most d_{max} in the original graph. The union of these path segments will be forced to

be node disjoint and will comprise the two required $s - t$ walks.

$$P1^{ij} = \begin{cases} 1, & \text{if closure arc}(i, j) \in A_H \text{ is used on the first closure path from } s \text{ to } t \\ 0, & \text{otherwise,} \end{cases}$$

$$P2^{ij} = \begin{cases} 1, & \text{if closure arc}(i, j) \in A_H \text{ is used on the second closure path from } s \text{ to } t \\ 0, & \text{otherwise,} \end{cases}$$

$$p1_{kl}^{ij} = \begin{cases} 1, & \text{if arc}(k, l) \in A \text{ is used on the first path's closure segment from } i \text{ to } j \\ 0, & \text{otherwise,} \end{cases}$$

$$p2_{kl}^{ij} = \begin{cases} 1, & \text{if arc}(k, l) \in A \text{ is used on the second path's closure segment from } i \text{ to } j \\ 0, & \text{otherwise,} \end{cases}$$

$$In1_l = \begin{cases} 1, & \text{if node } l \in V \text{ is used as an intermediate node on the first walk from } s \text{ to } t \\ 0, & \text{otherwise,} \end{cases}$$

$$In2_l = \begin{cases} 1, & \text{if node } l \in V \text{ is used as an intermediate node on the second walk from } s \text{ to } t \\ 0, & \text{otherwise.} \end{cases}$$

The flow based formulation is as follows.

$$\min \sum_{(i,j) \in A_H, (k,l) \in A} c_{kl}(p1_{kl}^{ij} + p2_{kl}^{ij}) \quad (3.61)$$

$$\text{s.t.} \quad \sum_{j:(i,j) \in A_H} P1^{ij} - \sum_{j:(j,i) \in A_H} P1^{ji} = \begin{cases} 1, & \text{for } i = s \\ -1, & \text{for } i = t \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in R^* \cup \{s, t\}, \quad (3.62)$$

$$\sum_{j:(i,j) \in A_H} P2^{ij} - \sum_{j:(j,i) \in A_H} P2^{ji} = \begin{cases} 1, & \text{for } i = s \\ -1, & \text{for } i = t \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in R^* \cup \{s, t\}, \quad (3.63)$$

$$\sum_{l:(k,l) \in A} p1_{kl}^{ij} - \sum_{l:(l,k) \in A} p1_{lk}^{ij} = \begin{cases} P1^{ij}, & \text{for } k = i \\ -P1^{ij}, & \text{for } k = j \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in V, (i, j) \in A_H, \quad (3.64)$$

$$\sum_{l:(k,l) \in A} p2_{kl}^{ij} - \sum_{l:(l,k) \in A} p2_{lk}^{ij} = \begin{cases} P2^{ij}, & \text{for } k = i \\ -P2^{ij}, & \text{for } k = j \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in V, (i, j) \in A_H, \quad (3.65)$$

$$\sum_{(k,l) \in A} c_{kl} p1_{kl}^{ij} \leq d_{max} P1^{ij} \quad \forall (i, j) \in A_H, \quad (3.66)$$

$$\sum_{(k,l) \in A} c_{kl} p2_{kl}^{ij} \leq d_{max} P2^{ij} \quad \forall (i, j) \in A_H, \quad (3.67)$$

$$\sum_{k:(k,l) \in A, (i,j) \in A_H} p1_{kl}^{ij} \leq M \text{In}1_l \quad \forall l \in V \setminus t, \quad (3.68)$$

$$\sum_{k:(k,l) \in A, (i,j) \in A_H} p2_{kl}^{ij} \leq M \text{In}2_l \quad \forall l \in V \setminus t, \quad (3.69)$$

$$\text{In}1_l + \text{In}2_l \leq 1 \quad \forall l \in V \setminus t, \quad (3.70)$$

$$P1^{ij}, P2^{ij} \in \{0, 1\} \quad \forall (i, j) \in A_H, \quad (3.71)$$

$$p1_{kl}^{ij}, p2_{kl}^{ij} \in \{0, 1\} \quad \forall (i, j) \in A_H, (k, l) \in A, \quad (3.72)$$

$$\text{In}1_l, \text{In}2_l \in \{0, 1\} \quad \forall l \in V \setminus t, \quad (3.73)$$

The objective function (3.61) minimizes the total length of the two $s - t$ walks to be constructed. The given objective function is simply chosen for completeness' sake. Flow balance constraints (3.62)-(3.63) construct two paths from s to t in H . If closure arc (i, j) is used in either of these closure paths, it should decompose into a path segment from i to j using the arcs in the original input network. This situation is satisfied with flow balance constraints (3.64)-(3.65). Moreover, because each of these constructed path segments is regenerator free, their lengths should be bounded by the d_{max} value, which is forced via constraints (3.66)-(3.67). Any node used as an intermediate node in a constructed walk will

be indexed via constraints (3.68)-(3.69) and node disjointness will be enforced through Constraints (3.70). $M = |R^*| + 1$ is satisfactorily large enough since in the worst case each one of the path segments might utilize the same node. Finally, constraints (3.71-3.73) enforce the proper variable restrictions.

3.3 Separation Algorithms

As our cut formulations in Section 3.1 contain an exponential number of constraints, we propose branch-and-cut algorithms for the three problems. The cut inequalities of all the formulations can be separated exactly in polynomial time by solving a series of minimum cut problems similar to the ones employed in the literature such as Labbé et al. [84], Fouilhoux et al. [42], and Karasan et al. [64]. Even though our cut inequalities can be separated exactly by solving minimum cut (maximum flow) problems as typically done in the literature, since the design of the hub network is induced by the design of the hub nodes, we can employ very effective heuristic separation procedures that are particular to our problems. In the following subsections, the three separation algorithms employed in the branch-and-cut procedures will be detailed.

3.3.1 Separation Algorithm for the RLP

Let (r^*, x^*) be a given solution of the LP relaxation to model (3.11)-(3.18) including a subset of the cut inequalities. We define the following sets. $R^* = \{i \in V : r_i^* > 0\}$, $E^* = \{\{i, j\} \in E^c | x_{ij}^* > 0\}$ and the undirected capacitated graph $G^* = (R^*, E^*, \omega)$ with the capacity of edge e , say ω_e set to x_e^* , i.e., G^* is the support graph induced by the fractional solution at hand. A cut inequality (3.16) is defined by a node set $S \subset V$ and fixed nodes $i \in S$ and $j \notin S$. For fixed $i, j \in R^*$, the most violated cut inequality can be found by solving a minimum cut problem on G^* separating i and j . If the minimum cut capacity is less than $r_i^* + r_j^* - 1$, then there is a violated cut inequality. Thus inequalities (3.16) can be separated exactly by solving $O(n^2)$ minimum cut problems. Due to constraints

(3.13)-(3.15), a violation for (3.16) can only occur for a pair i and j for which $\{i, j\} \notin E^c$.

To speed up the separation of cut inequalities, we employ a three phase procedure. In the first two phases violations are heuristically generated and only in the last phase we resort to the more costly exact separation.

3.3.2 Connectivity Separation (CS)

In the first phase we use a heuristic that can achieve separation in $O(|E^c(R^*)|)$ time complexity. If the support graph G^* is not connected then violated cut inequalities (3.16) can be easily derived. More precisely, let S be any connected component in this graph. The capacity of the cut $(S, V \setminus S)$ in the closure graph is zero, i.e., $x^*(\delta_{G^c}(S)) = 0$ which implies that for any pair $i \in S, j \in R^* \setminus S$ for which $r_i^* + r_j^* > 1$, the cut inequality induced by S , i , and j is violated. Hence, a simple breadth first search in G^* might lead to violations. This search can further be simplified. In particular, due to constraints (3.13)-(3.15), if there exists a violated inequality induced by S , i , and j because G^* is not connected, then the same triplet will induce a violation if simply the connectivity of $G^c(R^*)$ is checked. Ultimately, in this phase of our separation algorithm, we may ignore the x variables and work simply with the closure graph which provides great savings in CPU times since the numerical challenges of working with very small positive values are eliminated. Each connected component of $G^c(R^*)$ may lead to violated cut inequalities. In our application, we chose a strong component S for which $|\delta_{G^c}(S)|$ is smallest to generate cuts.

3.3.3 Global Minimum Cut Separation (GMCS)

In this phase, our algorithm searches for violations using a heuristic based on the Stoer-Wagner global minimum cut algorithm (SW) (Stoer and Wagner [130]). SW finds the global minimum cut (a cut with the minimum capacity among all

the partitions of the node set V) and its running time is equivalent to that of a single minimum cut problem. Let (S, T) be a global minimum cut in G^* . If the cut capacity $SWC = \omega(S, T) = x^*(\delta_{G^c}(S))$ is not smaller than $\max\{r_i^* + r_j^* - 1 : i, j \in R^*, i \neq j\}$, there cannot be a violated cut and the separation procedure is terminated. Else, (S, i, j) for every $i \in S, j \in T$ for which $r_i^* + r_j^* - 1 > \omega(S, T)$ defines a violated cut inequality. Among all potential pairs, we pick ones for which the violation is largest.

3.3.4 Minimum Cut Separation (MCS)

The last stage of the algorithm is an exact separation method. In the second phase, SW finds a global minimum cut without taking r^* values into account. Perhaps another cut with larger capacity separating two nodes with higher r^* values could lead to a violation. Note that for this to happen, the two nodes should belong to the same partition in the global minimum cut (S, T) . Therefore in this phase a series of minimum cut problems are solved separating pairs of nodes i and j that are both in S or both in T . The Ford and Fulkerson maximum flow algorithm (Ahuja et al. [2]) is implemented for solving these minimum cut problems. Only the pairs i and j for which $r_i^* + r_j^* - 1 > SWC$ need to be considered, where SWC is the value of the global minimum cut found by SW.

The details of these phases are depicted in Algorithm 1. As our computational results will attest to, the high performance of this three phase approach makes the branch-and-cut application for the RLP a very successful one. Most of the time, instead of directly solving up to $O(n^2)$ minimum cut problems, we can generate enough violated inequalities by a single graph search. One nice property of the problem is that when separating integral LP solutions, i.e. when r^* and hence x^* is integral, the first two separation algorithms become also exact. In our computational studies we observed that most of the time enough violated inequalities are found in the first phase and there are very few instances for which both the heuristics fail to produce a cut. Even in those rare cases the information derived during the second phase is very useful in reducing the number of minimum

cut problems to be solved during the exact separation phase.

3.3.5 Separation Algorithm for the RLPRF

Let (r^*, x^*) be a given solution of the LP relaxation to model (3.31)-(3.39) including a subset of the cut inequalities (3.37) inducing sets R^* and E^* and graph G^* as defined in Section 3.3.1. For every q in R^* , we additionally define the set $E_q^* = \{\{i, j\} \in E_q^c | x_{ij}^* > 0\}$ and the undirected capacitated graph $G_q^* = (R^* \setminus q, E_q^*, \omega)$ with the capacity of edge e , say ω_e set to x_e^* . For a fixed node $q \in V$, a cut inequality (3.37) is defined by a node set $S \subset V$ and fixed nodes $i \in S$ and $j \notin S$. For fixed $q \in R^*$, the most violated cut inequality can be found by solving a minimum cut problem on G_q^* separating i and j . If the minimum cut capacity is less than $r_i^* + r_j^* + r_q^* - 2$, then there is a violated cut inequality. Thus inequalities (3.37) can be separated exactly by solving $O(n^3)$ minimum cut problems.

Our separation algorithm mimics that for the RLP with an additional heuristic separation phase which became possible due to the need for survivability.

3.3.6 Two-Connectivity Separation (TCS)

A necessary condition to survive from regenerator failures is to have a two-connected hub network, i.e., $G^c(R^*)$ should be two-connected. In other words, the following constraint set must be satisfied.

$$\sum_{\{k,l\} \in E^c: k \in S, l \notin S} x_{kl} \geq 2(r_i + r_j - 1) \quad S \subset V, i \in S, j \notin S$$

Note that the above inequalities will be implied by their stronger form of cut inequalities (3.37) appearing in the model. However, if $G^c(R^*)$ is not connected then violated inequalities to the above constraints can be easily generated. In particular, for any connected component S of $G^c(R^*)$, the triplet (S, i, j) where

Algorithm 1: RLP Separation

Input: G^c, x^*, r^*, G^* **Output:** Cuts

```
1 begin
2   Cuts= $\emptyset$ 
3   Apply CS on  $G^c$  to get connected components  $S_1, \dots, S_t$ 
4   if  $t > 1$  then
5     Let  $q = \operatorname{argmin}\{|\delta_{G^c}(S_i)| : i = 1, \dots, t\}$ 
6     foreach  $i, j \in R^* : i \in S_q, j \notin S_q, r_i^* + r_j^* > 1$  do
7       Cuts=Cuts  $\cup (S_q, i, j)$ 
8     end
9   end
10  Apply GMCS on  $G^*$  to get the global minimum cut  $(S, T)$ 
11  Let  $SWC = x^*(\delta_{G^c}(S))$ 
12  if  $SWC < \max\{r_i^* + r_j^* - 1 : i, j \in R^*, i \neq j\}$  then
13    Let  $s^* = \operatorname{argmax}\{r_i^* : i \in S\}$ 
14    Let  $t^* = \operatorname{argmax}\{r_i^* : i \in T\}$ 
15    if  $SWC < r_{s^*}^* + r_{t^*}^* - 1$  then
16      foreach  $t \in T : r_{s^*}^* + r_t^* - 1 > SWC$  do
17        Cuts=Cuts  $\cup (S, s^*, t)$ 
18      end
19      foreach  $s \in S : r_s^* + r_{t^*}^* - 1 > SWC$  do
20        Cuts=Cuts  $\cup (S, s, t^*)$ 
21      end
22    end
23    foreach  $i, j \in S : r_i^* + r_j^* > SWC + 1$  do
24      Apply MCS in  $G^*$  to get minimum cut  $(U, V)$  separating  $i$  and  $j$ 
25      if  $\omega(U, V) < r_i^* + r_j^* - 1$  then
26        Cuts=Cuts  $\cup (U, i, j)$ 
27      end
28    end
29    foreach  $i, j \in T : r_i^* + r_j^* > SWC + 1$  do
30      Apply MCS in  $G^*$  to get minimum cut  $(U, V)$  separating  $i$  and  $j$ 
31      if  $\omega(U, V) < r_i^* + r_j^* - 1$  then
32        Cuts=Cuts  $\cup (U, i, j)$ 
33      end
34    end
35  end
36 end
```

$r_i + r_j - 1 > 0$ induces a violation for the above constraint set. Similar to the RLP separation, in our application we chose strong components with minimum number of outgoing closure edges to generate cuts.

For every $q \in R^*$, the following three phases similar to those of RLP separation are applied to the individual closure graphs.

3.3.7 Connectivity Separation (CS_q)

If G_q^c is not connected, then for every component S , $i \in S$, $j \notin S$ such that $r_i^* + r_j^* + r_q^* - 2 > 0$ a cut inequality of type (3.37) is violated. We pick the strong component with minimum $|\delta_{G_q^c}(S)|$ value to generate all such possible violations.

3.3.8 Global Minimum Cut Separation ($GMCS_q$)

Let (S, T) be the global minimum cut in G_q^* with capacity SWC . If $SWC \geq \max\{r_i^* + r_j^* + r_q^* - 2 : i, j \in R^* \setminus q, i \neq j\}$, there cannot be any violated cuts. Otherwise, every $i \in S, j \in T$ for which $r_i^* + r_j^* + r_q^* - 2 > 0$ defines a violated cut inequality. For fixed $q \in R^*$, as done in RLP Separation, i and j are chosen such that the violation is the greatest.

3.3.9 Minimum Cut Separation (MCS_q)

If the previous heuristic phases fail to generate enough cuts, this exact phase of the separation algorithm is executed. If (S, T) is the global minimum cut with capacity SWC , minimum cuts separating pairs of nodes both in S or both in T can lead to violations. In particular, only pairs $i, j \in S(T)$ for which $r_i^* + r_j^* > SWC + 2 - r_q^*$ might generate violations.

3.3.10 Separation Algorithm for the RLPNF

Apart from choosing the fixed node q in V rather than in R^* , the phases of this separation procedure for inequalities (3.58) are identical to those of the *RLPRF*.

The pseudo codes for the *RLPRF* and *RLPNF* separation procedures are not presented due to their close resemblance to Algorithm 1.

3.3.11 Branch-and-cut Implementation Details

In order to reduce the high symmetry present in the feasible region for an RLP instance, we augment our model (3.11)-(3.18) with some optimality cuts commonly used in the location literature [43]. In particular let $N(i) = \{j : \{i, j\} \in E^c\}$ define the neighborhood of each node $i \in V$ in the closure graph G^c . Then, we have:

Lemma 1. *If $N(i) \setminus \{j\} \subseteq N(j) \setminus \{i\}$ for $i \neq j$, then there exists an optimal solution for which $r_i = 0$.*

Proof. In order for i to be present in every optimal solution, it should have a neighbor that is only covered through a hub at node i . The above condition excludes this situation. $\square$

In order to strengthen the initial LP relaxations, we also added the following valid inequality forcing the hub network to be including at least as many edges as a tree hub network.

$$\sum_{\{i,j\} \in E^c} x_{ij} \geq \sum_{i \in V} r_i - 1$$

The branch-and-cut procedure is guided by three parameters, namely, Cut-limit, Totalcutlimit, and ESnodelimit. At every node of the branch-and-cut tree with node number at most ESnodelimit, we look for violations through CS, GMCS, and MCS in that order. We stop the separation algorithm as soon as

Cutlimit number of cuts are generated. If the current branch-and-cut tree node size is larger than ESnodeLimit, we stop with exact separation and look for violations (at most Cutlimit in number) through CS and GMCS only. In order to keep the cut augmented LPs at reasonable sizes, if the number of user cuts generated in this fashion is more than Totalcutlimit, we change the Cutlimit to 1. There is no limitation on the number of CPLEX generated cuts at each node.

We use the three parameters of the RLP algorithm with the same meanings during the branch-and-cut implementations for the RLPRF and the RLPNF as well. Given a solution (x^*, r^*) , violations are first searched through TCS in G^c . Then, for each $q \in R^*$, more violations are looked for by applying CS_q in G_q^c , and $GMCS_q$ and MCS_q in G_q^* . The next phase of the separation algorithm is executed only if the number of user generated cuts so far is below Cutlimit. The specific values of the search parameters are Cutlimit=10, Totalcutlimit=5000, and ESnodeLimit=1.

We implemented our branch-and-cut algorithms using Java under Linux and ILOG CPLEX 12.4. CPLEX default settings are used throughout the branch-and-cut implementations. To integrate our algorithms with CPLEX we utilized the *UserCutCallback* and *LazyConstraintCallback* classes provided by the ILOG CPLEX 12.4. In each branch-and-bound node:

- If the LP solution is not integral, CPLEX pauses and calls the former class which runs the proposed separation algorithms to find violated inequalities. If any are found, they are added to the model and CPLEX re-solves the modified model with the added constraints. This process repeats until either an integral solution is found or separation algorithms fail to detect a violated inequality in which case branching occurs. In variable selection we set the priority levels of each binary variable $r_i, i \in N$ to the total number of edges incident to the node i and CPLEX chooses the variables with the highest priority level first. This is done by employing the *setPriority* method available in the ILOG CPLEX 12.4.

- In case of an integral solution, CPLEX pauses and calls *LazyConstraint-Callback* class to run CS algorithm (which can perform exact separation in this case) to check whether incumbent solution is feasible. If connectivity separation detects a violated inequality it is added to the model and CPLEX continues with the modified model. Otherwise the incumbent solution is treated as a feasible integral solution and the branch-and-bound node is pruned by integrality.

3.4 Numerical Experiments

Extensive computational experiments are conducted to test the performances of the proposed models and algorithms and to derive insights from the instances closely representing the real world problems. All experiments are done on a $32 \times 2.6\text{GHz}$ AMD Opteron 6282 SE processor with 50GB RAM.

3.4.1 Performances of the Models and Algorithms

There are only a few exact algorithms in the literature that can solve moderate size RLP instances. Chen et al. [27] and Lucena et al. [93] present most efficient of those algorithms. To check the performances of our algorithms against this benchmark we emulated the random graph generation procedures commonly adopted in these two studies.

3.4.1.1 Branch-and-cut Algorithms

For a fixed d_{max} value, graphs are randomly generated based on four parameters. Parameter n is the number of nodes in the instance, $p \in [0, 1]$ is the ratio of edges included in the input graph G among all potential edges. Edge lengths are chosen from a uniform distribution $U[a * d_{max}, b * d_{max}]$ where $(0 < a \leq b \leq 1)$. To ensure the feasibility for the RLPRF and the RLPNF,

an arbitrary cycle spanning all the nodes is included in the set of chosen edges. In addition to the cycle edges, $\max(0, \lfloor \frac{p(n-1)n}{2} - n \rfloor)$ more edges are added. To keep the instances consistent with the instances in Chen et al. [27], we fix $a = 0.2$ and $b = 1$, and $d_{max} = 100$. We enlarge the experimentation set of Chen et al. [27] to values of $n \in \{100, 150, 200, 250, 300\}$ and $p \in \{0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.10, 0.20\}$ and generate 5 random instances for each pair of (n, p) . Table 3.1 aggregates the results of 825 instances. Under the multicolumn heading “Graph Specifications”, for each (n, p) pair, we provide the minimum, maximum, and mean degrees in the 5 randomly generated input graphs. Under column “cl. d”, acronym for closure density, we depict the average percentage of edges present in the closure graphs of these 5 instances. Under headings “NR” and “RT”, and for each problem, the average number of regenerators in the optimal solutions as well as the average times in seconds to reach optimality are provided, respectively.

Results presented in Table 3.1 show that in general RLPRF and RLPNF take more time to solve the same instance of the problem than RLP. The behaviors of RLPRF and RLPNF are similar with slightly higher solution times for the RLPNF. Very sparse as well as very dense networks are favored for both of these problems in terms of CPU times. This is in contrast to the performance for the RLP where the most challenging instances correspond to very sparse input networks. In particular, we ran into out of memory problems for the pairs $n = 250, p = 0.01$ and $n = 300, p = 0.01$. This is in part due to the weak LP bounds of the RLP cut formulation. As networks get dense, the RLP solution times also tend to decrease dramatically. This observation is consistent with the findings in Chen et al. [27] and Lucena et al. [93]. In fact, both of these papers could report exact solutions for either very small dimensions (at most 100 nodes) or very dense networks (p values at least 0.1). The solution times reported in this chapter are comparable with the heuristic times provided in the mentioned two studies. Our RLP branch-and-cut implementation could provide results for moderately dense networks in reasonable times for dimensions up to 500 nodes. However, since our scope has all three problems and since the RLPRF and RLPNF solution times did not scale to such dimensions, we do not provide the corresponding statistics

here.

The price of survivability manifests itself by increasing the regenerator number 1.4-117% with an average of 74.5% in going from RLP to RLPRF solutions and 0-100% with an average of 8.4% in going from RLPRF to RLPNF solutions, respectively. This cost is typically at its minimum for very sparse networks and at its maximum for very dense ones. In this setting of network generation, networks with larger number of nodes are more connected as can be observed through increasing closure densities with increasing n values. This in turn results in a decrease in the average number of regenerators necessary for each problem type. Though the absolute number of regenerators decreases with increasing dimension, the relative price of survivability seems to stay close throughout the n variations.

3.4.1.2 Flow Formulations

In our testbed instances to compare the flow and cut formulations, in the random graph generation scheme detailed above, we chose p as $\delta/(n-1)$ in order to generate graphs with average degree of δ . We ranged $n \in \{40, 60, 80, 100\}$ and $\delta \in \{3, 4, 5\}$ and randomly generated 5 instances for each parameter pair. Table 3.2 depicts the results of 360 runs. As done in the previous table, we report the average closure graph densities of the 5 instances. The average optimal solution values are presented under multicolumn NR for each of the three problems. The average CPU times in seconds for respective flow and cut formulations of the three problems are also tabulated. Due to the sparse nature of the generated graphs, the performance of RLP is the weakest for some instances. In general, the cut formulations are quite fast. However, for $\delta = 4$ and $\delta = 5$, the performance of RLP flow formulations is also quite good. Even with $n = 100$ optimum solutions for the RLP flow formulations are obtained in a few seconds.

Unlike the results of Table 3.1, the number of regenerators increases with increasing dimension due to the decrease in the closure graph densities resulting from fixing the average degrees. The price of survivability tends to increase slightly with the increasing density and shows a nearly steady behavior with

Table 3.1: Cut Formulation Results

n	p	Graph Specifications				NR			RT		
		min	max	mean	cl.d.	RLP	RLPRF	RLPNF	RLP	RLPRF	RLPNF
100	0.01	2.00	2.00	2.00	2.65	78.20	80.20	80.20	1.00	0.00	0.00
	0.02	2.00	2.00	2.00	2.65	78.20	80.20	80.20	1.00	0.00	0.00
	0.03	2.00	6.00	2.98	5.20	30.20	55.80	58.20	195.60	0.60	0.00
	0.04	2.00	7.60	3.98	8.94	18.40	37.00	39.00	1.60	1.20	0.20
	0.05	2.00	9.60	4.96	13.46	12.80	25.40	27.60	1.20	3.00	0.00
	0.06	2.00	11.80	5.96	18.30	9.20	18.40	20.00	1.40	3.20	1.20
	0.07	2.60	13.40	6.94	24.23	6.80	12.40	13.80	2.60	6.60	1.80
	0.08	3.00	14.80	7.92	29.94	5.80	10.80	11.80	2.00	7.80	3.20
	0.09	3.40	16.20	8.92	35.36	4.80	8.20	8.80	4.40	10.40	3.60
	0.1	4.20	17.40	9.90	42.59	4.00	7.20	7.20	3.00	8.80	5.40
	0.2	11.00	28.20	19.80	91.82	1.20	2.20	3.00	0.00	0.60	7.40
150	0.01	2.00	2.00	2.00	1.75	119.40	121.40	121.40	1.80	1.00	0.00
	0.02	2.00	6.60	2.99	3.76	45.40	83.20	86.00	72.70	1.00	0.20
	0.03	2.00	9.60	4.48	7.56	21.60	45.00	48.40	3.60	6.80	1.00
	0.04	2.20	11.80	5.97	12.66	14.00	26.60	29.20	3.20	19.20	3.40
	0.05	2.40	14.80	7.45	18.43	9.40	16.60	18.20	5.60	38.00	8.40
	0.06	3.00	16.60	8.95	26.63	6.60	12.20	13.00	5.20	63.20	11.60
	0.07	4.40	18.80	10.44	33.98	5.00	8.80	9.80	4.00	42.60	13.20
	0.08	5.40	20.00	11.92	43.52	4.20	7.20	7.60	5.80	58.40	21.80
	0.09	6.00	22.00	13.41	52.41	3.60	5.80	6.20	8.80	40.80	28.20
	0.1	7.20	24.60	14.91	60.56	3.00	5.00	5.00	5.80	36.60	52.00
	0.2	19.00	43.20	29.80	98.70	1.00	2.00	2.40	1.80	2.20	11.20
200	0.01	2.00	2.00	2.00	1.33	157.60	159.80	159.80	3.00	1.00	1.00
	0.02	2.00	9.40	3.99	4.42	36.20	71.20	75.40	142.00	14.60	1.40
	0.03	2.00	12.40	5.98	9.30	17.60	34.60	37.40	6.40	234.60	11.40
	0.04	2.60	16.40	7.97	16.11	10.80	19.80	21.80	8.60	122.00	17.00
	0.05	3.80	18.60	9.96	25.29	7.20	13.20	14.40	9.40	157.20	50.20
	0.06	5.20	21.40	11.95	34.88	5.20	8.60	9.40	13.00	171.60	74.40
	0.07	6.60	23.60	13.93	45.39	4.00	7.00	7.20	11.80	174.00	88.40
	0.08	7.60	26.60	15.92	56.11	3.20	5.40	5.60	16.60	122.40	157.80
	0.09	9.00	29.20	17.91	65.83	2.60	4.60	4.60	10.80	95.60	185.60
	0.1	10.80	31.80	19.90	74.48	2.20	3.80	3.80	11.60	76.20	203.80
	0.2	26.00	55.00	39.80	99.91	1.00	2.00	2.00	6.80	7.20	7.00
250	0.01	2.00	5.20	2.50	1.58	mem.	171.20	174.60	mem.	1.00	1.00
	0.02	2.00	10.60	4.98	5.55	30.00	60.40	65.40	35.20	62.60	7.20
	0.03	2.80	15.40	7.47	11.81	15.20	27.20	30.00	59.20	689.00	85.60
	0.04	3.80	18.60	9.97	20.42	8.80	16.20	17.40	22.60	710.60	360.40
	0.05	5.00	22.40	12.46	31.89	6.20	11.20	11.80	20.20	323.60	126.20
	0.06	6.20	25.60	14.94	43.07	4.40	7.40	7.60	27.40	304.60	363.20
	0.07	7.60	29.60	17.43	55.85	3.20	5.80	5.80	25.80	328.80	252.00
	0.08	10.00	35.00	19.92	66.81	2.80	4.40	4.60	27.00	176.60	586.20
	0.09	11.00	36.80	22.41	76.27	2.00	3.80	3.80	21.40	195.00	402.40
	0.1	13.20	39.00	24.90	85.21	2.00	3.00	3.00	22.20	78.60	388.20
	0.2	33.00	68.20	49.80	99.98	1.00	2.00	2.00	12.20	13.80	14.20
300	0.01	2.00	6.80	2.99	1.75	mem.	166.80	172.60	mem.	6.80	2.60
	0.02	2.00	13.00	5.99	6.40	26.40	51.80	57.60	85.20	1948.80	347.80
	0.03	3.00	18.20	8.97	14.17	13.40	23.80	26.20	46.60	2355.60	3448.00
	0.04	4.00	22.20	11.97	24.99	7.80	13.40	14.80	78.20	2178.60	2975.00
	0.05	6.20	27.00	14.95	38.00	5.20	8.40	9.40	37.20	766.00	1928.20
	0.06	8.00	31.40	17.95	51.91	4.00	6.00	6.60	61.80	672.40	2336.60
	0.07	9.60	34.80	20.93	65.63	3.00	4.20	4.80	42.80	307.60	3579.80
	0.08	11.80	39.20	23.92	77.77	2.00	3.40	3.80	34.60	261.20	833.60
	0.09	13.40	41.60	26.91	86.39	2.00	3.00	3.00	167.20	54.60	55.20
	0.1	15.60	44.80	29.90	92.25	1.20	2.60	3.00	30.40	43.60	64.20
	0.2	40.80	80.60	59.80	100.00	0.40	0.80	1.60	20.20	28.80	30.20

Table 3.2: Flow vs Cut formulations

n	δ	cl. d.	NR			RT					
			RLP	RLPRF	RLPNF	RLP		RLPRF		RLPNF	
						flow	cut	flow	cut	flow	cut
40	3	13.23	12.00	23.40	23.80	0.80	0.00	4.80	0.00	4.00	0.00
	4	20.38	7.60	14.60	15.80	0.00	0.00	3.00	0.00	2.40	0.00
	5	30.41	4.80	10.00	11.00	0.00	0.00	3.80	0.20	12.00	0.00
60	3	8.82	18.40	33.80	35.20	70.00	0.80	19.40	0.00	9.40	0.00
	4	13.99	11.60	23.40	24.80	0.20	0.40	11.80	0.00	17.20	0.00
	5	20.63	7.60	15.40	17.00	0.00	0.00	28.80	0.40	30.40	0.00
80	3	6.49	23.80	45.00	46.60	418.20	1.40	50.20	0.00	69.20	0.00
	4	10.98	14.60	29.80	31.40	1.00	0.80	224.40	0.40	65.80	0.00
	5	17.01	10.00	19.80	22.20	0.20	0.60	74.80	1.20	180.40	0.20
100	3	5.20	30.20	55.80	58.20	7400.60	195.60	431.20	0.60	232.00	0.00
	4	8.94	18.40	37.00	39.00	9.40	1.60	756.00	1.20	1379.40	0.20
	5	13.46	12.80	25.40	27.60	0.20	1.20	152.20	3.00	613.00	0.00

increasing dimension.

3.4.2 Spatial Insights

For most of the real world applications of the RLP, the length of an edge between two nodes is closely related to the Euclidean distance between those nodes. Therefore we believe that investigating problem instances with edge costs computed as the Euclidean distances can be beneficial to better understand the nature of the problem and gain insights which would be helpful in dealing with the real world problems. To this end, in the following sections we study networks of this nature.

3.4.2.1 32 Node Network

In this set of problems we study a relatively small size transport network based on US haul optical network as introduced in Yetginer and Karasan [152] and depicted in Figure 3.3. In Figure 3.4, solutions (NR numbers) of the three problems for varying transport range (d_{max}) values are illustrated. As expected the larger the transport range the less the number of regenerators needed. Figure 3.4 confirms this intuition and also shows that the optimal solutions become less sensitive to

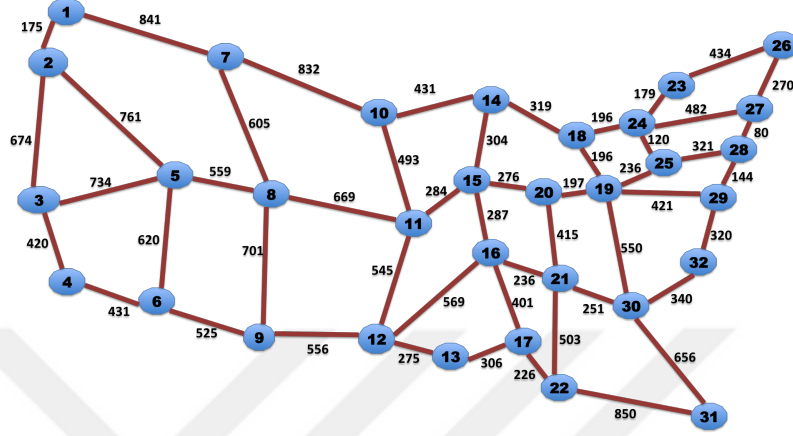


Figure 3.3: 32-Node Network

d_{max} as it gets higher. This conclusion is expected and valid for all three problem types RLP, RLPRF and RLPNF.

Confirming the conclusions of Section 3.4.1, the high price of survivability is also apparent in Figure 3.4. The average number of regenerators required for RLP, RLPRF, and RLPNF are 1.68, 3.46 and 3.84, respectively. We see that the number of regenerators can increase significantly when the network is required to be resilient against regenerator failures or resilient against node failures. On the average, the number of regenerators increases by 107 % with regenerator resilience and by 125% with node resilience. These high percentage changes are in part due to having a small number of regenerators as solutions for all three problem types in this network instance. This price is lower for more strict transport range values. However, the price of adding node survivability over regenerator survivability is on the average quite low at 8%. Our more extensive experiments presented in the next subsection also confirm all of these observations albeit with slightly smaller percentages due to the high number of regenerators in the solutions.

In Table 3.3 we list the regenerator locations for the transport range values at which RLPRF and RLPNF have different solutions. Revealing the different natures of the RLP problems, Figure 3.5 illustrates the optimal solutions for the instance $d_{max} = 1090$. Looking at these figures we can observe that RLPRF

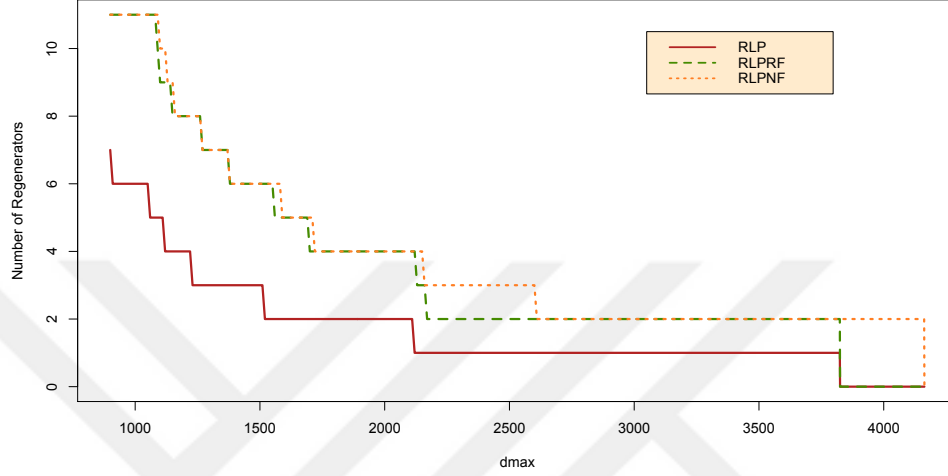
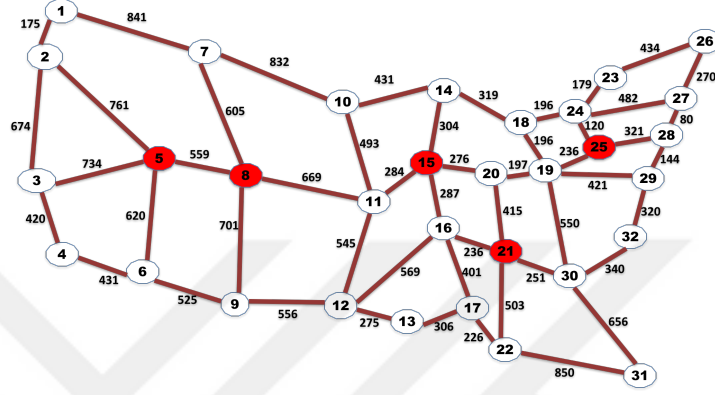


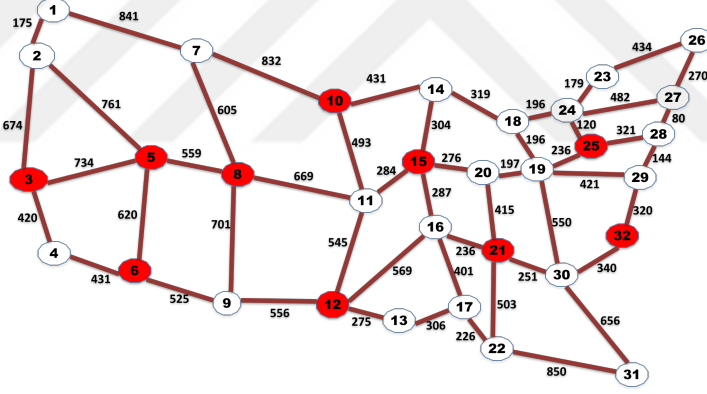
Figure 3.4: 32-Node Network: Number of Regenerators versus d_{max}

basically adds new regenerators to the RLP solution to establish a cycle among the regenerator nodes to guarantee that each node can reach at least two regenerators. In the RLPNF, not only one more regenerator is required but also the placement of those regenerators is different than that of RLPRF. Looking at node 31 we can see what makes RLPRF and RLPNF to have different solutions in this example. Far away peripheral nodes like 1 and 31 increase the NR values for the RLPNF solution. These nodes have just two neighbors and at least one of them is quite far. In this situation establishing node failure resilience forces two regenerators to be placed rather close to these nodes, which restricts the solution space significantly hence resulting in higher number of regenerators in total. Therefore for the problem instances where node degrees are small and edge lengths are close to d_{max} , it is more likely for RLPRF and RLPNF to have different solutions.

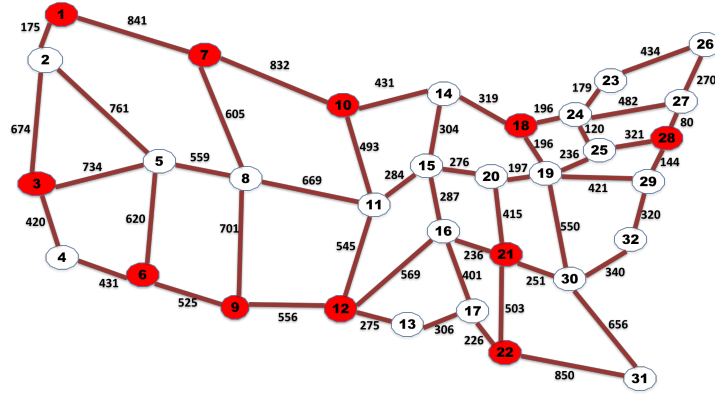
It is important to understand if any spatial property of a node makes it desirable as a hub. In this respect, the number of times a node appears in the alternative optimal solutions reveals crucial information. For four selected d_{max} values, Table 3.4 lists for each node its degree in the corresponding closure graph ($\Delta(G^c)$) and the percentage of optimal solutions having this particular node as a



(a) RLP Solution



(b) RLPRF Solution



(c) RLPNF Solution

Figure 3.5: Hub Locations (marked nodes) for $d_{max} = 1090$

Table 3.3: Computational Results for the 32-Node Network Example

d_{max}	RLP		RLPRF		RLPNF	
	NR	Regeneration Nodes	NR	Regeneration Nodes	NR	Regeneration Nodes
1090	5	5-8-15-21-25	10	3-5-6-8-10-12-15-21-25-32	11	1-3-6-7-9-10-12-18-21-22-29
1150	4	5-8-15-32	8	1-5-7-9-10-16-18-32	9	2-5-7-9-10-12-22-25-32
1560	2	5-15	5	4-5-12-15-19	6	1-6-8-12-19-21
1700	2	5-15	4	5-6-15-16	5	1-6-8-16-18
2130	1	10	3	1-11-14	4	1-9-11-32
2170	1	10	2	8-11	3	7-11-14
3825	0	-	0	-	2	8 -11

hub node for the three problem types (denoted as 1,2 and 3 to save space). The number of alternative optimal solutions is reported at the bottom of the table for each instance. Figure 3.4 can be referred to for optimal values.

Inspecting the table, we observe that peripheral nodes like 26 and 31 are consistently ruled out in all the solutions. The RLP designs are quite different from those of the survivable ones. In the RLP, the aim is to build a connected hub network covering all nodes. For this reason, the nodes on the central band $\{3, 5, 8, 11, 30\}$, which have relatively high closure graph degrees when compared with their neighbors, are more likely to be present in the optimal solution. This picture is more clear for smaller d_{max} values. When d_{max} gets larger, node 3 is excluded and in a typical solution of value 2 for $d_{max} = 1800$ node 8 pairs up with a central-east node of high closure graph degree.

On the other hand, when resilience is an issue and d_{max} values are small, the nodes on the periphery, preferably those with relatively high closure graph degrees among their neighbors, begin to appear in the solutions to build a two-connected hub network. For the most strict transport range, RLPRF and RLPNF optimal solutions replace nodes like 5 and 8 in a solution of RLP with pairs of nodes $\{6, 7\}$ and $\{9, 10\}$, respectively. But if d_{max} gets larger, the nodes in the center like 8 and 16 become advantageous in achieving two-connectivity. Even though for all the chosen d_{max} values, the RLPRF and RLPNF provide the same optimal values, the listed percentages interestingly point out differences in locations for medium d_{max} values.

We would like to emphasize that the previous findings linking spatial properties

to optimal hub locations are only pertinent to the hub covering aspect of our problems where demands as well as costs are not influential. One would expect that in a cost based hub model, the number of alternative solutions in Table 4 would diminish and hub locations of high demand and geographical advantages would appear as more robust optimal hub locations.

Table 3.4: Analysis of Alternative Solutions

Node	$d_{max} = 900$				$d_{max} = 1200$				$d_{max} = 1500$				$d_{max} = 1800$			
	$\Delta(G^c)$	1	2	3	$\Delta(G^c)$	1	2	3	$\Delta(G^c)$	1	2	3	$\Delta(G^c)$	1	2	3
1	3	0	100	100	4	0	13	13	6	10	2	0	8	0	0	0
2	3	0	0	0	5	0	27	25	7	10	16	2	7	0	0	0
3	5	100	100	100	5	0	0	0	7	14	39	49	8	0	0	0
4	2	0	0	0	5	0	27	25	6	5	5	0	9	0	0	0
5	4	100	0	0	8	100	87	88	9	38	49	49	14	24	20	0
6	4	0	100	100	6	6	47	50	8	10	21	6	13	12	100	100
7	3	0	100	100	5	0	100	100	8	5	44	51	14	0	0	0
8	4	100	0	0	7	18	0	0	17	57	49	49	23	65	80	100
9	4	0	100	100	9	76	67	63	14	29	80	94	18	0	0	0
10	5	3	100	100	13	0	100	100	20	0	13	9	26	0	0	0
11	10	100	0	0	18	12	0	0	24	24	15	9	27	18	0	0
12	8	0	50	50	13	12	33	38	17	0	5	0	23	0	0	0
13	6	0	50	50	9	0	0	0	16	0	21	11	20	0	0	0
14	11	5	0	0	18	6	0	0	23	0	59	83	25	6	20	0
15	14	16	0	0	21	12	0	0	24	26	24	0	26	12	40	0
16	12	14	0	0	19	53	40	0	23	0	10	21	26	18	100	100
17	7	0	0	0	13	12	40	75	17	0	8	13	24	0	0	0
18	14	22	100	100	17	6	20	0	21	2	11	0	24	6	20	50
19	15	22	0	0	19	12	20	0	23	17	19	36	24	6	20	50
20	14	22	0	0	20	12	20	0	23	17	23	0	25	12	0	0
21	13	19	0	0	20	0	7	0	24	26	39	38	24	6	0	0
22	7	5	100	100	12	0	13	25	18	0	9	17	22	0	0	0
23	10	5	0	0	15	0	20	0	17	0	0	0	20	0	0	0
24	11	8	0	0	17	6	0	0	18	0	0	0	23	6	0	0
25	14	19	25	0	17	6	20	100	20	2	11	26	23	6	0	0
26	8	0	0	0	12	0	0	0	14	0	0	0	17	0	0	0
27	11	14	25	50	14	0	0	0	17	0	0	0	20	0	0	0
28	11	14	25	50	14	0	0	0	18	2	4	9	20	0	0	0
29	12	14	25	0	16	6	0	0	19	2	6	13	21	0	0	0
30	14	95	100	100	21	35	47	88	22	2	11	0	24	6	0	0
31	2	0	0	0	6	0	0	0	14	0	0	0	20	0	0	0
32	9	5	0	0	16	12	53	13	20	2	8	17	22	0	0	0
#Alt		37	8	4		17	75	8		42	158	47		17	5	2

3.4.2.2 Euclidean Networks

To study Euclidean graphs further, we studied various randomly generated problem instances of different size and structure. In fact Chen et al. [27] have test instances for Euclidean networks. In that setting, to establish feasibility, an arbitrary spanning tree is constructed by drawing the lengths of the edges in the tree from a uniform distribution. In other words, $n - 1$ edges do not have Euclidean edge lengths. Moreover, with the parameters used in that study the closure graphs get too dense as the average degree of a node in the closure graph can be as large as 50. Although it is computationally easier to generate and solve the RLP on such instances, we believe that they may not be close approximations of the real world. In our case ensuring feasibility for the RLPRF and the RLPNF

would add even more artificialities. So we chose to generate Euclidean networks in a slightly different way.

We uniformly locate n nodes on an 80×80 grid. The parameter $\delta_{max} \leq d_{max}$ controls the connectivity of the graph, namely, edge $\{i, j\}$ is included if and only if the Euclidean distance between nodes i and j is less than δ_{max} . If the graph generated in this fashion is not two-connected, we disregard it and repeat the procedure with a different random number seed. It is clear that for smaller values of δ_{max} it gets harder to construct the graph and the resulting graph tends to concentrate on some smaller portion of the 80×80 grid. Thus δ_{max} cannot be chosen arbitrarily small.

Table 3.5: Computational Results For the Euclidean Networks

n	δ_{max}	RLP			RLPRF			RLPNF		
		12	19	25	12	19	25	12	19	25
	d_{max}									
80	25	10.2	8.0	7.5	15.4	12.8	12.0	15.9	13.0	12.3
	30	7.7	5.8	5.4	11.9	9.8	9.0	12.2	10.1	9.3
	35	5.8	4.1	4.0	9.4	8.0	7.9	9.6	8.2	8
100	25	9.6	7.9	7.3	14.9	12.6	11.9	16.1	13.0	12.2
	30	6.9	5.7	5.4	11.3	9.6	9.2	12	9.8	9.4
	35	5.5	4.1	4.0	8.9	7.9	7.8	9.1	8.0	7.9
120	25	9.1	7.6	7.2	14.3	12.2	11.6	14.8	12.7	12.1
	30	6.6	5.3	5.1	10.8	8.9	8.4	10.9	9.1	8.6
	35	5.1	4.1	4.0	8.5	7.9	7.7	8.5	7.9	7.7
150	25	8.2	7.3	6.9	13.3	11.7	11.5	13.7	12.0	11.9
	30	6.0	5.1	5.0	9.8	8.6	8.1	10	8.6	8.3
	35	4.3	4.0	4.0	8	7.8	7.8	8	7.8	7.8

Table 3.5 lists the solution averages of 10 instances for differing n, δ_{max} and d_{max} values. As expected, going from left to right in the table, the number of regenerators decreases due to the increased connectivity of the graphs with high δ_{max} values. This trend is persistent in all the three problems. However, it is interesting to see that the decrease is not more pronounced in the RLPRF and the RLPNF. Intuitively, one would expect that, in the case of survivability, the number of regenerators would be more sensitive to the changes in the connectivity

of the graph than in the case of RLP. However, at least for these instances, results do not confirm this expectation. For a fixed n value, we observe a sharper decrease in the optimal values with increasing d_{max} . This time we see that the solutions of the RLP are much more sensitive than the RLPRF and the RLPNF to the increasing transportation range. As a result, the relative price of survivability increases when d_{max} increases.

Table 3.6 provides further statistics for the Euclidean graphs under consideration when $d_{max} = 25$. In addition to the minimum, maximum, and mean degree averages of the 10 instances for specific n and δ_{max} pairs, the total edge length (TEL) averages are also reported. Since increased connectivity results in fewer hubs, a natural question is whether we can reduce the total hub setup cost by building new connections between nodes. Table 3.6 shows that this cannot be done in a naive way. Starting from a graph with $\delta_{max} = 12$ with 150 nodes and augmenting this graph by connecting all the nodes that are closer than d_{max} requires total edge length to grow from 5329 to 41792 (around 700% increase), while such an effort results in around 15% fewer regenerators on average without survivability and around 13% fewer when survivability is considered. However, not all those new connections are essential. Thus, it is an interesting optimization problem in its own right to choose the critical set of edges to add to an existing network such that the total construction cost is minimum while the network provides the regeneration capability of a complete network. Including edge design aspect to RLP we study this problem in more detail in Chapter 5.

In the case of change in the number of nodes in the network, two opposite effects are in place. First the higher number of nodes potentially increases the optimal value while increased connectivity pushes it down. Note that in this set of graphs the underlying grid has a fixed size and more nodes mean more connectivity. This is also apparent in Table 3.6 which shows that the average node degree almost doubles when we have 150 nodes instead of 80. This is the same for all δ_{max} values. Between these competing factors results show that the effect of connectivity is stronger than that of the number of nodes.

Table 3.6: Euclidean Network Statistics

n	$\delta_{max} = 12$				$\delta_{max} = 19$				$\delta_{max} = 25$			
	Node Degrees			TEL	Node Degrees			TEL	Node Degrees			TEL
	min	max	mean		min	max	mean		min	max	mean	
80	2.0	9.0	4.3	1515.9	3.1	19.7	10.8	5527.2	4.8	30.7	17.9	11784.2
100	2.1	10.9	5.0	2257.8	3.7	23.6	13.4	8680.6	5.9	38.2	23.0	18540.7
120	2.0	13.3	6.5	3266.9	4.4	28.4	16.7	12690.1	7.7	44.9	27.2	26541.7
150	2.3	17.3	8.8	5328.9	6.2	35.8	21.2	20051.7	10.0	57.0	34.7	41791.6

3.5 Summary

In this chapter, we revisit the regenerator location problem with the hub location viewpoint and introduce two new variations of it involving grades of survivability. For the three problems under our scope, flow formulations, cut formulations, and branch-and-cut implementations are proposed in a unifying framework. The hub location perspective enabled the formulations to be mainly defined on the hub network thus resulting in higher efficiency. The RLP literature is computationally extended and it has been shown that optimal designs under hub and node failure resilience are tractable even for very high network dimensions.

Extensive computational experiments are conducted to understand what makes some nodes desirable as hubs. It has been observed that survivability has a great price and drastically changes the resulting designs. On the other hand, interestingly, designs hedging against all node failures rather than just hub failures are not much more costly. Though the designs might have similar numbers of regenerators, their spatial properties seem to differ.

The utility of our results of this chapter are not just confined to RLP context. The problems as defined in this chapter are intrinsic parts of several telecommunications and transportation problems. As a basic problem RLP lacks some aspects of more specific relay location problems that arise in different applications but the proposed hub location perspective, notions of regenerator connectivity and closure graph transformations can be quite useful to develop efficacious solution methods for more complicated problems as we show in the following chapters.

Chapter 4

Regenerator Location Problem in Flexible Optical Networks

In this chapter we extend RLP by considering the flexible optical network architecture and relax all pairs connectivity, infinite capacity links and single modulation level assumptions to introduce regenerator location problem in flexible optical networks (RLP-FON).

As we already discussed in Chapter 1, despite its practical urgency, due to the novelty and the high complexity of the problem, there are not so many studies in the literature that address RLP-FON. Considering a static demand structure and assuming fixed routes for each transmission demand Klinkowski [76] proposes a heuristic algorithm for jointly solving spectrum allocation and regenerator location problems. Similar to our results we present in this chapter, his numerical experiments indicate that a smart placement of regenerators could significantly increase bandwidth efficiency in the network. Relaxing the fixed route assumption, Kahya [61] presents a sequential solution heuristic approach to solve routing, regenerator location and spectrum allocation problems in flexible optical networks. In this study a fixed modulation level is assumed. Motivated by the recent developments in virtualized elastic regenerator (VER) technologies that enable efficient regeneration of various bandwidth super-channels Jinno et al. [60] propose

a heuristic algorithm for calculating the minimal VER placement, routing, and the least congestion resources assignment in a translucent FON based on Nyquist wavelength-division multiplexing (WDM) super-channels. The authors assume a single modulation level and present computational results which indicate significant efficiency gains in the network due to the strategic VER placements. In a recent study, Wang et al. [142] investigate the impact of modulation conversion in FONs. The authors present an MIP formulation to solve RPP-FON. Since their formulation cannot solve realistic size problems they propose a sequential solution heuristic in which they randomly partition the demand set. Their computational study results show that benefiting from the elastic network structure, proper use of signal regenerators and wavelength converters can significantly decrease the bandwidth requirement depending on the topology of the network. To the best of our knowledge, this dissertation is the first study to present an exact algorithm to solve routing, bandwidth allocation, modulation level selection and regenerator location problems jointly for realistic size problem instances.

Path based formulations are very powerful to model problems for which the amount of cost incurred/profit gained or resource depleted depends on the routes chosen. As such, they are widely used in network design and management problems in telecommunications and transportation. Despite their advantages, path based formulations usually suffer from the exponential number of variables with only a fraction of them actually appearing in a feasible solution. Column generation and branch-and-price methods have been successfully applied to those problems to develop efficient algorithms [10–12, 30, 35, 36, 111, 112].

Within RLP-FON, for each transmission demand a routing problem is solved to find a path that connects source and sink nodes and that respects regeneration constraints. However, with a path based formulation, it is hard to address signal regeneration constraints while generating new columns in a column generation framework. Because of that we define path-segments as the simple paths on which the signal does not get into a regeneration and build the routes by the proper concatenation of those path-segments.

In this part of our dissertation we:

- Introduce RLP-FON problem that adds two new facets to RLP:
 - RLP-FON jointly solves routing, modulation level selection and regenerator location problems.
 - RLP-FON respects the bandwidth capacity limitations of the fiber links.
- Present a path-segment formulation for RLP-FON and develop a branch-and-price algorithm to solve it.
- Investigate theoretical complexities for RLP-FON and its special cases including RLP.
- Conduct extensive numerical experiments on realistic reference network topologies to test the computational performance of the proposed algorithm and to offer managerial insights. In particular, results of these experiments show that a strategic deployment of regenerators on a small portion of nodes can achieve capacity enhancements comparable to the case where all the nodes in the network have regeneration capability.

4.1 Mathematical Model

In this section we formally define RLP-FON and present the details of the proposed branch-and-price algorithm.

4.1.1 Definitions and Notation

For each connection demand, there is a certain amount of data at the origin coded into optical signals to be carried to the destination in a unit time. This coding is done with one of the technologically available modulation levels. For an optical signal, the chosen modulation level determines the number of bandwidth slots required to transfer this signal on a fiber link and sets the optical

reach, i.e., the maximum distance to be covered before a regeneration. Higher modulation levels use optical bandwidth more efficiently (require less bandwidth) but they have shorter optical reach. An optical signal is a light-path, that is a path from the source node to the destination node in the given optical network. When regenerator nodes are visited on this path, the light-path can be viewed as the concatenation of several path-segments where a path-segment is a simple path joining two consecutive regenerators on the light-path or joining a regenerator with the source or the destination node. In other words, except for the one that ends in the destination node, at the end of each path-segment there is a regenerator that restores the signal quality. Regenerators are also capable of re-coding and emitting the incoming signal with a different modulation level. i.e. regenerators have the capability to switch the modulation level of an optical signal. Each fiber link in the network has a certain bandwidth capacity which will be consumed by the light-paths passing through it. Considering all the demands simultaneously, a solution for the RLP-FON needs to respect these bandwidth capacities of fibers. Moreover, the modulation level and the path-segment chosen for a particular demand should be in harmony with respect to optical reach considerations. Thus, the solution of the problem consists of the routing decisions for each demand, location decisions of the regeneration equipment and the modulation level selections to be used for each demand on each one of its path-segments on its light-path. The objective is to find a solution that minimizes the regenerator deployment cost and obeys the link capacity and optical reach constraints.

Abiding by the common nomenclature of the text, we now specify the use of some notation already defined in 2 and define some new notation which would be specific to the solution approaches we present in this chapter.

The undirected weighted graph $G = (V, E, l)$ represents a flexible optical network instance. Note that in this chapter we consider capacitated edges and the number bandwidth slots for each edge $e \in E$ is denoted by ω_e .

We define $M = \{1, 2, \dots, \mu\}$ as the set of modulation levels and assume that

the m^{th} component of vector $\Delta = (\Delta^1, \dots, \Delta^\mu)$, is the threshold of regeneration-free communication (optical reach) for the modulation level m . In other words, two nodes with distance at most Δ^m can communicate without any need for signal regeneration using modulation m . Note that, having multiple modulation levels requires having multiple optical reach limits and thus we use the vector Δ instead of the scalar d_{max} used in the previous chapters. We assume without loss of generality that $\Delta^m \geq \Delta^{\bar{m}}, \forall m < \bar{m}$ and $l(e) \leq \Delta^1$ for every $e \in E$ since any edge violating this condition can simply be deleted from G .

Our formulation depends on the notion of *path-segments* defined in Chapter 2. To fit this definition to the specific aspects of RLP-FON we define a *path-segment* p as a directed simple path with an associated modulation level $m(p)$. Thus, by associating different modulations to the same simple path, it is possible to generate different path-segments. In our formulations we only consider path-segments with total length less than the optical reach of the associated modulation level and call such path-segments as *feasible*. More formally a path-segment p is feasible if $l(p) \leq \Delta^{m(p)}$. As defined in Chapter 2 the indicator function $I(e, p)$ returns 1 if $e \in \bar{p}$ and 0 otherwise.

For each OD pair (communication demand) in K , we define $\psi(k)$ as the data rate demanded by k . Recall that in FON, the number of bandwidth slots a demand k requires on any fiber link is a function of the modulation level chosen, say m , and the required data transfer rate of the demand $\psi(k)$. For practical purposes, it is important to note that for the same transfer rate $\psi(k)$, higher modulation levels require less bandwidth (less number of slots) but have more limited optical reach. We define $v(k, m)$ as the number of slots a path-segment of demand k occupies on each fiber optical link it traverses for a chosen modulation level m .

As a better fit to the practical context, we will call routes as light-paths in this chapter and say a light-path $P = \{p^1, p^2, \dots, p^q\}$ is feasible for a demand $k \in K$ if $o(p^1) = \mathcal{O}(k)$, $d(p^q) = \mathcal{D}(k)$, $l(p^i) \leq \Delta^{m(p^i)}, i = 1, \dots, q$, and $d(p^i)$ is a regenerator node for $i = 1, \dots, q - 1$.

A solution for RLP-FON is allowed to use only a portion $\alpha \in (0, 1]$ of the available transmission capacity (bandwidth slots) on a link. That is, the number of *slots* available on a link $e \in E$ is given by $\lfloor \omega(e) \times \alpha \rfloor$. The parameter α actually represents a managerial decision. Due to the quality of service considerations (such as uninterrupted service, accommodating unexpected demands etc.) network management does not want to use all the existing bandwidth of a link and smaller values of α are preferred. However, smaller values for α limit the data transfer capacity of the network and may increase the required number of regenerators. A more detailed discussion about this topic is presented in Section 4.3.

The formal definition of RLP-FON is as follows.

Definition 1. *Regenerator Location Problem for Flexible Optical Networks (RLP-FON): An RLP-FON instance has associated data $\langle G, K, M, l, \omega, h, v, \alpha, \eta \rangle$. The aim is to find the minimum cost regenerator deployment and signal routing such that for each $k \in K$ there is a feasible light-path P^k in G such that for all links $e \in E$, $\sum_{k \in K} \sum_{p \in P^k} I(e, p) v(k, m(p)) \leq \lfloor \omega(e) \times \alpha \rfloor$.*

4.1.2 RLP-FON Path-Segment Formulation (PS)

In this subsection we present the path-segment formulation PS for RLP-FON and provide the details of the proposed branch-and-price algorithm to solve it.

Recall that each demand k is required to follow a union of path-segments from $\mathcal{O}(k)$ to $\mathcal{D}(k)$ with a regenerator at the end of each used path-segment p for which $d(p) \neq \mathcal{D}(k)$. As such, our path-segment formulation admits a very natural representation of signal regeneration constraints. We define the following decision variables.

$$r_i = \begin{cases} 1, & \text{if node } i \in V \text{ is a regeneration point} \\ 0, & \text{otherwise,} \end{cases}$$

$$x_p^k = \begin{cases} 1, & \text{if path-segment } p \in \mathcal{P} \text{ is used by the demand } k \in K \\ 0, & \text{otherwise,} \end{cases}$$

We name $r_i, i \in V$ as the regeneration variables and $x_p^k, k \in K, p \in \mathcal{P}$ as the arc flow variables. For brevity of notation, we use $x^k(\mathcal{P}') = \sum_{p \in \mathcal{P}'} x_p^k$ for $\mathcal{P}' \subseteq \mathcal{P}$.

Our multicommodity flow formulation PS can be stated as follows.

$$\min \sum_{i \in V} h_i r_i + \sum_{\substack{k \in K \\ p \in \mathcal{P} \\ d(p) \neq \mathcal{D}(k)}} \eta x_p^d \quad (4.1)$$

$$\text{s.t. } x^k(\delta^+(i)) - x^k(\delta^-(i)) = \begin{cases} 1, & \text{if } i = \mathcal{O}(k) \\ -1, & \text{if } i = \mathcal{D}(k) \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in V, k \in K, \quad (4.2)$$

$$\sum_{k \in K} \sum_{\substack{p \in \mathcal{P}: \\ e \in \bar{p}}} v(k, m(p)) x_p^k \leq \lfloor \omega(e) \times \alpha \rfloor \quad \forall e \in E, \quad (4.3)$$

$$x^k(\delta^-(i)) \leq r_i \quad \forall k \in K, i \in V \setminus \mathcal{D}(k) \quad (4.4)$$

$$r_i \in \{0, 1\} \quad \forall i \in V, \quad (4.5)$$

$$x_p^k \in \{0, 1\} \quad \forall k \in K, p \in \mathcal{P}. \quad (4.6)$$

The objective function (4.1) has two components. The first one represents the fixed cost of setting up a regenerator site on a node. Since some of its constituents can be node dependent, regenerator placement cost h_i can take different values for different nodes $i \in V$. The second piece represents the cost of adding a new regenerator device to a regenerator site. Depending on the features of the practical setting, this cost can also represent the signal regeneration cost incurred at the end of a path-segment. Note that, with this flexible construction, the objective function is quite general accommodating the cost structure of a wide range of practical problems. In the next section we present a more detailed discussion on this topic. Constraints (4.2) are the flow balance equations that force each demand to be carried from its source to its destination by the concatenation of feasible path-segments. Constraints (4.3) are the capacity constraints which ensure that the number of slots occupied is not more than the maximum allowed. Constraints (4.4) enforce regeneration requirements by ensuring regeneration at the end of each feasible path-segment that does not end in the destination node of the associated demand. Constraints (4.5)-(4.6) are the domain restrictions. Note that this formulation is equivalent to a flow formulation on a network where

different path-segments between pairs of nodes are simply represented by parallel arcs.

Although this formulation considers a static problem where all the connection demands are known/estimated and regenerator locations are chosen accordingly, it is flexible enough to cover a more realistic case where demands emerge in an incremental manner. That is when there are a collection of flows already and we need to solve the problem to accommodate additional flows incurring minimum additional cost. In that case we can simply fix $r_i = 1$ or assume $h_i = 0$ for those regenerators that are in use (already located) to enable existing light-paths. Regarding the edge capacities in the new problem we have two alternatives. If the initial routes have to remain fixed we would reduce the number of available bandwidth slots on the edges which have already been used. On the other hand, if we have the rerouting capability there is no need to update edge capacities and we can solve the problem considering existing and new demands together by simply updating regenerator placement costs as stated above.

4.1.3 Solution Approach:

In this section we present a novel branch-and-price algorithm to solve PS. During the algorithm, column generation technique is employed to solve the linear relaxation of PS, say PS-LP, and obtain a lower bound for each node of the branch-and-bound tree.

4.1.3.1 LP Solution (Column Generation):

4.1.3.1.1 Pricing Problem: Let RPS be the restricted PS-LP formulation with a fraction of its columns. At every iteration we determine whether there exists a column with negative reduced cost such that including it in the RPS might improve the objective function. If such columns are detected we add them to the RPS and repeat the procedure until there is no column left with a negative reduced cost.

Let π_i^k represent the unrestricted dual variables associated with Constraints (4.2), and κ_e and γ_i^d be the nonnegative dual variables associated with constraints (4.3) and (4.4), respectively. For a path-segment p of demand k the reduced cost $\bar{x}_p^d$ for a fixed modulation level m is given as:

$$\bar{x}_p^d = \begin{cases} \pi_{d(p)}^k - \pi_{o(p)}^k + \sum_{e \in \bar{p}} v(k, m) \kappa_e, & \text{if } d(p) = \mathcal{D}(k) \\ \pi_{d(p)}^k - \pi_{o(p)}^k + \sum_{e \in \bar{p}} v(k, m) \kappa_e + \gamma_{d(p)}^k + \eta, & \text{if } d(p) \neq \mathcal{D}(k) \end{cases} \quad (4.7)$$

In this chapter we define the *plausible-pairs* as follows.

Definition 2. An ordered node pair $(i, j), \in N \times N$ is called a *plausible-pair* for demand k if the potential difference:

$$\pi_{(i,j)}^k = \begin{cases} \pi_j^k - \pi_i^k, & \text{if } j = \mathcal{D}(k) \\ \pi_j^k - \pi_i^k + \gamma_j^k + \eta, & \text{if } j \neq \mathcal{D}(k) \end{cases} \quad (4.8)$$

is negative. The set of all the plausible-pairs for demand k is denoted by Π^k .

In order to identify columns that price out it is required to pick out plausible-pairs (i, j) for each demand $k \in K$ and check whether there exists a path p of modulation m from node i to j with length $\sum_{e \in \bar{p}} v(k, m) \kappa_e < -\pi_{(i,j)}^k$. If the signal regeneration was not necessary, such a path could be efficiently identified by solving a shortest path problem for each modulation level $m \in M$, over a graph with arc costs equal to $v(k, m) \kappa_e$ for each arc $\bar{e}, \underline{e} \in A$. However, a path-segment p is feasible only if it satisfies signal regeneration constraint $l(p) = \sum_{e \in \bar{p}} l(e) \leq \Delta^{m(p)}$. Thus, the pricing problem actually requires to solve a number of *constrained shortest path problems (CSP)* [44]. Recall that given a directed graph with costs and resources associated with arcs, the CSP problem seeks a minimum cost path from a given source node to a given destination node with a side constraint on the total resource of the path. Our pricing problem can be solved exactly by solving a CSP instance from node i to node j on a directed graph (V, A) where the cost is $v(k, m) \kappa_e$ and the resource is $l(e)$ for each $\bar{e}, \underline{e} \in A$ and the resource limit is Δ^m . We denote this pricing graph as G_m^k .

Since the number of plausible-pairs is $O(n^2)$ and since CSP is NP-Hard [44], we propose a heuristic method (H_q) to solve the pricing problem and resort to

the exact solution of CSP, which employs the solution approach proposed by Santos et al. [121], if the heuristic method fails to produce a negative reduced cost column.

For each node pair $(i, j) \in V \times V$, paths with short lengths are *good* paths in a sense that they can support higher bit rate modulations and thus use less network resources to transmit data. Therefore, those paths are more likely to be detected as solutions of the pricing problem. Thus, it is a fruitful idea to store some limited number, say q , of those *good path-segments* and at each pricing step check those paths first before resorting to the costly solution of CSP.

Let $P_{(i,j)}^q = \{p_{(i,j)}^1, p_{(i,j)}^2, \dots, p_{(i,j)}^q\}$ be the set of q -shortest paths from node i to node j in G with non-decreasing order of lengths. For notational simplicity, we also define the cost of the path-segment $p \in P_{(i,j)}^q$ in the pricing graph G_m^k as $l_m^k(p) = \sum_{e \in \bar{p}} v(k, m) \kappa_e$. Initially, we store q -shortest paths for each node pair (i, j) and call Algorithm 2 (H_q) to detect negative reduced cost path-segments.

Algorithm 2: H_q

Input: $\langle G, K, M, P_{(i,j)}^q, \pi, \kappa, \gamma \rangle$
Output: $\langle \Omega \rangle$

```

1 begin
2   Set  $\Omega = \emptyset$ 
3   forall the  $k \in K$  do
4     forall the  $(i, j) \in \Pi^q$  do
5       Set  $m = |M|$ 
6       Set  $GoToNextPair = false$ 
7       while  $m > 0$  or  $GoToNextPair = false$  do
8         Set  $\sigma = 1$ 
9         while  $\sigma \leq q$  or  $GoToNextPair = false$  do
10          if  $\Delta^m \geq l(p_{(i,j)}^\sigma)$  and  $\pi_{(i,j)}^k + l_m^k(p_{(i,j)}^\sigma) < 0$  then
11             $\Omega = \Omega \cup \{p_{(i,j)}^\sigma\}$ 
12             $GoToNextPair = true$ .
13          end
14           $\sigma = \sigma + 1$ 
15        end
16         $m = m - 1$ 
17      end
18    end
19  end
20 end

```

If H_q for a chosen q returns $\Omega = \emptyset$, then we continue with the exact solution methodology. When solving the pricing problem after the application of H_q there is no need to consider plausible node pairs (i, j) such that $l(p_{(i,j)}^k) > \Delta^m$. Thus,

exact solution of the pricing problem requires significantly less computational effort when we first apply H_q . Note that, once we have the solution of the q-shortest path problem, which we solve just once at the very beginning, H_q requires $O(n^2|K|\mu)$ time when seeking for a negative reduced cost column among all plausible pairs, all demands $k \in K$ and modulation levels $m \in M$. So the total time complexity of the heuristic solution of the pricing problem is bounded by $O(qn(|A| + n \log n) + n^2|K|\mu)$.

4.1.3.1.2 Determining an Initial Set of Columns: Defining variables as the path-segments instead of light-paths diverts from the widely used path based formulations for which column generation technique has been applied very successfully for a wide range of problems (see Lübbecke and Desrosiers [92] for a detailed survey). Path-segments as variables necessitate a more careful approach to determine the initial variable pool. In a typical column generation algorithm it is sufficient to have a feasible solution at hand to start the procedure. However, in PS-LP, it is not enough to start with an arbitrary feasible solution. Figure 4.1 depicts a simple problem instance where there is only one level of modulation with the maximum optical reach of 1 unit. This instance contains a single data transfer demand from 1 to 4 for which just one bandwidth slot is enough to carry the signal with the available modulation. Links contain two slots and have lengths of 1 unit. All nodes have unit regenerator placement costs and $\eta = 0$. Figure (4.1b) shows an initial column pool that consists of three path-segments (numerically depicted). With these columns, one can build a feasible solution that requires two regenerators (at nodes 2 and 3). Figure (4.1c) depicts the optimal solution with one regenerator at node 5. However, starting with the three path-segments given at (b) there is no way to detect negative reduced cost columns and move to a better feasible solution. Thus, PS-LP is stuck with the initial solution and cannot obtain the optimal solution from here.

We apply Algorithm 3 to obtain the set of initial variables. Note that the data transfer capacity of the network is maximum when all the nodes have the regeneration capability and each link $e \in E$ uses the most bandwidth efficient

(highest bit rate) modulation level m^* such that $l(e) \leq \Delta^{m^*}$. Thus, if the restricted problem with the column set Ω_0 is infeasible, PS-LP is infeasible as well. Moreover, for each (k, a) pair $k \in K, a \in A$, there exists a variable x_p^k in Ω_0 with $p = a$. Thus, values of all the dual variables can be properly calculated once a solution is obtained with the variables in Ω_0 . Thus, employing Algorithm 3 we can obtain the initial set of columns in $O(\mu|A|)$ time.

Algorithm 3: Initial variable set generation

```

Output:  $\langle \Omega_0 \rangle$ 
1 begin
2   Set  $\Omega_0 = \emptyset$ 
3   forall the  $a \in A$  do
4     -find the highest bit rate modulation  $m^*$  such that  $\Delta^{m^*} \geq l(a)$ 
5     -build the single arc path-segment  $p = a$  with modulation level  $m(p) = m^*$ 
6     -Set  $\Omega_0 = \Omega_0 \cup \{x_p^k | k \in K\}$ 
7   end
8 end

```

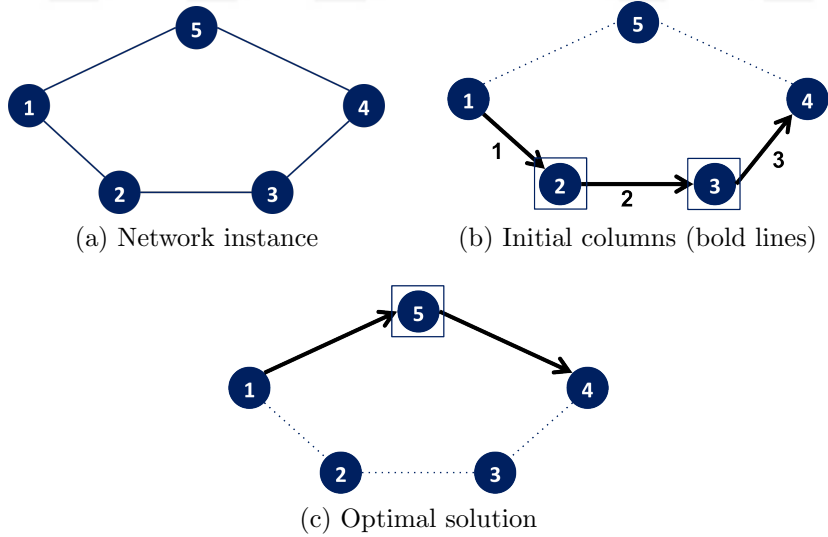


Figure 4.1: A simple example depicting the fact that initial columns matter

4.1.3.2 IP Solution

4.1.3.2.1 Branching Rules: One key step towards developing an effective branch-and-price algorithm is to identify a branching rule which eliminates the

fractional solutions but does not disrupt the special structure of the pricing problem. Keeping this in mind we propose the following two branching rules: one for the regeneration variables r_i and one for the arc flow variables x_p^k .

Branching on Regeneration Variables: Encountering a fractional solution we first detect fractional regeneration variables and branch on the variable $0 < r_i < 1$ where $i = \operatorname{argmin}_{j \in V} \{|r_j - 0.5|\}$.

Note that in formulation PS arc flow variables x_p^k are tied to the regeneration variables r_i by the constraints (4.4). Thus, branching decisions on regeneration variables affect a significant number of arc flow variables. Let r_i have a fractional value:

- **Branching-cut-1** $r_i = 0$: In this case the set of arc flow variables $\underline{X}_i = \{x_p^k | k \in K, \mathcal{D}(k) \neq i \text{ and } p \in \mathcal{P}, d(p) = i\}$ are implicitly set to 0. Thus, we must make sure that in the pricing problem any path-segment $x_p^k \in \underline{X}_i$ should not appear as a negative reduced cost column. This can be easily done by setting $\gamma_i^k = \infty \quad \forall k \in K, \mathcal{D}(k) \neq i$. Note that with this modification only the lengths of the arcs in the pricing graph are changed and the structure of the pricing problem is not affected.
- **Branching-cut-2** $r_i = 1$: Implementation of this branching cut is straightforward and does not require any change in the pricing problem.

Branching on Arc Flow Variables: Our branching rule on the arc flow variables is closely related with the one proposed by Barnhart et al. [12]. For a branching rule which is based on the arc flow variables, it is very likely that branching cuts destroy the special structure of the pricing problem. One remedy is to consider original links and base the branching decisions on the usage of an arc in A by a demand $k \in K$.

We derive our branching rule by observing that if an arc flow variable x_p^k has a fractional value, then there must exist a node $i \in V$ such that there are at least

two variables $x_{p^1}^k > 0, x_{p^2}^k > 0$ where $o(p^1) = o(p^2) = i$. We call node i as the root node. For the distinct path-segments p^1 and p^2 , starting with the *root node* and inspecting one arc at a time we can find two different arcs a^1 and a^2 where $s(a^1) = s(a^2) = \bar{i}$. The node $\bar{i}$ is called the *divergence node*. We denote the set of arcs originating from $\bar{i}$ as $A(\bar{i})$ and let $A(\bar{i}, a^1)$ and $A(\bar{i}, a^2)$ represent a partition of $A(\bar{i})$ where $A(\bar{i}, a^1)$ contains a^1 and $A(\bar{i}, a^2)$ contains a^2 . Let $\mathcal{P}(a)$ denote the set of path-segments containing arc $a \in A$. Now consider the following two sets of arc flow variables.

- $X^1 = \{x_p^k | o(p) = i, p \in \mathcal{P}(a) \text{ for some } a \in A(\bar{i}, a^1)\}$
- $X^2 = \{x_p^k | o(p) = i, p \in \mathcal{P}(a) \text{ for some } a \in A(\bar{i}, a^2)\}$

The main idea for the branching rule for the flow variables follows from the observation that in the optimal solution either arc flow variables X^1 or those of X^2 are all set to 0.

- **Branching-cut-1** $\sum_{x_p^k \in X^1} x_p^k = 0$: In this case the set of arc flow variables X^1 are set to 0. Let i be the root node and $\bar{i}$ be the divergence node. In order to force this constraint in the following pricing problems we simply remove arcs $A(\bar{i})$ from the arc set of the constrained shortest path instances $\langle G_m^k, i, j, l_m^k, l, \Delta^m \rangle \quad \forall m \in M, k \in K \text{ and } j \in V$, where G_m^k is the input graph, i is the origin node, j is the destination node, l_m^k is the cost vector and l is the resource vector. Similarly for H_q we can simply update $l_m^k(\bar{e}) = \infty, \forall \bar{e} \in A(\bar{i})$ when calculating the lengths of the paths $p \in P_{(i,j)}^q \forall j \in V$.
- **Branching-cut-2** $\sum_{x_p^k \in X^2} x_p^k = 0$: The implementation of this branching cut is analogous to the previous one.

4.1.4 Heuristic Solutions

The bulk of the columns generated in the branch-and-price algorithm is actually obtained during the column generation in the root node. Thus,

solving the problem with only those columns can provide a good heuristic for RLP-FON. We call this heuristic as H-Root and apply it to obtain an initial feasible solution to reduce the overall size of the branch-and-bound tree. Obviously a similar procedure can be applied at any given branch-and-bound node (other than the root node). Indeed, during our implementation phase, at the end of some definite intervals we pause the branch-and-price algorithm and try to find an integer solution with the columns generated so far.

4.2 Insights to Problem Complexity

In this section we investigate theoretical results about the complexity of RLP-FON and its special cases in an attempt to understand the challenges involved. We start by presenting the computational complexity of RLP-FON.

Theorem 2. *RLP-FON is NP-Hard.*

Proof. Chen et al. [27] prove that RLP is NP-Hard. The result follows from the observation that RLP is a special case of RLP-FON where:

- only a single modulation level m is considered,
- regenerator usage cost η is zero,
- between any node pair $(i, j) \in V \times V$ such that $i < j$, the set K contains a demand $k_{i,j}$ with a fixed transmission rate, i.e., we have $\sigma = v(k, m)$ for each demand $k \in K$,
- all the links $e \in E$ have capacities $\omega(e) = \frac{n(n-1)}{2}\sigma$.

□

Establishing the computational complexity of RLP-FON, the next question is to explore what makes this problem hard. For most network problems properties of the input graph is an important dimension in an answer to this question. For various well know NP-Hard problems there are polynomial time algorithms to solve them if the input graph has some special structure. However, as we show by the next theorem, this is not the case for RLP-FON which retains its computational challenge even when we consider a tree as an input graph.

Theorem 3. *RLP-FON is NP-Hard even if the input graph is a tree.*

Proof. We provide a polynomial time reduction that transforms an arbitrary 0-1 knapsack instance into an RLP-FON instance on a tree.

Consider a knapsack instance with n items of capacity W with z_i corresponding to the value and w_i corresponding to the weight of item $i \in \{1, \dots, n\}$ respectively. We construct an RLP-FON instance as follows.

- $V = \{s, \bar{s}, \bigcup_{i=1}^n t_i, \bigcup_{i=1}^n \bar{t}_i\},$
- $E = \{\{s, \bar{s}\}, \bigcup_{i=1}^n \{\bar{s}, \bar{t}_i\}, \bigcup_{i=1}^n \{t_i, \bar{t}_i\}\},$
- $l(e) = 1$ and $\omega(e) = W \ \forall e \in E,$
- $h_s = h_{\bar{s}} = \infty, h_{t_i} = 0$ and $h_{\bar{t}_i} = z_i$ for $i \in \{1, 2, \dots, n\}$, and $\eta = 0,$
- $M = \{1, 2\}$ with $\Delta^1 = 2$ and $\Delta^2 = 3,$
- $K = \{1, \dots, n\}$ where $\mathcal{O}(i) = s, \mathcal{D}(i) = t_i$ for $i \in \{1, \dots, n\},$
- $v(i, 1) = 0$ and $v(i, 2) = w_i, \forall i \in \{1, 2, \dots, n\}.$

Figure 4.2 depicts a small example of building the input graph G for a given knapsack instance.

Note that, for the given RLP-FON instance, we have two choices for each demand $i \in K$. We can either use modulation level 2 and reach to the destination

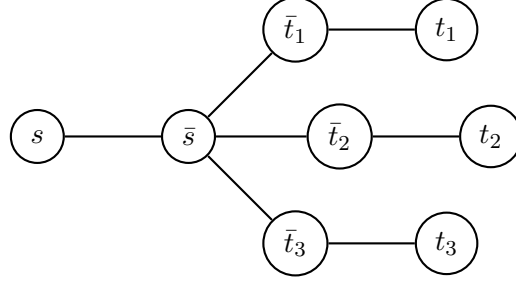


Figure 4.2: Depiction of a small transformation example with three knapsack items 1, 2 and 3.

without any regeneration but occupying w_i number of slots on the edge $\{s, \bar{s}\}$ or we can use modulation level 1 and reach to the destination by visiting a regenerator at $\bar{t}_i$ and without consuming any bandwidth on the edge $\{s, \bar{s}\}$. Observing that it is always advantageous to use modulation 2 and our freedom of using this modulation is limited with the W capacity of the bottleneck edge $\{s, \bar{s}\}$, it is easy to see the relation between the given knapsack problem and its RLP-FON transformation. Item i for $i \in \{1, \dots, n\}$ will be chosen in an optimal solution of the knapsack problem if and only if demand i in RLP-FON uses modulation level 2 in an optimal solution for the RLP-FON and RLP-FON instance will have a solution of cost at most Z if and only if the knapsack instance has a solution with value at least $\sum_{i=1}^n z_i - Z$. $\square$

The number of transmission demands is another significant dimension of the problem complexity. As we see with the following two theorems, while there is a polynomial time algorithm to solve RLP-FON when we consider a single transmission demand, RLP-FON is still a challenging problem even if we have only two transmission demands.

Theorem 4. *There is a polynomial time algorithm to solve an RLP-FON instance with $|K| = 1$.*

Proof. Let $\langle G, K, M, l, \omega, h, v, \alpha, \eta \rangle$ be an RLP-FON instance with $K = \{k\}$. For each modulation level $i \in M$ we denote its *feasible graph* $G_i = (V, E_i)$ as the subgraph of $G = (V, E)$ where $E_i = \{e \in E | v(k, i) \leq \lfloor \omega(e)\alpha \rfloor\}$. For each feasible graph G_i , consider the closure graph $G_i^c = (V, E_i^c)$ where $\{i, j\} \in E_i^c$ if and only

if the length of the shortest path from i to j in G_i is at most Δ^i . Now we define the united closure graph $G^c = (V, E^c)$ where $E^c = \bigcup_{i=1}^{\mu} E_i^c$.

Note that, we can generate feasibility graphs and solve all pairs shortest path problem on these graphs to obtain their closure graphs in polynomial time. So the generation of the united closure graph can be accomplished in polynomial time. Let each edge in E^c have cost η and node $i \in V$ have cost h_i . Solving a node weighted shortest path problem on G^c from $\mathcal{O}(k)$ to $\mathcal{D}(k)$ gives the optimal solution to the RLP-FON instance. $\square$

Theorem 5. *RLP-FON is NP-Hard even if $|K| = 2$.*

Proof. We prove this theorem by showing that a set partitioning problem can be reduced to an RLP-FON instance with two transmission demands. Let $S = \{a_1, \dots, a_{|S|}\}$ where $\sum_{i=1}^{|S|} a_i$ is even be an arbitrary instance of a set partitioning problem. We construct an RLP-FON instance as follows.

- $V = \{\bigcup_{i=1}^{|S|} \{i, \bar{i}\}, |S| + 1\}$,
- $E = \{\bigcup_{i=1}^{|S|} \{i, \bar{i}\}, \bigcup_{i=1}^{|S|} \{\bar{i}, i + 1\}, \bigcup_{i=1}^{|S|} \{i, i + 1\}\}$,
- $l(\{i, \bar{i}\}) = l(\{\bar{i}, i + 1\}) = 0$ for $i \in \{1, \dots, |S|\}$,
- $l(\{i, i + 1\}) = a_i$ for $i \in \{1, \dots, |S|\}$,
- $\omega(e) = 1$ for $e \in E$,
- $h_i = 1$ for $i \in V$, $\eta = 0$, $\alpha = 1$,
- $K = \{1, 2\}$, $\mathcal{O}(1) = \mathcal{O}(2) = 1$ and $\mathcal{D}(1) = \mathcal{D}(2) = |S| + 1$,
- $M = \{1\}$ and $\Delta^1 = \frac{\sum_{i=1}^{|S|} a_i}{2}$,
- $v(1, 1) = v(2, 1) = 1$.

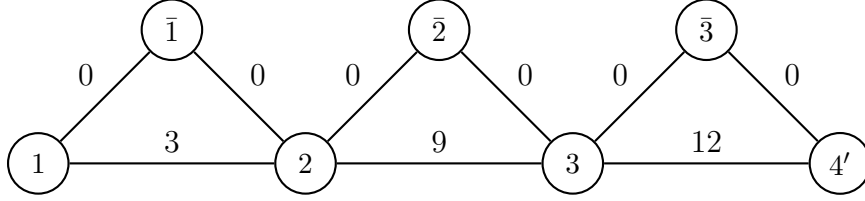


Figure 4.3: Depiction of a small transformation example for $S = \{3, 9, 12\}$.

In Figure 4.3 we present a small example of transforming a set partitioning problem into an RLP-FON instance with the desired properties.

Now observe that any solution to the given RLP-FON instance will construct two edge disjoint light-paths say P_1 and P_2 for each connection demand in K . Let $S_1 = \{a_i \in S : (i, \bar{i}) \in P_1\}$ and $S_2 = S \setminus S_1$. Then the given set partitioning instance has a solution if and only if RLP-FON has a zero cost solution. $\square$

4.2.1 Uncapacitated Edges

Obviously, having capacity limits for the edges increases the difficulty of RLP-FON. Now we investigate the case in which these constraints are relaxed. In many studies, motivated by different practical applications, uncapacitated edges are considered for regenerator/relay/refueling station placement [17, 27, 28, 40, 67, 80, 110, 136, 152, 154, 155]. We denote this problem as RLP-FON-U. Note that, once the edge capacities are relaxed, we can find the optimal routing by using only the lowest modulation level that can transmit signals furthest since occupying more bandwidth is not an issue when the capacity limits are not considered for the edges. However, such a simplification does not make the problem an easy one. Indeed RLP is a special case of RLP-FON-U in which all nodes are required to communicate with each other and the regenerator usage cost η is assumed to be zero. Thus RLP-FON-U is also NP-Hard.

Depending on the application area, one of the two costs: regenerator placement or usage costs, would dominate the overall cost. Now we first consider the case where the regenerator placement costs are negligible and show that there exists

a polynomial time algorithm to solve this special case.

Theorem 6. *There exists a polynomial time algorithm to solve an instance $\langle G, K, l, h, \eta \rangle$ of RLP-FON-U where $h_i = 0$ for each $i \in V$.*

Proof. By solving an all pairs shortest path problem on G and considering the reach limit Δ^1 we can easily generate a closure graph with edge lengths all equal to regeneration cost η . Then solving the RLP-FON-U instance entails finding the shortest path from $\mathcal{O}(k)$ to $\mathcal{D}(k)$ on this closure graph for each demand $k \in K$. $\square$

The second case we investigate is the one for which the regenerator placement costs dominate the total cost. Unlike the previous case, this time RLP-FON-U remains an NP-Hard problem. So we consider special network topologies, i.e., path and ring networks, for which we can present interesting characterizations for the optimal solutions and attain polynomial time algorithms.

Considering a special case of RLP-FON-U by assuming unit costs for the regenerator placement, unit edge lengths and fixed routes for OD pairs, Flammini et al. [40] show that there exist polynomial time algorithms to solve path (line) and ring network topologies. Now we show that these results can be extended even when unit cost, unit edge lengths and fixed route assumptions are relaxed.

Although path networks are seldom in real world applications, investigating properties of the optimal solutions for these networks can be quite useful to develop solution algorithms for general networks. Consider an RLP-FON-U instance $\langle G, K, l, h, \eta \rangle$ where the input graph $G = (V, E)$ is a path (line). Assume without loss of generality that the nodes are labelled from 1 to n such that $\{i, j\} \in E$ if and only if $|i - j| = 1$. For $i \leq j$, let $[i, j] = \{q \in V | q \leq j \text{ and } i \leq q\}$ be the set of nodes lying in the interval from i through j in this ordering. Since G is undirected, without loss of generality we may assume that $\mathcal{O}(k) < \mathcal{D}(k)$ with respect to this ordering for each $k \in K$.

Lemma 2. *Consider an RLP-FON-U instance $\langle G, K, l, h, \eta \rangle$ where $\eta = 0$ and*

$K = \{1, 2\}$. If $[\mathcal{O}(2), \mathcal{D}(2)] \subset [\mathcal{O}(1), \mathcal{D}(1)]$, then solving the problem for only $K = \{1\}$ provides the optimal solution for $K = \{1, 2\}$.

Proof. It is clear that a subset of the regenerator locations enabling a feasible light-path from $\mathcal{O}(1)$ to $\mathcal{D}(1)$ with Δ^1 reach limitation will enable a feasible light-path from $\mathcal{O}(2)$ to $\mathcal{D}(2)$ as well. $\square$

Lemma 3. Consider an RLP-FON-U instance $\langle G, K, l, h, \eta \rangle$ where $\eta = 0$ and $K = \{1, 2\}$. If $[\mathcal{O}(1), \mathcal{D}(1)] \cap [\mathcal{O}(2), \mathcal{D}(2)] = \emptyset$, then it is possible to attain an optimal solution by solving the single demand subproblems individually.

Proof. Let R_1 and R_2 be the regenerator locations when solving the given RLP-FON-U instance with $K = \{1\}$ and $K = \{2\}$, respectively. It is clear that $R_1 \cup R_2$ will be an optimal solution for $K = \{1, 2\}$. $\square$

Lemma 4. Consider an RLP-FON-U instance $\langle G, K, l, h, \eta \rangle$ where $\eta = 0$ and $K = \{1, 2\}$. Assume $[\mathcal{O}(1), \mathcal{D}(1)] \cap [\mathcal{O}(2), \mathcal{D}(2)] \neq \emptyset$, $[\mathcal{O}(1), \mathcal{D}(1)] \not\subset [\mathcal{O}(2), \mathcal{D}(2)]$, $[\mathcal{O}(2), \mathcal{D}(2)] \not\subset [\mathcal{O}(1), \mathcal{D}(1)]$ and $\mathcal{O}(1) < \mathcal{O}(2)$. Then one of the following must hold.

- Solving the single demand subproblems individually will provide an optimal solution for the original problem,
- Solving the single demand subproblem from $\mathcal{O}(1)$ to $\mathcal{D}(2)$ will provide an optimal solution for the original problem.

Proof. Let R be an optimal solution of the RLP-FON-U instance. There are two cases to consider.

- **Case-1:** There are two feasible light-paths connecting $\mathcal{O}(1)$ to $\mathcal{D}(1)$ and $\mathcal{O}(2)$ to $\mathcal{D}(2)$ visiting nonintersecting sets of regenerators. In this case it is clear that the first claim holds.

- **Case-2:** Every pair of light-paths connecting $\mathcal{O}(1)$ to $\mathcal{D}(1)$ and $\mathcal{O}(2)$ to $\mathcal{D}(2)$ in an optimal solution share at least one regenerator. Let r be such a shared regenerator. It is clear that nodes $\mathcal{O}(1), \mathcal{O}(2), \mathcal{D}(1)$ and $\mathcal{D}(2)$ all have feasible light-paths to connect to r . Let π_1 and π_2 be the light-paths through which nodes $\mathcal{O}(1)$ and $\mathcal{D}(2)$ connect to r . Then the light-path $\pi = (\pi_1, \pi_2)$ is a feasible one that connects $\mathcal{O}(1)$ to $\mathcal{D}(2)$ as desired. By Lemma 2, the regenerator locations on π provide feasible light-paths for both demands. $\square$

$\square$

Theorem 7. *There is a polynomial time algorithm to solve an RLP-FON-U instance $\langle G, K, l, h, \eta \rangle$, if $\eta = 0$ and the input graph G is a path.*

Proof. Without loss of generality assume that the demand set cannot be partitioned into nonoverlapping intervals, otherwise one can solve the problem by solving the disjoint intervals independently as suggested by Lemma 3. In a similar fashion, assume that $[\mathcal{O}(k), \mathcal{D}(k)] \not\subset [\mathcal{O}(k'), \mathcal{D}(k')]$ for distinct $k, k' \in K$ since otherwise demand k can be ignored without loss of generality using Lemma 2. Assume the demands are ordered such that $\mathcal{O}(1) < \mathcal{O}(2) < \dots < \mathcal{O}(\delta)$.

Let $Z(i), i = 1, 2, \dots, \delta$ be the optimal solution value of the RLP-FON-U instance $\langle G_i, K_i, l, h, \eta \rangle$ where $G_i = [\mathcal{O}(\delta - i + 1), \mathcal{D}(\delta)]$ and $K_i = K \setminus \bigcup_{j=1}^{\delta-i} \{j\}$, i.e., $Z(i)$ is the optimal solution of the problem considering only the last i demands in K . In particular $Z(\delta)$ is the optimal solution value for $\langle G, K, l, h, \eta \rangle$. We also define $\bar{Z}(i)$ as the optimal solution value of the RLP-FON-U instance considering the single OD pair $(\mathcal{O}(1), \mathcal{D}(i))$. Finally, let $\mathcal{D}(i^*) \in \{\mathcal{D}(1), \mathcal{D}(2), \dots, \mathcal{D}(n)\}$ be the largest index node that $\mathcal{O}(1)$ would have a feasible light-path in an optimal solution of $\langle G, K, l, h, \eta \rangle$. Now observe that for $\langle G, K, l, h, \eta \rangle$ adding an artificial demand $k^* = (\mathcal{O}(1), \mathcal{D}(i^*))$ to K does not change the optimal solution and at the presence of k^* we can disregard demands $i \leq i^*$ since they are all dominated by it. Moreover, $\mathcal{O}(1)$ cannot reach nodes $\mathcal{D}(i), i > i^*$ in an optimal solution implies that, none of the demands $i > i^*$ uses a regenerator that could be reached by

$\mathcal{O}(1)$ via a feasible light-path since otherwise $\mathcal{O}(1)$ could reach $\mathcal{D}(i)$ first reaching the shared regenerator. So we have:

$$Z(\delta) = \begin{cases} \bar{Z}(i^*) + Z(\delta - i^*), & \text{if } i^* < \delta \\ \bar{Z}(\delta), & \text{if } i^* = \delta \end{cases} \quad (4.9)$$

Since we have only δ possible values for i^* and $\bar{Z}(1), Z(1)$ can both be calculated in polynomial time by Theorem 4, a dynamic programming algorithm can find $Z(\delta)$ in polynomial time by recursively calculating $Z(i)$ for each $i < \delta$. $\square$

In telecommunications, ring networks are pervasive as cost effective and easy to implement solutions to protect network traffic against edge and node failures [138]. As minimal cycles, they have interesting properties from theoretical perspective as well. So we pay a special attention to these networks and present two important results regarding the RLP-FON-U.

Let $G = (V, E)$ represent a ring. For a pair of nodes a and b on this ring, let the interval $[a, b]$ depict the path starting from a and traversing the ring in a clockwise manner to reach node b . Let $\langle G, K, l, h, \eta \rangle$ be an RLP-FON-U instance where $\eta = 0$, the input graph G is a ring and R is the set of regenerator locations in an optimal solution. Consider the graph $T = (R, \bar{E})$ where $\bar{E} = \{\{i, j\} : i \text{ and } j \text{ are two consecutive nodes of } R \text{ on } G \text{ with clockwise distance at most } \Delta^1\}$.

Proposition 1. *The graph T is a forest.*

Proof. Assume to the contrary that $\bar{E}$ includes a cycle. Let $r \in R$ be a regenerator node. Let r_1 and r_2 be the neighbors of r in the clockwise and counter-clockwise directions on the ring. Any node i in V which communicates through r will communicate through either r_1 or r_2 , thus removal of r will not spoil feasibility which contradicts the optimality of R . $\square$

Theorem 8. *There is a polynomial time algorithm to solve an RLP-FON-U instance $\langle G, K, l, h, \eta \rangle$ when the input graph G is a ring and η is zero.*

Proof. For the sake of simplicity and without loss of generality we assume that there are at least four regenerator locations in any optimal solution. Note that

if this is not the case one can find an optimal solution by a simple enumeration since the feasibility of a given regenerator set can be checked in polynomial time. We also assume that there is no OD pair in K with shortest path distance less than or equal to Δ^1 since it could simply be omitted from K without loss of generality. Let Y^* be the value of the optimal solution for the problem instance $\langle G, K, l, h, \eta \rangle$ and $\mathcal{R}$ be the collection of sets of feasible regenerator locations for this instance.

Let $i, j \in V$ be such that the length of the interval $[i, j]$ denoted as $l_{[i,j]}$ is larger than Δ^1 . We name nodes i and j as detachment and attachment nodes, respectively. Considering the demand set $K_{[i,j]} = \{(i, j)\}$ with the single OD pair (i, j) and the path graph $G_{[i,j]}$ derived from G by removing all the edges on $[i, j]$ we obtain the reduced problem instance $\langle G_{[i,j]}, K_{[i,j]}, l, h, \eta \rangle$ for which we denote the optimal regenerator locations as $R_{[i,j]}$. We define:

$$\bar{Y}_{[i,j]} = \begin{cases} \sum_{i \in R_{[i,j]}} h_i, & \text{if } R_{[i,j]} \in \mathcal{R} \\ \infty & \text{otherwise} \end{cases} \quad (4.10)$$

Note that if T is a tree, then $Y^* = \min_{i,j \in V} \bar{Y}_{[i,j]}$.

If T is a forest with more than one disconnected trees, then there exist two sets of detachment-attachment nodes $(i, j), (l, m)$ that we can visit i, j, l, m when traversing G in the clockwise direction. Let $\langle G_{[i,j]}, K_{[i,j][l,m]}^1, l, h, \eta \rangle$ and $\langle G_{[i,j]}, K_{[i,j][l,m]}^2, l, h, \eta \rangle$ be two RLP-FON-U instances where the demand set $K_{[i,j][l,m]}^q, q = 1, 2$ initially set to K is updated as follows. For an OD pair $(s, t), s \in [i, j]$, if (s, t) can reach each other using i, j, l or m as a regenerator we remove it from $K_{[i,j][l,m]}^q, q = 1, 2$. If this is not the case we proceed as follows.

- If s can reach only one of the regenerators, say i , we replace (s, t) with (i, t) in $K_{[i,j][l,m]}^q, q = 1, 2$
- If s can reach both i and j ,
 - if $t \in [m, i]$ we replace the OD pair (s, t) with (i, t) in $K_{[i,j][l,m]}^q, q = 1, 2$ (similarly we replace it with (j, t) if $t \in [j, l]$).

- If $t \notin [m, i]$ and:
 - * it can reach only one of the regenerators, say m , we replace (s, t) with (i, m) in $K_{[i,j][l,m]}^q$, $q = 1, 2$
 - * it can reach both regenerators l and m we replace (s, t) with (i, m) and (j, l) in $K_{[i,j][l,m]}^1$ and $K_{[i,j][l,m]}^2$, respectively.

Now let $R_{[i,j][l,m]}^q$ be the optimal regenerator locations for the problem instances $\langle G_{[i,j]}, K_{[i,j][l,m]}^q, l, h, \eta \rangle$, $q = 1, 2$, respectively. We define:

$$\bar{Y}_{[i,j][l,m]}^q = \begin{cases} \sum_{i \in R_{[i,j][l,m]}^q} h_i, & \text{if } R_{[i,j][l,m]}^q \in \mathcal{R} \\ \infty & \text{otherwise} \end{cases} \quad (4.11)$$

Let $\bar{Y}_{[i,j][l,m]} = \min_{q=1,2} \bar{Y}_{[i,j][l,m]}^q$. Then it is clear that if T is a forest with more than one tree then we have $Y^* = \min\{\bar{Y}_{[i,j][l,m]} | i, j, l, m \in V, l_{[i,j]} > \Delta^1 \text{ and } l_{[l,m]} > \Delta^1\}$.

By Proposition 1, it must be the case that T is a single tree or a forest with more than one disconnected trees. We have shown that for each case we can enumerate a polynomially bounded number of RLP-FON-U instances on path networks, $O(n^2)$ for the single tree case and $O(n^4)$ for the forest case, and find the optimal solution. Hence the result follows by Theorem 7. $\square$

4.2.2 Computational Performance of the Branch-and-Price Algorithm for Special Problem Instances

In this subsection we investigate the computational performance of our branch-and-price algorithm on some special problem instances. In the first part we present some problem instances for which we can put a theoretical performance guarantee for our pricing algorithm and in the second part we propose a useful optimality cut to improve the PS formulation when regenerator usage costs are assumed to be negligible.

4.2.2.1 Ring Networks

In many practical settings ring networks are quite common in telecommunications. For these networks our H_q heuristic can solve the pricing problem exactly when $q = 2$. This is simply due to the fact that in these networks there could be at most two different simple paths that connect any two nodes. Since the computational complexity of our heuristic solution approach is $O(qn(|A| + n \log n) + n^2|K|\mu)$, we can solve the pricing problem in polynomial time. Obviously, for any network where the number of feasible path-segments between any two nodes is bounded by some positive integer Q (such as path and tree networks), H_Q can solve the pricing problem exactly in polynomial time. Moreover, since we solve the Q-shortest path problem just once in the beginning, such a computational efficiency can boost the performance of the branch-and-price algorithm significantly.

4.2.2.2 Networks with Equal Edge Lengths

Optical or electrical signals do not lose their quality just traveling long distances, they also deteriorate when they pass through a switch, router or any other network device represented as nodes. So especially when the distances between the network nodes are not large, the main concern becomes the number of nodes a signal visits instead of the total distance it travels. Considering this situation there are a wide stream of RLP literature which consider the number of hops as the reach constraint instead of the distance travelled [26, 52, 54, 94, 110, 117, 158]. Note that considering such hop constraints is equivalent to having equal edge lengths in the input graph and using the reach limit as usual. For this case solving the pricing problem can be accomplished in polynomial time by iterating the Bellman-Ford shortest path algorithm on the pricing graph as many times as the allowed number of hops.

4.2.2.3 Strengthening the PS Formulation:

In this part, we present a logical cut to tighten the LP lower bound and improve the strength of the branching cuts specifically for r variables when the regenerator usage cost η is assumed to be zero.

Definition 3. A node $i \in V$ is called an internal node of path-segment p if it is visited by p and it is neither the source nor the destination node of p . The set of path-segments that contain a node i as an internal node is denoted by $\mathcal{P}(i)$.

Proposition 2. Let (x, r) be a feasible solution of an RLP-FON instance. If there exists a node $i \in V$ for which $r_i = 1$ and i is an internal node of a path-segment p such that $x_p^k = 1$ for some $k \in K$, then there exists an alternative feasible solution $(\bar{x}, r)$ which satisfies the following conditions.

- i is not an internal node of any path-segment p that satisfies $\bar{x}_p^k = 1$ for some $k \in K$,
- for each arc $\bar{e} \in A$ the number of slots occupied by the solution $(\bar{x}, r)$ is less than or equal to that of (x, r) .

Proof. Let (x, r) be a feasible solution of an RLP-FON instance and assume $r_i = 1$ and i is an internal node of a path-segment $p = (p_1, p_2)$ where $d(p_1) = o(p_2) = i$. Let $\bar{K} = \{k \in K | x_p^k = 1\}$. Assume $\bar{K}$ is not empty.

Since $l(p) > l(p^1)$ and $l(p) > l(p^2)$ we can choose $m(p^1) \geq m(p)$ and $m(p^2) \geq m(p)$. Now we modify x by setting $x_p^k = 0$ and $x_{p^1}^k = x_{p^2}^k = 1 \quad \forall k \in \bar{K}$ and obtain the vector $\bar{x}$. By our assumption we have $r_i = 1$ and $r_{d(p)} = 1$. Thus, replacing x with $\bar{x}$ does not necessitate a change in the number and locations of the regenerators. The same will be true if $d(p) = \mathcal{D}(k)$ holds as well. Moreover, $m(p^1) \geq m(p)$ and $m(p^2) \geq m(p)$ implies that p^1 and p^2 utilize optical bandwidth more efficiently and the amount of bandwidth slots used by $(\bar{x}, r)$ is less than or equal to the one used by (x, r) . Since i was arbitrary and this procedure can be repeated as many times as needed the result follows. $\square$

By Proposition 2, the following is an optimality cut for PS.

$$\sum_{\substack{k \in K, \\ p \in \mathcal{P}(i)}} x_p^k \leq Q(1 - r_i) \quad \forall i \in V, \quad (4.12)$$

where Q is a large number.

The proposed cut forces that if a node $i \in V$ is chosen as a regeneration node (i.e. $r_i = 1$), none of the path-segments utilized by a positive x variable should contain i as an internal node. The modified formulation containing (4.12) is denoted $\overline{PS}$. Note that, denoting the highest capacity edge as e^* , choosing $Q = \lfloor \omega(e^*) \times \alpha \rfloor$ is sufficient to assure the validity of the cut, since each path-segment x_p^k occupies at least 1 bandwidth slot.

In a branch-and-price framework, adding cuts to the model requires special attention since maintaining the structure of the pricing problem is crucial to retain tractability of the solution approach. In our case, we can easily modify the reduced cost calculations and preserve the special structure of the pricing problem as follows.

Let $\theta_i, i \in V$ be the dual variables associated with the constraints (4.12) in $\overline{PS}$ and p^o denote the set of internal nodes of a path-segment p . With the following modifications, solution of the pricing problem for $\overline{PS}$ follows exactly the same steps explained above.

- The reduced cost calculation (6.11) is changed as:

$$\bar{x}_p^d = \begin{cases} \pi_{d(p)}^k - \pi_{o(p)}^k + \sum_{e \in \bar{p}} v(k, m) \kappa_e + \sum_{i \in p^o} \theta_i, & \text{if } d(p) = \mathcal{D}(k) \\ \pi_{d(p)}^k - \pi_{o(p)}^k + \sum_{e \in \bar{p}} v(k, m) \kappa_e + \sum_{i \in p^o} \theta_i + \gamma_{d(p)}^k, & \text{if } d(p) \neq \mathcal{D}(k) \end{cases} \quad (4.13)$$

- The length function for the pricing graph $l_m^k(e)$ modified as:

$$l_m^k(\bar{e}) = \begin{cases} v(k, m) \kappa_e + \theta_i, & \text{if } t(\bar{e}) \in p^o \\ v(k, m) \kappa_e & \text{o.w.} \end{cases} \quad (4.14)$$

The main function of the cut (4.12) is not to raise the LP bound in the root node, it helps to speed up the solution procedure by reinforcing the strength of the branching cuts that remove the fractional solutions for the regeneration variables r . This is because when a variable r_i is set to its upper bound 1, variables $x_d^p, \forall k \in K$ and $p \in \mathcal{P}(i)$ are all forced to 0 with the presence of the constraints (4.12). As such, we need to modify our branching rule for the Branching-cut-2 as follows.

- **Branching-cut-2** $r_i = 1$: In this case the set of arc flow variables $\overline{X}_i = \{x_p^k | k \in K, p \in \mathcal{P} \text{ and } i \in p^o\}$ are implicitly set to 0. In order to make sure that any path-segment $x_p^k \in \overline{X}_i$ would not appear as a solution of the pricing problem we can simply remove the node i from all the pricing graphs where i is neither the source nor the sink node. Similarly for H_q we can update $l_m^k(\bar{e}) = \infty, \forall \bar{e} \in \{a \in A | o(a) = i, \mathcal{D}(k) \neq i \text{ and } \mathcal{O}(k) \neq i\}$ and implement the algorithm without any change.

4.3 Numerical Experiments

Extensive numerical experiments are conducted in order to both test the performance of the proposed solution methodology and derive insights from the instances closely representing the real world problems. We implemented the branch-and-price algorithm using Java under Linux and CPLEX 12.4. All experiments are done on an AMD Opteron(tm) Processor 6282 SE machine with 2GB RAM.

4.3.1 Network and Traffic

For our experiments we studied two well known network topologies from the literature: NSF-US network (Figure 4.4a) and COST-266 Pan European network (Figure 4.4b) [55]. Table 4.1 presents the topological parameters of these networks. In this table, for each optical network, we denote the minimum (min),



maximum (max) and average (mean) values for the node degrees and physical edge lengths in kilometers.

90

Table 4.1: Topological parameters

Network	#nodes	#edges	Node Degree			Edge Length (km)		
			min	max	mean	min	max	mean
NSF	14	21	2	4	3	312	3408	1299.1
COST-266	28	41	2	5	2.9	218	1500	625.4

traffic volume as reported in Hulsermann et al. [55].

Abiding by the general approach in the RLP literature [27, 28, 37, 67, 73, 110, 152, 154], we assume $\eta = 0$ and consider unit cost for the regenerator placement. There are mainly two practical reasons for such an approach. The first one is the high set up and maintenance costs of regeneration sites for OEO regenerators [142, 150]. Thus, placement and operation of such a regenerator site is much more costly when compared with the addition of an extra regenerator slot in an existing site. Secondly, once a regenerator site is deployed on a node, it is very costly to relocate it. Thus, as a strategic level problem, RLP is more concerned about the locations of the regenerator sites rather than the number of regenerator devices in each site. Moreover, from the algorithmic perspective as we have shown in the previous section, the problem gets harder under the cost structure when the regenerator placement costs are dominant.

4.3.2 Fiber and Modulation Level Parameters

Cables are assumed to be non-zero dispersion-shifted fiber (NZDSF) and four modulation formats are considered: BPSK, QPSK, 8-QAM and 16-QAM. The number of frequency slots per fiber is 360 and Table 4.2 shows the optical reaches (Δ) and the number of slots required by each modulation format (MF) [15, 75].

We generated problem instances with 6 different α values from the set $C = \{\alpha_{min}, 0.2, 0.4, 0.6, 0.8, 1\}$ where the value α_{min} represents the smallest α value for which the problem is feasible with the given set of parameters. Since the problem has no solution for $\alpha < \alpha_{min}$, for each problem setting we studied $\alpha \in C$ such

that $\alpha \geq \alpha_{min}$.

There are two main motivations to generate problem instances with the minimum bandwidth allocation (i.e. $\alpha = \alpha_{min}$). From the algorithmic point of view, as our numerical results will attest to, problem instances with the limited arc capacities are harder to solve and those hard problem instances are required for a comprehensive performance test of algorithmic efficiency. On the other hand, from the managerial perspective, α_{min} constitutes an upper bound on the spectrum efficiency in a network. Thus, it is interesting to solve problem instances with the minimum α values to see the trade-off between the number of regenerators and bandwidth utilization efficiency. Indeed, finding the α_{min} value for each problem setting is an optimization problem in its own right. Thus, we solve the following mixed integer program for finding the minimum α value for which the problem stays feasible when deploying regenerators at each node of the network.

$$\min \alpha_{min} \quad (4.15)$$

$$\text{s.t.} \quad \sum_{\substack{\bar{e} \in A: \\ o(\bar{e})=i}} x_{\bar{e}}^k - \sum_{\substack{\bar{e} \in A: \\ d(\bar{e})=i}} x_{\bar{e}}^k = \begin{cases} 1, & \text{if } i = \mathcal{O}(k) \\ -1, & \text{if } i = \mathcal{D}(k) \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in V, k \in K, \quad (4.16)$$

$$\sum_{k \in K} v(k, m_{\bar{e}}^*) (x_{\bar{e}}^k + x_{\underline{e}}^k) \leq \omega(e) \times \alpha_{min} \quad \forall e \in E, \quad (4.17)$$

$$x_{\bar{e}}^k \in \{0, 1\} \quad \forall k \in K, \bar{e} \in A \quad (4.18)$$

The above formulation assumes that each node in the network has the regeneration capacity. The objective is to minimize the α_{min} value for the given problem instance. Constraints (4.16) are the flow balance equations that force each demand to be carried from its source to its destination. The decision variable $x_{\bar{e}}^k$ attains the value 1 if the route for the demand $k \in K$ includes the arc $\bar{e} \in A$ and 0 otherwise. Constraints (4.17) are the capacity constraints which ensure that the total number of slots occupied is not more than the maximum allowed. For each arc $\bar{e} \in A$, the best modulation $m_{\bar{e}}^*$ is defined as the highest modulation level which has an optical reach larger than or equal to the length of the arc. We have $m_{\bar{e}}^* = m_{\underline{e}}^*$ since $l(\bar{e}) = l(\underline{e}) = l(e)$. Also note that $m_{\bar{e}}^*$ is indeed the most spectrum efficient feasible modulation since higher modulation levels can transmit more

Table 4.2: Modulation level parameters

Modulation Format (m)	$\Delta^m(\text{km})$	DTR(Gbps)			
		10	40	100	400
BPSK (1)	2880	2	4	8	32
QPSK (2)	1080	2	2	4	16
8-QAM (3)	630	2	2	4	12
16-QAM (4)	270	2	2	2	8

data with less number of bandwidth slots. Constraints (4.18) are the variable restrictions.

4.3.3 Implementation Details

Before presenting the results of the numerical experiments we briefly state the implementation details of the proposed algorithm.

For each problem instance we run H-Root to find an initial feasible solution and repeat this procedure at every 75 branch-and-bound nodes to improve the current solution at hand. As we present in the following section, our numerical experiments show that H-Root can produce very high quality solutions and boosts the performance of the branch-and-price algorithm drastically.

For the exact solution of the pricing problem we employed the state-of-the art algorithm proposed by Santos et al. [121] (Alg 3 by their notation). As a subroutine, Alg 3 requires to solve several k-shortest path problems and the authors use the algorithm presented by de Azevedo et al. [34] for this task. Although very efficient, this algorithm can produce non-simple paths and performs rather poorly in our pricing graph instances that can include arcs with zero length. Therefore, different than Santos et al. [121], we implemented Yen’s loop-less k-shortest path algorithm [151] as a subroutine in Alg 3.

Our preliminary results have shown that for all the problem instances we have

studied, employing heuristic H_q has been very useful to reduce solution times. For some problem instances (mostly for those with α_{min} in COST-266 network) we were not able to find the optimal solution within the time limit of 1 hour unless we apply H_q . Since the 4th and 15th shortest loop-less paths between any two nodes of the NSF and COST-266 networks, respectively, has lengths more than the optical reach of the BPSK modulation (2880 km), the pricing problem is solved exactly by the heuristics H_4 and H_{15} . Hence, as the solution of the pricing problem we use H_4 and H_{15} for NSF and COST-266 networks, respectively.

We conduct a best bound search to explore the branch-and-bound tree. Although this strategy cannot explore as many nodes as the depth first search strategy, our computational studies showed that it can converge much faster by exploiting the high quality heuristic solution of H_q that can prune a significant part of the search tree.

4.3.4 Performance of the Branch-and-Price Algorithm

In this subsection we investigate the performances of the proposed solution methodology and discuss the effects of the various problem parameters on the difficulty of the resulting instances.

Our experimental design has 480 problem instances in total (2 networks, 4 demand sizes, 2 demand distributions, 6 α choices, and 5 random seeds). However, for some instances $\alpha = 0.2, 0.4$ and even $\alpha = 0.6$ are larger than α_{min} and hence there is no feasible solution. Tables 4.3, 4.4, 4.5 and 4.6 report the solutions of the branch-and-price algorithm for the total of 316 problem instances for which there exists a feasible solution. In these tables, $|K|$ is the number of connection demands, Seed is the key used to generate random numbers for the relevant problem instances, $NR.LP$ is the solution of the linear relaxation, HS is the number regenerators found by the heuristic solution H-Root, $HS.RT$ is the run time (in seconds) for H-Root, NR is the solution found by the branch-and-price algorithm, $NR.LP$ is the proven lower bound for the number of regenerators, RT is the solution time (in seconds) for the branch-and-price algorithm, $\#BB$ is the

number of branch-and-bound nodes explored by the branch-and-price algorithm, $\#Col.G.$ is the total number of columns generated during the branch-and-price algorithm, $\%SW.$ shows the percentage of demands that have gone through at least one modulation swap (at a regenerator node) on its path and finally $Reg.Loc.$ reports the nodes that are chosen to be a regenerator point by the branch-and-price algorithm.

As we can see from the results, 303 out of 316 problems were solved optimally and for the remaining 13 the optimality gap was reduced to just 1 regenerator within the given time limit of **3600** seconds. For 12 out of the 13 unsolved problem instances, the bandwidth utilization level is equal to the minimum (i.e. $\alpha = \alpha_{min}$).

Table 4.7 depicts a summary of the results in Tables 4.3,4.4,4.5 and 4.6. Cells occupied with "-" are the cases where the related problem has no feasible solution. It is easy to read from Table 4.7 that the smaller α values (limited link capacities) makes the RLP-FON instances harder to solve. This is due to the fact that the short supply of bandwidth slots entails high percentage of the flow variables to assume fractional values in the LP solution of the problem and the branch-and-bound tree grows significantly.

The results also show that problem instances with the NSF network are much easier to solve than those of COST-266. This is in part due to the higher number of nodes and edges in the latter. However, the substantial difference in the solution times points to a more significant effect in play. In NSF network the mean edge length is more than two times larger than that of the COST-266 network and consequently the optical reach constraints are more binding. As a result, for each demand, the number of alternative paths is rather limited for the NSF network compared to the COST-266 network. Moreover, the topology is quite different between the two networks. In the NSF network most of the nodes are in the periphery. On the other hand, the COST-266 network has a much crowded core which contains more than half of the nodes. Such a composition significantly increases the number of alternative routings for each demand and makes problem instances challenging.

Table 4.3: Results for COST-266 network with uniform demand distribution

$ K $	Seed	α	NR.LP	HS	H.RT	NR	NR.LB	RT	#BB	#Col.G.	% SW.	Reg.Loc.
75	1	0.23	3.00	8	37	8	7	3600	410	21765	53.3	5 6 9 14 15 20 23 24
		0.40	1.50	2	14	2	2	19	0	9819	21.3	4 14
		0.60	1.50	2	4	2	2	7	0	4102	24.0	5 17
		0.80	1.50	2	3	2	2	6	0	4242	26.7	5 17
		1.00	1.50	5	4	2	2	12	6	5253	21.3	4 27
	2	0.27	2.52	5	38	5	5	223	29	18845	49.3	5 6 20 23 25
		0.40	1.50	2	35	2	2	43	0	14167	33.3	5 23
		0.60	1.50	2	6	2	2	9	0	4674	40.0	4 5
		0.80	1.50	2	4	2	2	7	0	4385	52.0	4 5
		1.00	1.50	2	3	2	2	6	0	4070	21.3	5 17
	3	0.13	2.62	6	33	6	6	353	73	18049	49.3	3 5 14 15 20 24
		0.20	1.50	3	21	2	2	168	75	15224	21.3	3 24
		0.40	1.50	2	3	2	2	6	0	3846	21.3	10 20
		0.60	1.50	4	4	2	2	12	7	6028	10.7	3 13
		0.80	1.50	4	4	2	2	12	7	5983	32.0	4 5
		1.00	1.50	2	1	2	2	5	0	5334	4.0	9 16
	4	0.16	6.04	12	52	12	11	3600	3777	9754	57.3	3 4 5 6 10 13 14 15 16 20 24 27
		0.20	2.70	6	48	6	6	868	113	22633	52.0	5 14 16 22 24 26
		0.40	1.50	2	5	2	2	8	0	5265	41.3	4 5
		0.60	1.50	2	3	2	2	6	0	4617	8.0	4 13
		0.80	1.50	2	2	2	2	5	0	4027	37.3	5 14
		1.00	1.50	2	2	2	2	4	0	4363	1.3	9 14
	5	0.29	1.50	2	23	2	2	34	0	14597	20.0	23 24
		0.40	1.50	2	10	2	2	13	0	6153	9.3	14 26
		0.60	1.50	2	3	2	2	6	0	4857	6.7	9 20
		0.80	1.50	4	4	2	2	13	6	6258	37.3	23 28
		1.00	1.50	2	4	2	2	7	0	4341	30.7	5 17
100	1	0.26	5.73	11	204	10	10	414	75	14752	53.0	3 4 5 6 9 14 15 20 23 24
		0.40	1.50	3	55	3	2	168	10	22279	32.0	5 22 24
		0.60	1.50	2	4	2	2	10	0	6127	16.0	13 23
		0.80	1.50	3	6	2	2	30	14	7591	15.0	10 14
		1.00	1.50	2	9	2	2	14	0	6410	26.0	5 19
	2	0.33	3.66	6	66	6	6	363	26	22108	40.0	5 6 9 20 23 24
		0.40	1.63	3	134	3	3	224	4	22460	34.0	5 23 25
		0.60	1.50	7	14	2	2	99	75	13323	23.0	5 20
		0.80	1.50	7	11	2	2	28	5	8586	24.0	4 25
		1.00	1.50	8	9	2	2	29	9	8213	20.0	11 20
	3	0.19	4.20	8	54	8	7	3600	1590	21501	55.0	4 5 6 14 15 20 23 24
		0.20	3.01	7	44	6	6	482	75	25693	46.0	4 5 6 20 23 25
		0.40	1.50	2	17	2	2	27	0	13523	15.0	10 23
		0.60	1.50	4	8	2	2	25	6	8278	14.0	3 9
		0.80	1.50	7	7	2	2	22	5	9245	24.0	14 24
		1.00	1.50	4	7	2	2	20	5	8211	18.0	3 14
	4	0.21	6.76	12	69	12	11	3600	1701	20007	56.0	4 5 6 9 13 14 15 16 20 22 24 25
		0.40	1.50	2	29	2	2	44	0	17230	14.0	4 15
		0.60	1.50	2	7	2	2	12	0	6200	22.0	14 24
		0.80	1.50	3	5	2	2	22	6	7544	14.0	6 15
		1.00	1.50	3	8	2	2	21	4	7423	33.0	23 24
	5	0.32	2.07	5	58	4	4	956	75	32616	48.0	5 20 23 24
		0.40	1.50	3	60	2	2	356	75	23494	26.0	23 24
		0.60	1.50	2	5	2	2	11	0	6182	13.0	9 23
		0.80	1.50	2	5	2	2	10	0	6455	11.0	1 24
		1.00	1.50	5	7	2	2	29	11	8414	25.0	10 23
125	1	0.32	4.63	9	66	9	9	191	26	18262	57.6	4 5 6 9 14 15 20 22 24
		0.40	1.78	4	87	4	4	687	28	37169	44.0	5 15 23 24
		0.60	1.50	3	36	2	2	193	75	22814	16.0	3 24
		0.80	1.50	4	11	2	2	88	75	12872	18.4	23 24
		1.00	1.50	6	11	2	2	49	15	13058	34.4	5 20
	2	0.40	2.98	6	216	6	5	3600	551	35617	48.0	3 5 6 15 23 24
		0.60	1.50	3	151	2	2	435	75	29420	28.0	3 5
		0.80	1.50	8	22	2	2	146	75	16685	9.6	11 14
		1.00	1.50	2	7	2	2	15	0	8626	26.4	5 15
	3	0.23	4.25	9	44	8	8	589	75	32706	54.4	4 5 6 7 14 20 23 25
		0.40	2.50	2	43	2	2	65	0	21391	17.6	14 24
		0.60	1.50	3	15	2	2	54	19	11774	16.8	4 14
		0.80	1.50	7	10	2	2	36	7	11457	23.2	23 24
		1.00	1.50	8	11	2	2	27	4	9321	17.6	4 27
	4	0.29	5.68	11	58	10	10	352	75	20767	55.2	4 5 6 13 14 15 16 22 24 26
		0.40	1.79	4	107	4	4	814	33	35450	44.0	3 5 14 24
		0.60	1.50	2	34	2	2	45	0	15497	19.2	14 23
		0.80	1.50	5	23	2	2	60	19	10327	18.4	24 28
		1.00	1.50	5	8	2	2	29	9	9626	26.4	5 28
	5	0.37	2.65	6	86	6	6	1809	146	43467	52.0	5 14 19 20 23 25
		0.40	2.13	4	80	4	4	362	11	30128	47.2	5 20 23 24
		0.60	1.50	3	63	2	2	329	75	26272	28.8	23 24
		0.80	1.50	2	5	2	2	13	0	7728	24.0	5 23
		1.00	1.50	4	15	2	2	45	7	10982	44.0	4 5
150	1	0.36	4.50	8	62	8	8	205	21	16932	54.0	4 5 6 14 15 20 23 24
		0.40	2.35	5	69	5	5	254	18	30980	45.3	3 5 14 15 24
		0.60	1.50	2	63	2	2	90	0	24709	32.7	4 5
		0.80	1.50	3	42	2	2	185	75	17660	12.0	6 14
		1.00	1.50	4	11	2	2	136	75	14450	6.0	17 24
	2	0.49	2.92	6	156	5	5	1800	75	48989	32.7	5 6 9 23 24
		0.60	1.61	3	163	3	3	361	5	40331	22.7	4 6 24
		0.80	1.50	2	40	2	2	58	0	16952	22.7	3 5
		1.00	1.50	2	10	2	2	23	0	10800	6.7	4 27
	3	0.31	3.58	6	257	6	6	572	33	32811	34.7	3 5 6 9 15 24
		0.40	1.54	3	95	2	2	542	75	34671	28.0	5 23
		0.60	1.50	2	49	2	2	66	0	20486	19.3	20 24
		0.80	1.50	8	20	2	2	156	75	16714	6.0	3 13
		1.00	1.50	6	24	2	2	105	36	16419	20.0	3 10
	4	0.36	5.41	9	199	9	9	283	13	15145	48.0	3 4 5 6 13 14 15 20 24
		0.40	2.94	6	303	6	6	706	46	36044	50.7	4 5 13 15 20 24
		0.60	1.50	3	112	2	2	338	75	32311	26.7	5 23
		0.80	1.50	3	23	2	2	115	62	17032	35.3	5 23
		1.00	1.50	2	24	2	2	35	0	12711	26.7	3 5
	5	0.38	3.47	7	56	7	7	601	64	42469	54.0	4 5 13 14 16 20 23
		0.40	2.99	6	80	6	6	849	62	44329	42.0	5 14 16 20 23 24
		0.60	1.50	3	157	2	2	564	75	37924	25.3	24 28
		0.80	1.50	5	28	2	2	167	75	18406	16.0	3 28
		1.00	1.50	6	25	2	2	162	75	16697	25.3	23 28

Table 4.4: Results for COST-266 network with traffic density demand distribution

$ K $	Seed	α	NR.LP	HS	H.RT	NR	NR.LB	RT	#BB	#Col.G.	% SW.	Reg.Loc.
75	1	0.18	5.17	10	33	10	10	218	42	14669	50.7	1 3 5 6 7 14 15 20 21 22 24
		0.20	3.26	6	41	6	6	189	18	17946	52.0	3 5 6 20 23 24
		0.40	1.50	2	10	2	2	16	0	11138	16.0	9 28
		0.60	1.50	2	3	2	2	7	0	6072	28.0	23 26
		0.80	1.50	3	4	2	2	18	8	6596	20.0	16 24
		1.00	1.50	3	2	2	2	12	5	6711	18.7	4 14
	2	0.14	5.64	12	21	11	10	3600	4528	9385	49.3	3 4 5 14 15 16 20 21 23 24 26
		0.20	1.76	4	47	4	3	3600	1088	22945	42.7	3 5 22 24
		0.40	1.50	2	3	2	2	6	0	4202	16.0	17 24
		0.60	1.50	2	2	2	2	6	0	4725	0.0	9 21
		0.80	1.50	2	2	2	2	5	0	4087	21.3	24 28
		1.00	1.50	2	2	2	2	6	0	4192	17.3	4 14
	3	0.15	3.57	9	40	8	8	262	75	14120	53.3	3 4 5 14 15 20 22 24
		0.20	1.70	4	33	4	4	237	33	20547	38.7	3 5 22 24
		0.40	1.00	1	2	1	1	6	0	5976	12.0	3
		0.60	1.00	1	2	1	1	5	0	4601	22.7	3
		0.80	1.00	1	2	1	1	5	0	4811	20.0	3
		1.00	1.00	1	2	1	1	5	0	4023	21.3	3
	4	0.28	3.07	6	36	6	6	143	20	13670	38.7	3 5 14 16 20 23
		0.40	1.54	3	45	2	2	184	75	17244	30.7	22 24
		0.60	1.50	3	8	2	2	69	75	6583	9.3	10 14
		0.80	1.50	2	3	2	2	6	0	3628	16.0	1 24
		1.00	1.50	3	4	2	2	11	5	5574	20.0	4 25
	5	0.16	8.49	15	23	14	13	3600	3490	7400	58.7	3 4 5 6 13 14 15 20 21 22 23 24 26 27
		0.20	3.33	7	35	6	6	450	75	15863	38.7	3 5 9 20 23 24
		0.40	1.50	2	7	2	2	11	0	8213	16.0	3 4 24
		0.60	1.50	2	2	2	2	5	0	3331	21.3	3 13
		0.80	1.50	2	2	2	2	5	0	3437	21.3	19 24
		1.00	1.50	2	2	2	2	5	0	3895	13.3	19 26
100	1	0.21	6.11	12	39	12	12	105	20	9965	57.0	3 4 5 6 7 14 15 16 20 21 22 24
		0.40	1.50	2	33	2	2	52	0	19702	16.0	22 24
		0.60	1.50	4	10	2	2	34	10	10007	42.0	5 23
		0.80	1.50	3	4	2	2	23	8	10600	44.0	4 5
		1.00	1.50	3	7	2	2	24	6	9637	29.0	5 15
	2	0.22	5.16	10	25	10	10	174	40	13698	44.0	3 4 5 14 15 16 17 20 23 24
		0.40	1.50	3	31	2	2	236	75	19887	23.0	23 24
		0.60	1.50	5	8	2	2	67	75	8456	21.0	23 24
		0.80	1.50	2	3	2	2	8	0	4645	11.0	9 14
		1.00	1.50	3	4	2	2	17	6	6538	21.0	14 23
	3	0.20	4.39	9	31	9	8	3600	1483	23023	53.0	3 4 14 15 16 20 22 24 27
		0.40	1.50	2	15	2	2	23	0	12525	23.0	23 24
		0.60	1.50	3	9	2	2	24	7	7847	23.0	14 24
		0.80	1.50	2	3	2	2	7	0	4612	25.0	3 13
		1.00	1.50	3	6	2	2	15	3	7442	42.0	23 24
	4	0.33	3.37	7	45	6	6	390	75	27118	41.0	3 5 14 16 20 23
		0.40	2.13	4	50	4	4	328	21	28533	39.0	3 5 20 23
		0.60	1.50	2	19	2	2	25	0	10958	24.0	14 24
		0.80	1.50	2	5	2	2	11	0	5317	20.0	3 13
		1.00	1.50	2	2	2	2	7	0	5044	25.0	3 9
	5	0.22	6.86	12	31	12	11	3600	1125	16181	61.0	3 4 5 6 13 14 15 20 21 22 24 27
		0.40	1.50	2	34	2	2	48	0	15599	20.0	3 24
		0.60	1.50	2	8	2	2	14	0	9783	11.0	16 24
		0.80	1.50	6	5	2	2	16	5	6097	11.0	4 27
		1.00	1.50	2	2	2	2	7	0	4747	26.0	23 28
125	1	0.26	6.26	13	55	12	12	187	75	13689	55.2	1 3 4 5 6 7 14 15 20 21 22 24
		0.40	1.74	3	70	3	3	115	1	25145	37.6	3 22 24
		0.60	1.50	2	37	2	2	48	0	15076	6.4	9 16
		0.80	1.50	2	9	2	2	18	0	8843	20.0	3 9
		1.00	1.50	4	7	2	2	49	23	12239	12.8	9 23
	2	0.26	5.28	9	33	9	9	334	54	18718	43.2	3 5 14 15 16 19 20 23 24
		0.40	1.76	4	107	3	3	1056	75	37491	27.2	3 5 16
		0.60	1.50	3	17	2	2	138	75	13172	21.6	14 24
		0.80	1.50	2	10	2	2	19	0	6969	30.4	14 24
		1.00	1.50	5	8	2	2	26	6	8032	20.8	5 23
	3	0.26	4.10	8	51	8	8	913	110	32527	44.0	3 4 15 16 20 21 22 24
		0.40	1.50	3	87	2	2	502	75	27471	21.6	3 5
		0.60	1.50	2	12	2	2	18	0	7837	25.6	3 24
		0.80	1.50	4	8	2	2	32	8	11077	35.2	3 5
		1.00	1.50	4	7	2	2	30	8	10870	22.4	5 23
	4	0.41	3.05	6	59	6	6	253	26	29247	38.4	3 5 14 16 20 23
		0.60	1.50	2	76	2	2	113	0	20730	24.8	22 24
		0.80	1.50	2	25	2	2	36	0	16851	22.4	3 15
		1.00	1.50	2	9	2	2	18	0	7270	21.6	20 24
	5	0.28	6.98	13	41	12	11	3600	75	18344	54.4	3 4 5 13 14 15 20 21 24 26 27 28
		0.40	1.61	3	72	3	3	155	3	22477	32.0	3 5 24
		0.60	1.50	2	15	2	2	34	0	20330	4.8	9 14
		0.80	1.50	6	9	2	2	117	75	15128	13.6	3 13
		1.00	1.50	2	6	2	2	12	0	6320	17.6	23 24
150	1	0.31	4.95	9	70	9	9	284	27	25216	52.0	1 3 5 6 15 20 21 22 24
		0.40	2.21	4	332	4	3	3600	779	44496	44.7	3 5 20 24
		0.60	1.50	3	53	2	2	284	75	30497	26.7	24 28
		0.80	1.50	2	12	2	2	26	0	9671	3.3	9 14
		1.00	1.50	3	12	2	2	46	9	15357	10.0	4 11
	2	0.33	5.27	10	53	10	10	378	53	26104	46.7	3 5 14 15 16 17 19 20 21 23
		0.40	3.20	6	92	6	6	804	35	45207	36.7	3 15 16 17 20 24
		0.60	1.50	3	110	2	2	323	75	26560	24.7	3 15
		0.80	1.50	3	22	2	2	113	75	15019	14.0	4 15
		1.00	1.50	5	17	2	2	79	35	12884	40.0	4 5
	3	0.33	4.66	9	198	9	9	1233	139	40488	32.0	3 4 14 16 17 20 21 23 24
		0.40	2.83	5	121	4	4	934	75	44488	32.7	3 20 24 28
		0.60	1.50	3	44	2	2	190	75	25936	8.7	10 15
		0.80	1.50	3	18	2	2	140	75	15966	5.3	11 28
		1.00	1.50	3	13	2	2	80	31	15993	30.7	3 28
	4	0.49	3.23	7	106	7	7	480	50	35523	46.0	3 4 5 14 16 20 23
		0.60	1.88	4	128	4	4	859	24	45629	32.7	15 20 23 24
		0.80	1.50	3	95	2	2	444	75	35013	28.0	22 24
		1.00	1.50	2	33	2	2	47	0	21249	14.7	24 28
	5	0.34	6.60	11	244	11	10	3600	75	11607	52.0	3 4 5 6 14 15 16 20 21 22 24
		0.40	3.22	5	222	5	5	295	6	20825	38.7	3 5 20 23 24
		0.60	1.50	2	77	2	2	98	0	23414	18.7	24 28
		0.80	1.50	2	19	2	2	31	0	15521	6.7	9 28
		1.00	1.50	4	23	2	2	147	75	13260	11.3	1 24

Table 4.5: Results for NSF network with uniform demand distribution

$ K $	Seed	α	NR.LP	HS	H.RT	NR	NR.LB	RT	#BB	#Col.G.	% SW.	Reg.Loc.
75	1	0.48	2.11	3	1	3	3	3	0	901	5.3	3 4 6
		0.60	2.00	3	1	3	3	3	1	718	8.0	3 4 6
		0.80	2.00	2	1	2	2	2	0	673	2.7	4 6
		1.00	2.00	2	0	2	2	1	0	619	18.7	4 11
	2	0.44	2.19	4	1	4	4	10	9	1635	13.3	3 4 6 7
		0.60	2.00	3	1	3	3	3	2	675	14.7	3 4 6
		0.80	2.00	3	1	3	3	4	3	998	21.3	3 4 11
		1.00	2.00	2	0	2	2	1	0	487	21.3	4 11
	3	0.42	2.29	4	2	4	4	6	3	1323	12.0	3 4 6 11
		0.60	2.00	3	1	3	3	3	1	844	5.3	3 4 6
		0.80	2.00	2	1	2	2	2	0	724	2.7	4 6
		1.00	2.00	2	0	2	2	2	0	614	22.7	4 11
	4	0.41	2.17	3	1	3	3	3	0	824	10.7	3 4 6
		0.60	2.00	2	1	2	2	2	0	601	2.7	4 6
		0.80	2.00	2	0	2	2	1	0	536	18.7	4 11
		1.00	2.00	2	0	2	2	2	0	574	20.0	4 11
	5	0.51	2.19	4	1	4	4	5	3	1054	9.3	3 4 6 11
		0.60	2.01	4	1	3	3	7	22	948	14.7	3 6 7
		0.80	2.00	3	1	3	3	2	1	813	2.7	3 4 6
		1.00	2.00	2	1	2	2	2	0	631	1.3	4 6
100	1	0.68	2.20	4	4	4	4	10	2	1541	12.0	3 4 6 11
		0.80	2.02	3	2	3	3	5	0	1310	9.0	3 4 6
		1.00	2.00	3	2	3	3	5	1	1066	15.0	4 6 11
	2	0.62	2.17	3	2	3	3	5	0	1340	10.0	3 4 6
		0.80	2.00	4	2	3	3	9	12	1235	17.0	3 4 6
		1.00	2.00	3	1	3	3	4	2	917	16.0	3 4 6
	3	0.67	2.15	3	2	3	3	5	0	1313	7.0	3 4 6
		0.80	2.00	3	1	3	3	4	1	1057	5.0	3 4 6
		1.00	2.00	3	1	3	3	4	1	1041	4.0	4 6 11
	4	0.44	2.37	4	3	4	4	9	3	1603	17.0	3 4 6 11
		0.60	2.00	3	1	3	3	6	2	1213	9.0	4 6 11
		0.80	2.00	2	1	2	2	3	0	845	3.0	4 6
		1.00	2.00	2	1	2	2	2	0	801	20.0	4 11
	5	0.62	2.17	3	2	3	3	4	0	1243	8.0	3 4 6
		0.80	2.00	4	1	3	3	11	10	1326	7.0	3 4 6
		1.00	2.00	3	1	3	3	3	1	922	6.0	3 4 6
125	1	0.84	2.22	4	4	4	4	14	4	2106	11.2	3 4 6 11
		1.00	2.04	3	2	3	3	7	0	1804	5.6	3 4 6
	2	0.86	2.11	3	2	3	3	6	0	1544	11.2	3 4 6
		1.00	2.00	3	2	3	3	7	1	1195	10.4	3 4 6
	3	0.77	2.12	3	3	3	3	8	0	1843	8.8	3 4 6
		0.80	2.08	3	2	3	3	6	0	1737	11.2	3 4 6
		1.00	2.00	3	2	3	3	5	1	1340	10.4	3 4 6
	4	0.65	2.41	4	6	4	4	14	2	1933	12.8	3 4 6 11
		0.80	2.13	3	3	3	3	8	0	1855	9.6	3 4 6
		1.00	2.00	3	1	3	3	6	2	1176	10.4	3 4 6
	5	0.76	2.18	4	4	4	4	10	3	1877	18.4	3 4 6 11
		0.80	2.13	3	2	3	3	6	0	1318	8.0	3 4 6
		1.00	2.00	3	2	3	3	6	1	1183	5.6	3 4 6
150	1	0.96	2.25	4	6	4	4	14	4	2262	12.7	3 4 6 11
		1.00	2.20	4	11	4	4	16	2	1954	14.0	3 4 6 11
	2	1.04	2.12	4	4	4	4	12	4	2140	14.7	3 4 6 11
	3	0.94	2.14	3	3	3	3	8	0	1980	6.7	3 4 6
		1.00	2.07	3	2	3	3	7	0	1656	11.3	3 4 6
	4	0.79	2.45	4	5	4	4	14	2	2056	14.7	3 4 6 11
		0.80	2.42	4	5	4	4	13	2	1914	12.7	3 4 6 11
		1.00	2.13	3	4	3	3	9	0	2209	9.3	3 4 6
	5	0.93	2.15	3	4	3	3	8	0	1888	6.7	3 4 6
		1.00	2.07	3	3	3	3	7	0	1524	8.0	3 4 6

Table 4.6: Results for NSF network with traffic density demand distribution

$ K $	Seed	α	NR.LP	HS	H.RT	NR	NR.LB	RT	#BB	#Col.G.	% SW.	Reg.Loc.
75	1	0.43	2.25	3	1	3	3	2	0	661	2.7	3 4 6
		0.60	2.00	3	1	3	3	2	1	616	1.3	3 4 6
		0.80	2.00	2	1	2	2	2	0	546	0.0	4 6
		1.00	2.00	2	1	2	2	2	0	555	13.3	2 7
	2	0.38	2.34	5	2	5	5	7	5	1227	32.0	3 4 6 9 11
		0.60	2.00	3	1	3	3	2	1	899	2.7	3 4 6
		0.80	2.00	3	1	3	3	3	3	972	6.7	3 4 6
		1.00	2.00	2	0	2	2	2	0	650	30.7	4 11
	3	0.48	2.09	3	1	3	3	2	0	713	6.7	3 4 6
		0.60	2.00	3	1	3	3	2	1	739	10.7	3 4 6
		0.80	2.00	3	1	3	3	2	2	687	8.0	3 6 7
		1.00	2.00	2	0	2	2	2	0	571	22.7	4 11
	4	0.71	2.08	3	1	3	3	2	0	749	9.3	3 4 6
		0.80	2.00	3	1	3	3	2	1	759	6.7	3 4 6
		1.00	2.00	3	1	3	3	2	1	719	9.3	3 4 6
	5	0.67	2.19	4	2	4	4	6	4	1138	9.3	3 4 6 11
		0.80	2.00	3	1	3	3	3	1	846	8.0	3 4 6
		1.00	2.00	3	1	3	3	3	1	785	4.0	3 4 6
100	1	0.52	2.19	4	2	3	3	18	53	1524	14.0	3 5 6
		0.60	2.04	3	1	3	3	3	0	904	4.0	3 4 6
		0.80	2.00	2	1	2	2	2	0	697	0.0	4 6
		1.00	2.00	2	1	2	2	2	0	722	2.0	4 6
	2	0.63	2.25	4	3	4	4	9	3	1298	18.0	3 4 6 11
		0.80	2.00	3	1	3	3	4	0	1251	5.0	3 4 6
		1.00	2.00	3	1	3	3	4	1	1045	18.0	4 6 11
	3	0.72	2.00	3	1	3	3	4	1	951	9.0	3 4 6
		0.80	2.00	3	1	3	3	3	1	916	10.0	3 4 6
		1.00	2.00	3	1	3	3	3	1	925	7.0	3 6 7
	4	0.79	2.10	4	1	4	3	5	2	1029	7.0	3 4 6 7
		0.80	2.08	3	1	3	3	4	0	1001	7.0	3 4 6
		1.00	2.00	3	1	3	3	4	1	953	7.0	3 4 6
	5	0.75	2.22	3	2	3	3	6	0	1199	7.0	3 4 6
		0.80	2.14	3	2	3	3	4	0	1112	8.0	3 4 6
		1.00	2.00	3	1	3	3	4	1	1117	5.0	3 4 6
125	1	0.62	2.28	4	3	4	4	10	4	1527	7.2	3 4 6 11
		0.80	2.00	3	2	3	3	7	1	1546	16.8	3 5 6
		1.00	2.00	3	1	3	3	4	1	948	1.6	4 6 11
	2	0.77	2.22	4	3	4	4	10	3	1711	16.0	3 4 6 11
		0.80	2.18	4	4	4	4	11	3	1798	4.8	3 4 6 11
		1.00	2.00	3	2	3	3	5	1	1390	3.2	3 4 6
	3	0.89	2.04	3	2	3	3	5	0	1392	8.0	3 4 6
		1.00	2.00	3	2	3	3	5	1	1237	9.6	3 4 6
	4	0.98	2.15	4	2	4	4	9	3	1494	8.0	3 4 6 7
		1.00	2.13	4	2	4	4	11	4	1786	8.8	3 4 6 7
	5	0.93	2.17	4	2	3	3	26	75	2282	8.0	3 4 6
		1.00	2.08	3	3	3	3	8	0	1397	5.6	3 4 6
150	1	0.73	2.28	4	4	4	4	9	2	1695	7.3	3 4 6 11
		0.80	2.19	4	3	3	3	16	50	1974	14.0	3 5 6
		1.00	2.00	3	2	3	3	7	1	1428	4.7	3 4 6
	2	0.88	2.24	4	4	4	4	9	3	1774	17.3	3 4 6 11
		1.00	2.11	4	6	4	4	11	3	1957	5.3	3 4 6 11
	3	1.01	2.07	3	3	3	3	7	0	1569	6.0	3 4 6
	4	1.25	2.14	4	6	4	4	12	3	1888	13.3	3 4 6 7
	5	1.09	2.21	3	4	3	3	8	0	1816	5.3	3 4 6

Not surprisingly, problem instances with higher number of demands are harder to solve. What is interesting is the higher solution times for the problem instances with the *traffic density* demand distribution. A possible explanation for this result could be the higher concentration of connection demands on some specific node pairs which exacerbates the problem of bandwidth capacity limitations.

Our numerical experiments show that H-Root can produce high quality solutions. Among the 316 problem instances, H-Root could find the optimal solution in 214 (67.72%). For the 65 instances out of the remaining 102, H-Root could find a solution with an optimality gap of just 1 regenerator. Interestingly, for some instances, H-Root would return a higher number of regenerators when α increases (e.g. Table 4.3, $|K| = 75$, $Seed = 1$). The reason for such a result is the lack of some critical path-segments that are not generated in the root node when α is larger and LP relaxation of the problem has to be solved with a more restricted set of columns.

4.3.5 Managerial Insights

The relation between the number of regenerators and the spectral efficiency is an interesting one for the network management who wants to deploy/operate minimum number of regenerators to lower the capital investment and operational costs (such as energy and maintenance costs) but also wants to achieve higher bandwidth utilization efficiency to be able to satisfy more demand and build resilience against the failures in the network components. As detailed below, our results show that with FON architecture, a smart deployment of a rather limited number of regenerators can achieve high levels of bandwidth efficiency. These promising results indicate that FON architecture can provide significant cost reductions and capacity enhancements at the same time.

Table 4.8 shows the lower bounds for the optimal solution values of the 27 problem instances with COST-266 network and traffic density distribution of 75,100,125 and 150 connection demands generated by random number seed 2. Cells depicted with "-" are the cases where the problem has no feasible solution

Table 4.7: Exact solution results for COST-266 and NSF networks

Network	Demand	K	Average Regenerator Number					Average Run Time(seconds)					Average #BB nodes					Average #Col Generated								
			α_{min}	0.2	0.4	0.6	0.8	1.0	α_{min}	0.2	0.4	0.6	0.8	1.0	α_{min}	0.2	0.4	0.6	0.8	1.0	α_{min}	0.2	0.4	0.6	0.8	1.0
COST-266	TD	75	9.8	5.0	1.8	1.8	1.8	1564.4	1119.1	44.6	18.3	7.7	7.8	1631.0	303.5	15.0	15.0	1.6	2.0	11848.8	19325.3	9354.6	5062.4	4511.8	4879.0	
		100	9.8	-	2.4	2.0	2.0	1573.9	-	137.3	33.0	13.0	14.0	548.6	-	19.2	18.4	2.6	3.0	17997.0	-	19249.2	9410.2	6254.2	6681.6	
		125	9.4	-	2.8	2.0	2.0	1057.4	-	456.8	70.2	44.3	27.1	68.0	-	38.5	15.0	16.6	7.4	22505.0	-	28146.0	15429.0	11773.6	8946.2	
		150	9.2	-	4.8	2.4	2.0	2.0	1195.3	-	1408.2	350.7	150.9	80.1	68.8	-	223.8	49.8	45.0	30.0	27787.6	-	38754.0	30407.2	18238.0	15748.6
	U	75	6.6	4.0	2.0	2.0	2.0	1561.9	518.2	17.7	8.1	8.5	6.8	857.8	94.0	0.0	1.4	2.6	1.2	16602.0	18928.5	7850.0	4855.6	4979.0	4672.2	
		100	8.0	6.0	2.4	2.0	2.0	2.0	1786.7	482.1	163.6	31.3	22.3	22.6	693.4	75.0	17.8	16.2	6.0	5.8	22196.8	25693.0	19797.2	8022.0	7884.2	7734.2
		125	7.8	-	3.5	2.0	2.0	2.0	1308.3	-	482.2	211.3	68.7	32.9	174.6	-	18.0	48.8	35.2	7.0	30163.8	-	31034.5	21155.4	11813.8	10322.6
		150	7.0	-	4.8	2.2	2.0	2.0	692.1	-	587.9	283.8	136.3	92.0	41.2	-	50.3	31.0	57.4	37.2	31269.2	-	36506.0	31152.2	17352.8	14215.4
NSF	TD	75	3.6	-	-	3.0	2.8	2.4	4.1	-	-	2.1	2.5	2.1	1.8	-	-	1.0	1.4	0.4	897.6	-	-	751.3	762.0	656.0
		100	3.4	-	-	3.0	2.8	2.8	8.1	-	-	3.4	3.6	3.4	11.8	-	-	0.0	0.2	0.8	1200.2	-	-	904.0	995.4	952.4
		125	3.6	-	-	-	3.5	3.2	11.9	-	-	-	8.7	6.7	17.0	-	-	-	2.0	1.4	1681.2	-	-	-	1672.0	1351.6
		150	3.6	-	-	-	3.0	3.5	8.9	-	-	-	15.8	8.9	1.6	-	-	-	50.0	2.0	1748.4	-	-	-	1974.0	1692.5
	U	75	3.6	-	-	2.8	2.4	2.0	5.4	-	-	3.6	2.3	1.6	3.0	-	-	5.2	0.8	0.0	1147.4	-	-	757.2	748.8	585.0
		100	3.4	-	-	3.0	2.8	2.8	6.6	-	-	6.2	6.3	3.7	1.0	-	-	2.0	4.6	1.0	1408.0	-	-	1213.0	1154.6	949.4
		125	3.6	-	-	-	3.0	3.0	10.6	-	-	-	6.7	6.3	1.8	-	-	-	0.0	1.0	1860.6	-	-	-	1636.7	1339.6
		150	3.6	-	-	-	4.0	3.3	11.2	-	-	-	13.1	9.8	2.0	-	-	-	2.0	0.5	2065.2	-	-	-	1914.0	1835.8

Table 4.8: Number of required regenerators for COST-266 network with traffic density demand distribution.

Percentage of bandwidth allocated (α)										
K	0.14	0.20	0.22	0.26	0.33	0.40	0.60	0.80	1.00	
75	10	3	3	2	2	2	2	2	2	2
100	-	-	10	6	3	2	2	2	2	2
125	-	-	-	9	5	3	2	2	2	2
150	-	-	-	-	10	6	2	2	2	2

and for each row, the first cell with a numeric value is the case where the percentage of allocated bandwidth is the minimum (i.e $\alpha = \alpha_{min}$). In order not to leave any dents in Table 4.8 we generate and solve additional problem instances, using different α values than we have in Table 4.4.

A quick look at the table reveals that certain combinations of regenerator deployment and bandwidth utilization (α) levels could be very attractive for the network management. For example, the first row ($|K| = 75$) of the table shows that with just 3 regenerators (deploying regenerators at 11% of nodes), it is possible to satisfy all the demand by using only 20% of the available bandwidth in each link. Note that for this case the lowest possible utilization level is 0.14 which requires 10 regenerators. Moreover, Tables 4.3-4.6 together show that regardless of the network and demand distribution differences, the same conclusion stays valid. Our results also show that some nodes are more likely to appear as regeneration nodes (in the optimal solutions regenerators are placed/activated on those nodes). In Table 4.9, each row depicts the results of 5 different RLP-FON instances with the COST-266 network and traffic density demand distribution.

Table 4.9: Percentage of nodes appearing as regeneration points in the optimal solutions for the problem instances with COST-266 network and traffic density demand distribution.

$ K $	α	Node Number																											
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
75	α_{min}	20	0	100	60	100	40	20	0	0	0	0	0	20	100	80	40	0	0	0	100	60	60	60	60	0	20	0	0
	0.20	0	0	100	0	100	25	0	0	25	0	0	0	0	0	0	0	0	0	0	50	0	50	50	100	0	0	0	0
	0.40	0	0	40	0	0	0	0	0	40	0	0	0	0	0	0	0	20	0	0	0	0	20	0	40	0	0	0	20
	0.60	0	0	40	0	0	0	0	0	20	20	0	0	20	20	0	0	0	0	0	0	20	0	20	0	0	20	0	0
	0.80	20	0	20	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	20	0	0	0	80	0	0	0	20	0
	1.00	0	0	20	60	0	0	0	0	0	0	0	0	0	40	0	0	0	0	20	0	0	0	0	0	20	20	0	0
100	α_{min}	0	0	100	80	80	40	20	0	0	0	0	0	20	100	80	80	20	0	0	100	40	60	40	60	0	0	20	0
	0.40	0	0	40	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	20	60	80	0	0	0	0
	0.60	0	0	0	0	20	0	0	0	0	0	0	0	0	40	0	20	0	0	0	0	0	0	40	80	0	0	0	0
	0.80	0	0	40	40	20	0	0	0	20	0	0	0	40	20	0	0	0	0	0	0	0	0	0	0	0	0	20	0
	1.00	0	0	20	0	20	0	0	0	20	0	0	0	0	20	20	0	0	0	0	0	0	0	60	20	0	0	0	20
125	α_{min}	20	0	100	60	80	20	20	0	0	0	0	0	20	80	80	60	0	0	20	100	60	40	40	60	0	20	20	0
	0.40	0	0	100	0	75	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	25	0	50	0	0	0	0
	0.60	0	0	20	0	0	0	0	0	40	0	0	0	0	40	0	20	0	0	0	0	0	20	0	60	0	0	0	0
	0.80	0	0	80	0	20	0	0	0	20	0	0	0	20	20	20	0	0	0	0	0	0	0	0	20	0	0	0	0
	1.00	0	0	0	0	40	0	0	0	20	0	0	0	0	0	0	0	0	0	0	20	0	0	80	40	0	0	0	0
150	α_{min}	20	0	100	60	80	40	0	0	0	0	0	0	0	80	60	80	40	0	20	100	80	40	60	60	0	0	0	0
	0.40	0	0	100	0	50	0	0	0	0	0	0	0	0	0	25	25	25	0	0	100	0	0	25	100	0	0	0	25
	0.60	0	0	20	0	0	0	0	0	0	20	0	0	0	0	60	0	0	0	0	20	0	0	20	60	0	0	0	40
	0.80	0	0	0	20	0	0	0	0	40	0	20	0	0	20	20	0	0	0	0	0	0	20	0	20	0	0	0	40
	1.00	20	0	20	40	20	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	40

For each node, the percentage of solutions for which that node appears in the set of regenerator nodes is given in the table. For example, looking at the first row we can see that nodes 3,5,14 and 20 have been chosen as the regenerator nodes at all of the 5 problem instances solved with the minimum bandwidth allocation ($\alpha = \alpha_{min}$), whereas, nodes 1,13 and 26 were selected just once. Looking at the table, one can see that the optimal locations of the regenerators do not change drastically with the fluctuations in the demand. From Tables 4.5 and 4.6 we can see the same conclusion is also true for the NSF network for which the various combinations of the nodes 3,4,6 and 11 constitute approximately the 89.3% of the optimal solutions. Thus, the solutions obtained by the proposed algorithm can be considered as somewhat robust solutions. This is a desired property for the network management especially when it is hard to accurately estimate the demand at the time of planning.

Another interesting data in the given tables is the percentage of modulation swaps. We call it a modulation swap if a demand uses more than one modulation level (by going through modulation conversion in a regenerator node) on its light-path. Our results show that, in general, more strict bandwidth limitations

necessitate more modulation swaps to satisfy connection demand with less network resources. For example, in the first line of Table 4.3, 53.3% of the demands have gone through at least one modulation swap on their determined light-paths when the allowed bandwidth is at its minimum ($\alpha = \alpha_{min}$). This number reduces to 21.3% when all the bandwidth is allowed to be used. A very similar trend is clearly visible in other tables as well. Thus, we can see that, at least according to the results of our numerical experiments, FON architecture's novel capability of multi modulation transmission appears to be very useful in increasing resource utilization efficiency for the optical networks.

4.4 Summary

In this chapter we revisit the regenerator location problem from the flexible optical network architecture perspective and introduce this problem to the operations research (OR) literature. Since the concept of FON architecture is quite new (due to the recent maturation of the enabling hardware technologies), this practically significant and theoretically interesting problem is not well studied in the literature. One of the purposes of this part of our dissertation study is to draw the attention of the OR researchers to this gap in the literature and promote new studies in this promising area of research.

For the considered problem, we presented a path-segment based formulation. Our solution methodology introduces a new perspective to general path based formulations. In particular, our novel path-segment formulation makes it easy to include some special constraints on the paths which are otherwise harder to incorporate in a plain path based formulation. We use this powerful formulation approach in the following chapters to extend our study to different relay location problems with different limitations on the routes to connect OD pairs.

We propose an efficient branch-and-price algorithm to solve RLP-FON. We conducted extensive numerical experiments to test the performance of the proposed algorithm. As explained above, RLP-FON requires to solve regenerator

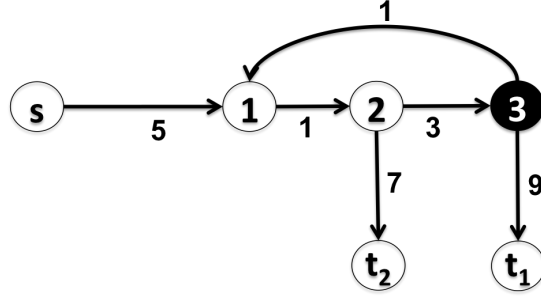
location, routing, spectrum allocation and modulation selection problems jointly. The performance of the proposed algorithm is comparable to the state-of-the-art heuristic algorithms that solve these problems sequentially. From the practical point of view, these numerical studies provide significant managerial insights about this urgent problem. In particular, our findings show that with the FON architecture, it is possible to enhance network capacity and reduce the capital and operational costs of the optical network.

In its current form, arc costs are not considered in the RLP-FON formulation (they are simply assumed to be 0). But this is only because of the practical setting of the problem where arc costs are either negligible or do not scale up to the regenerator deployment costs. From the mathematical point of view, adding arc costs does not disturb the main structure of the proposed algorithm at all. Note that adding arc costs to the objective function or considering new constraints on some resource usage at arcs can be included in the pricing problem by just adding a new term in the calculation of arc lengths of the pricing graph. Similarly edge design aspect can be easily incorporated with some slight modifications as we show in the next chapter when we consider the NDR. Thus, as we show in the subsequent chapters, our formulation and solution methodology can be easily adapted to the rather general multi commodity, multi modal flow problems arising in different contexts than the optical networks.

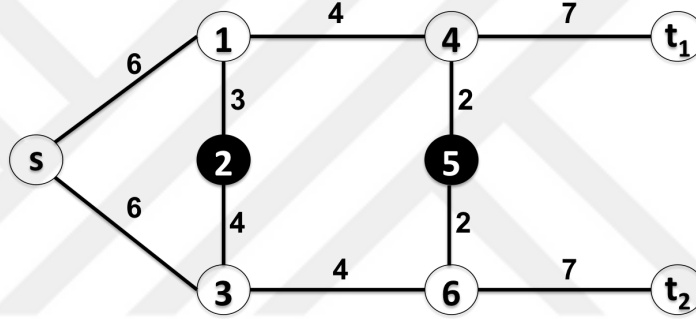
Chapter 5

Network Design Problem with Relays

In this chapter we extend our solution approach for RLP-FON to consider edge design and relay location decisions jointly to solve the NDR. In Chapter 3 we show that, in relay location problems non-simple paths can be a part of the optimal solutions. This adds extra challenges to classical network design problems. For instance, in DNDR an optimal solution can contain paths with loops. In Figure 1(a) a simple example is given to illustrate this fact. There are two OD pairs, (s, t_1) and (s, t_2) and the number given on each edge is the length for that edge. Suppose all edge costs are zero and all node costs are 1. In this example the threshold length is 9 and the only optimal solution is to put a relay on node 3, use the path $s - 1 - 2 - 3 - t_1$ to connect the OD pair (s, t_1) and employ the non-simple path $s - 1 - 2 - 3 - 1 - 2 - t_2$ to connect s to t_2 . It may also happen that the path for an OD pair in an optimal solution of NDR traverses the same edge more than once. Figure 1(b) shows such an example. Assuming the same cost structure, the first OD pair (s, t_1) follows the path $s - 1 - 2 - 1 - 4 - 5 - 4 - t_1$ and the second OD pair (s, t_2) follows $s - 1 - 2 - 1 - 4 - 5 - 6 - t_2$. Nodes 2 and 5 are chosen as the relay nodes. Note that this solution is the unique optimal solution and in this solution the edges $\{1, 2\}$ and $\{4, 5\}$ are traversed back and forth. The fact that OD pairs may use paths with cycles makes NDR a very



(a) Non-simple paths in directed networks



(b) Traversing an edge more than once in undirected networks

Figure 5.1: Examples of non-simple paths in directed and undirected graphs with threshold set to 9

challenging network design problem. Here we better see the advantage of our path segment based formulation introduced in Chapter 4 by observing that for an OD pair, in the optimal solution the whole path could be a non-simple one but the path-segments it uses are always simple. This key advantage of path-segment based formulations explains the computational efficiency of our solution approach for NDR that we explain in detail in the following sections. We first investigate the related literature about the NDR.

For the solution of NDR, Cabral et al. [17] present a path-based integer programming formulation and propose a column generation approach to solve it. Since their model requires to generate all the combinations of feasible paths and relay locations in advance, it is not applicable to solve large problem instances and the authors suggest using a subset of these combinations to obtain high quality feasible solutions. They also propose four different heuristics that construct a

solution by considering OD pairs one by one. They use randomly generated grid graphs to test their solution methods on NDR-S instances. Kulturel-Konak and Konak [83] propose a multi-commodity flow formulation and present a heuristic algorithm that integrates local search and genetic algorithm approaches. The solution pool is generated by the genetic algorithm and solutions are improved by a series of local search procedures. Konak [78] separates NDR into two problems, one to find paths and the other to locate relays. He presents an efficient heuristic algorithm in which feasible paths for each OD pair are generated by a genetic algorithm and a set covering problem is then solved with these paths to locate relays in the network. The resulting algorithm is tested on problem instances with up to 160 nodes, 3624 edges and 10 OD pairs. Considering a directed graph instead of an undirected one, Li et al. [90] propose a node-arc and an arc-path formulation for DNDR. To solve the node-path formulation they suggest a branch-and-price algorithm for which the pricing problem is an NP-Hard problem called the minimum cost path problem with relays. They use a slightly modified version of the algorithm by Laporte and Pascoal [85] to solve it. In the computational experiments, for the sake of simplicity, the authors only consider simple paths in the pricing phase and omit those with cycles. In their computational studies they consider NDR-S instances. Unfortunately, it is not possible to use an algorithm that solves DNDR directly to solve NDR by simply modifying the problem data. A recent study by Leitner et al. [89] considers NDR problems in the context of refuelling station location problem and presents a multi-commodity flow formulation to solve it. Characterizing the solutions for an NDR problem and applying some graph transformations, they propose a branch-and-price algorithm which can solve small size problems up to 60 nodes and 10 OD pairs, exactly. Similar to our approach they pay a special attention to NDR-S instances and propose a branch-and price-and-cut algorithm to solve it. Considering a closely related problem to NDR: fixed-charge relay network design in telecommunications, Kewcharoenwong and Uster [67] combine the modeling approaches from multi-commodity flow network design and single assignment hub location and propose a Benders decomposition algorithm to solve this problem. They consider different reach thresholds for terminal to relay and relay to relay connections and include edge design and usage costs in their model. Different than NDR, they

require single assignments for the terminal nodes and do not allow connections through union of edges even if their total length is less than the threshold value.

5.1 Definitions and Notation

In this section, we provide definitions and notation pertinent throughout this chapter. In concordance with the common nomenclature, an NDR problem instance is specified by a physical network G with associated data l, c and h , OD pair set K , and a threshold value d_{max} . The aim is to establish a feasible route for every $k \in K$ such that the total of edge design and relay design cost is minimized. We use the path-segments as defined in 2 and call a path segment feasible if its length is less than the relay free communication range d_{max} . Additional definitions and notation will be listed on a need basis.

In this chapter, we use the notion of a directed virtual network that originates from the path-segment formulation of Chapter 4. The virtual network is derived from the physical one; it has the same nodes as the physical network but the arcs of the virtual network correspond to simple directed paths in the original graph with lengths not more than the threshold. We propose a multi-commodity flow and a tree formulation on this virtual network to solve NDR and NDR-S problems, respectively. Since the number of arcs on the virtual network grows exponentially with the size of the original network and we have a variable for each of these arcs in both formulations, we propose branch-and-price algorithms to solve our formulations.

A physical network G and a given threshold value d_{max} induce a *virtual network* $\Gamma = (V, \mathcal{A})$ where $\mathcal{A} = \{(i, j, p) : p \in \mathcal{P}, o(p) = i, d(p) = j\}$. By this construction, a feasible route in the physical network corresponds to a directed path in the virtual network. Conversely, any directed path from $\mathcal{O}(k)$ to $\mathcal{D}(k)$ in Γ corresponds to a feasible route for $k \in K$ provided that every intermediate node on this path is equipped with the relay property. Our formulations rely on the correspondence between the physical and the virtual networks. Figure 5.2

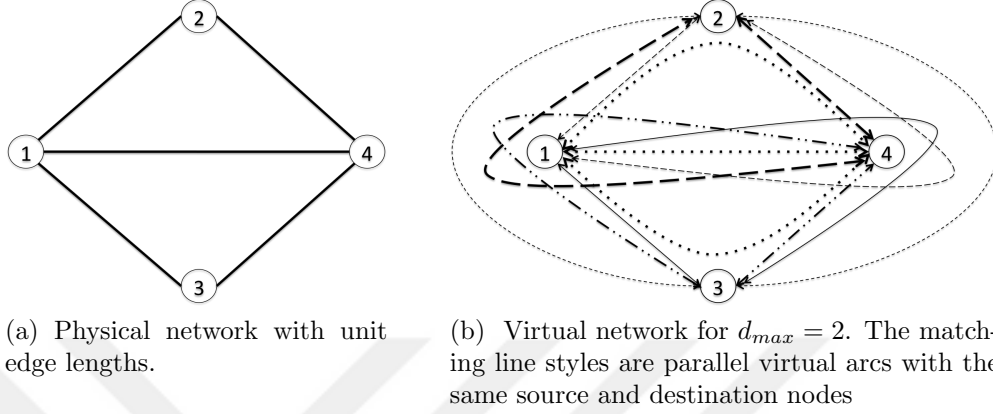


Figure 5.2: Virtual network example

presents an example for a virtual graph. In this example we consider a quite simple input graph with four nodes and five edges with unit lengths in Figure 2(a). Considering a d_{max} value of two, we obtain the virtual network depicted in Figure 2(b). In Figure 2(b) the parallel edges between the same node pairs are drawn with the same style and they are depicted in a way to show the physical path-segments they represent. For example the double headed dashed lines on the rightmost and leftmost boundaries of the figure, represent the parallel virtual network arcs between nodes 2 and 3 corresponding to the path-segments $((2,4), (4,3))$, $((2,1), (1,3))$, $((3,4), (4,2))$ and $((3,1), (1,2))$. As we see in Figure 5.2, the induced virtual graph for a quite simple input graph and moderate d_{max} value, can get quite dense. For the pair of arcs $\bar{e}, \underline{e}$ that are induced by the edge e we denote $\epsilon(\bar{e}) = \epsilon(\underline{e}) = e$.

A solution for an NDR problem in the physical network G is a pair $\langle R, T \rangle$ where $R \subseteq V$ is the set of relay locations and $T \subseteq E$ is the set of edges included in the design. The relay locations and the edges in the resulting design enable feasible routes for each OD pair $k \in K$. A solution for an NDR problem $\langle R, T \rangle$ induces a directed subgraph of the virtual network $\Gamma = (V, \mathcal{A})$ on which for every $k \in K$, there exists a directed path from $\mathcal{O}(k)$ to $\mathcal{D}(k)$ with every intermediate node in R .

5.2 Solving NDR

In this section, we present a branch-and-price algorithm to solve the general problem NDR. We first list some structural properties satisfied by certain optimal solutions. Then we provide our multi-commodity flow formulation strengthened by optimality cuts based on these properties and detail the algorithm.

5.2.1 Properties of NDR Solutions

The following result plays a key role in strengthening our formulations.

Lemma 5. *There exists an optimal solution $\langle R^*, T^* \rangle$ to NDR such that for all $k \in K$, there exists a feasible route P^k using edges $T^k \subseteq T^*$ for which T^k is a tree.*

Proof. Consider an arbitrary $k \in K$. Among all feasible routes from $\mathcal{O}(k)$ to $\mathcal{D}(k)$ pick one using the fewest number of edges T^k from T^* . Assume T^k has a cycle. Since each path-segment in P^k is simple, in order for T^k to create a cycle, there should exist two distinct path-segments $\hat{p}$ and $\hat{q}$ of P^k sharing a particular node, say i . Without loss of generality assume that $\hat{p} = (p_1, p_2)$ comes before $\hat{q} = (q_1, q_2)$ where p_1, p_2, q_1, q_2 are path-segments with $d(p_1) = d(q_1) = o(p_2) = o(q_2) = i$. Since the route traverses a directed cycle, $q'_1 \neq p_2$.

Since both $\hat{p}$ and $\hat{q}$ are path-segments, we have

$$l(p_1) + l(p_2) \leq d_{max} \quad \text{and} \quad l(q_1) + l(q_2) \leq d_{max} \quad (5.1)$$

We must have

$$l(p_1) + l(q_1) > d_{max}, \quad (5.2)$$

since otherwise, we can replace the portion of the $\mathcal{O}(k)$ – $\mathcal{D}(k)$ path from $o(p_1)$ to $d(q_2)$ with (p_1, q'_1, q_1, q_2) removing all edges on p_2 , the edges on the intermediate segments between $\hat{p}$ and $\hat{q}$ and thus violating our assumption of P^k using the

minimum number of edges from T^* . Note that the case $p_2 = \emptyset$ implies node $i \in R^*$ for which (p_1, q_2) is a feasible route from $o(p_1)$ to $d(q_2)$ again contradicting the minimum usage of edges.

By a similar line of reasoning, we must have

$$l(p_2) + l(q_2) > d_{max}, \quad (5.3)$$

However, inequalities (5.1), (5.2) and (5.3) cannot simultaneously hold.

Lemma 5 has several obvious implications.

Corollary 1. *There exists an optimal solution $\langle R^*, T^* \rangle$ to NDR such that for all $k \in K$, there exists a feasible route P^k with the following properties:*

- (i) *Each node $i \in V$ appears at most twice on P^k ,*
- (ii) *Each arc $a \in A$ appears at most once on P^k ,*
- (iii) *None-consecutive path-segments on P^k are disjoint,*
- (iv) *If two consecutive path-segments say $\hat{p} = (p_1, p_2)$ and $\hat{q} = (q_1, q_2)$ on P^k with $d(p_2) = o(q_1)$ share an intermediate node $i \in V$, with $d(p_1) = d(q_1) = o(p_2) = o(q_2) = i$, then $p_2 = q'_1$.*

We would like to note that Lemma 5 does not hold for DNDR as the example in Figure 5.1a attests to.

5.2.2 Multi-commodity Flow Formulation (MCF)

Considering OD pairs as separate commodities and routing each one from its source to the destination by establishing a feasible route, i.e., a directed path in the virtual layer, gives a multi-commodity flow formulation for NDR that is in essence a modification of the one we have for the RLP-FON in Chapter 4.

We define the following decision variables for this formulation:

$$\begin{aligned} r_i &= \begin{cases} 1, & \text{if node } i \in V \text{ is a relay point,} \\ 0, & \text{otherwise,} \end{cases} \\ w_e &= \begin{cases} 1, & \text{if edge } e \in E \text{ is used in the network design,} \\ 0, & \text{otherwise,} \end{cases} \\ x_p^k &= \begin{cases} 1, & \text{if path-segment } p \in \mathcal{P} \text{ is used by the OD pair } k \in K, \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

We name these variables as *relay*, *edge* and *flow variables*, respectively. Using the same notation defined in Chapter 4, we have $x^k(\mathcal{P}') = \sum_{p \in \mathcal{P}'} x_p^k$ for $\mathcal{P}' \subseteq \mathcal{P}$.

The multi-commodity formulation that we refer to as MCF is:

$$\min \sum_{i \in V} h_i r_i + \sum_{e \in E} c_e w_e \quad (5.4)$$

$$\text{s.t. } x^k(\delta^+(i)) - x^k(\delta^-(i)) = \begin{cases} 1 & \text{if } i = \mathcal{O}(k) \\ -1 & \text{if } i = \mathcal{D}(k) \\ 0 & \text{otherwise} \end{cases} \quad k \in K, i \in V, \quad (5.5)$$

$$x^k(\delta^-(i)) \leq r_i \quad k \in K, i \in V \setminus \mathcal{D}(k), \quad (5.6)$$

$$x_p^k \leq w_{\epsilon(a)} \quad k \in K, p \in \mathcal{P}, a \in p, \quad (5.7)$$

$$r_i \in \{0, 1\} \quad i \in V, \quad (5.8)$$

$$w_e \in \{0, 1\} \quad e \in E, \quad (5.9)$$

$$x_p^k \geq 0 \quad k \in K, p \in \mathcal{P}. \quad (5.10)$$

The objective is to minimize the total relay placement and edge design cost. Constraints (5.5) are flow balance constraints to route each commodity from its origin to its destination by a concatenation of path-segments. Constraints (5.6) ensure that path-segments end at either a relay node or the destination node. Constraints (5.7) are the edge design constraints that enforce that an edge used by an active path-segment should be included in the solution. Finally, constraints (5.8)-(5.10) are the domain restrictions for the variables. Although all decision variables are assumed to be binary, we can relax the integrality requirement for the flow variables since once relay and edge design variables are fixed, flows on

different routes for an OD pair can be consolidated on just one arbitrarily chosen path (among those that carry flow) without any change in the objective function value.

Lemma 5 enables us in strengthening constraints (5.7) and decreasing the size of the formulation. In particular,

Corollary 2. *Let $k \in K$ and $a \in A$. The inequality*

$$\sum_{p \in \mathcal{P}: a \in p} x_p^k \leq w_{\epsilon(a)} \quad (5.11)$$

is an optimality cut for (5.4)-(5.10).

Proof. Corollary 1(ii) states that MCF has an optimal solution where two path-segments employed by the same OD pair do not share an arc. Thus we can sum constraints (5.7) over all path-segments without losing optimality. $\square$

During our experimentations, we replaced constraints (5.7) with constraints (5.11) in model MCF.

Note that even though Lemma 5 is not valid for a DNDR instance, our original MCP model (5.4)-(5.10) can solve the directed network design problem with relays with minor changes.

5.2.3 Solving MCF

We employ a column generation procedure to solve the linear relaxation of MCF that we denote as MCF-LP. To solve MCF we devise a branch-and-price algorithm. Below, we explain how we solve the pricing problem, how we choose the initial subset of flow variables, the branching rule, the search rule and other implementation details.

5.2.3.0.1 Pricing Problem: After solving the restricted MCF-LP that contains only a subset of the flow variables, we look for columns (flow variables) with negative reduced costs. We solve the pricing problem to find such a column or conclude that none exists. Let α_i^k , $-\beta_i^k$ and $-\gamma_a^k$ denote the dual variables associated with the constraints (5.5), (5.6), and (5.11), respectively. Then the reduced cost for a flow variable x_p^k can be calculated as follows:

$$\bar{x}_p^k = \begin{cases} \alpha_{d(p)}^k - \alpha_{o(p)}^k + \sum_{a \in p} \gamma_a^k, & \text{if } d(p) = \mathcal{D}(k) \\ \alpha_{d(p)}^k - \alpha_{o(p)}^k + \sum_{a \in p} \gamma_a^k + \beta_{d(p)}^k, & \text{otherwise.} \end{cases} \quad (5.12)$$

The pricing problem is to check for each $k \in K$ and $(o(p), d(p), p) \in \mathcal{A}$ whether $\bar{x}_p^k$ is negative. Let $A^c = \{(i, j) : (i, j, p) \in \mathcal{A} \text{ for some path-segment } p\}$ denote the set of *plausible* node pairs. Since we have $\gamma_a^k \geq 0$ for all $a \in A, k \in K$ and we look for a negative reduced cost column, the pricing problem for OD pair $k \in K$ and a plausible pair $(i, j) \in A^c$ is actually a CSP instance (as defined in Chapter 4) from node i to node j on a graph $G^k = (V, A)$ in which the resource is $\tau_a = l_a$ and the cost is $\bar{c}_a = \gamma_a^k$ for all $a \in A$ and the resource limit is $\tau = d_{max}$.

In this approach, one needs to solve $O(|K||A^c|)$ CSP problems to solve the pricing problem. This can in fact be done in a more efficient way by solving $O(|K|)$ CSP problems as follows. Consider the pricing graph $\tilde{G}^k = (\tilde{V}, \tilde{A})$, where:

- $\tilde{V} = V \cup \{\tilde{s}\} \cup \{\tilde{t}\}$,
- $\tilde{A} = A \cup \{(\tilde{s}, i) | i \in V\} \cup \{(i, \tilde{t}) | i \in V\}$,
- for arc $a \in A$, the cost is $\bar{c}_a = \gamma_a^k$ and the resource is $\tau_a = l_a$,
- for arc $a \in \tilde{A} \setminus A$, the resource $\tau_a = 0$,
- for arc $(\tilde{s}, i), i \in V$ the cost is $\bar{c}_{\tilde{s}, i} = -\alpha_i^k$,
- for arc $(i, \tilde{t}), i \in V$, the cost is $\bar{c}_{i, \tilde{t}} = \alpha_i^k$ if $\mathcal{D}(k) = i$ and $\bar{c}_{i, \tilde{t}} = \alpha_i^k + \beta_i^k$ otherwise.

Figure 5.3 illustrates a small example for the generation of the pricing graph. For the input graph shown in Figure 5.3a and the dual variable values α , β and γ , the related pricing graph for the OD pair $k = (i, m)$ is depicted in Figure 5.3b.

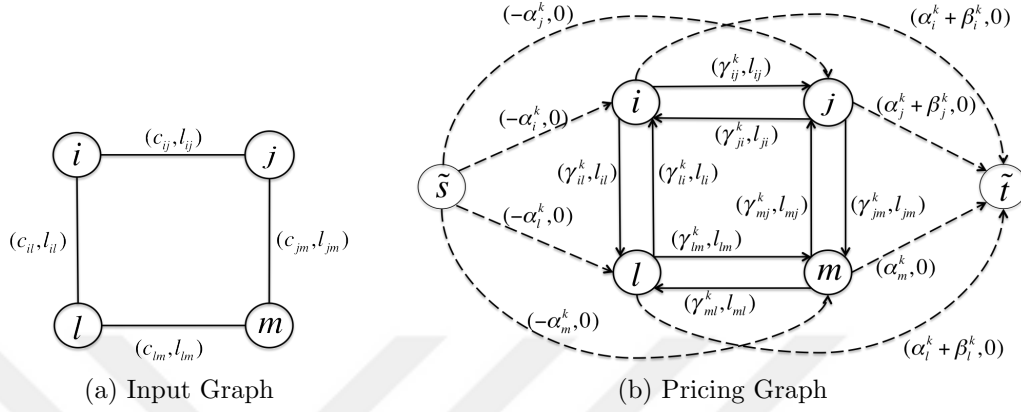


Figure 5.3: Pricing graph corresponding to the detection of negative reduced cost variables. The artificial arcs are depicted with dashed lines.

Observe that if the solution of CSP on the pricing graph for commodity k returns a path $\tilde{p} = ((\tilde{s}, i), p, (j, \tilde{t}))$ with a cost less than 0, then we have $\bar{x}_p^k < 0$ and we can continue to the next iteration of the column generation procedure by adding the new flow variable x_p^k to the restricted problem. On the other hand, if the solution of CSP returns a path with a nonnegative cost for all $k \in K$, then we can safely terminate the column generation procedure and declare the current best solution as an optimal solution for MCF-LP.

Similar to our approach in the previous chapter, we first try to find the negative reduced cost variables by a heuristic approach and resort to the exact solution methodology if the heuristic method fails. For that purpose we use the heuristic algorithm H_q presented in Chapter 4. Recall that, for each plausible pair $(i, j) \in A^c$, H_q stores the first q shortest paths from i to j in G and checks them first to detect a negative reduced cost column. Since H_q requires to solve a q -shortest path problem once in the beginning of the branch-and-price algorithm and there are very efficient algorithms that solve it on graphs with nonnegative lengths [34], this heuristic can drastically improve the performance of the pricing phase in exchange of a quite moderate increase in computer memory cost.

5.2.3.0.2 Initial Variable Pool: Recall that as we discuss in more detail in Chapter 4 path-segments as variables necessitate a more careful approach to determine the initial variable pool of flow variables. So we choose the following initial variable pool to start our branch-and-price algorithm.

Let p_{ij} be the trivial path-segment that contains only the arc $(i, j) \in A$. We can define the initial variable pool as $X_0 = \{x_{p_{ij}}^k | k \in K, (i, j) \in A\}$. Note that, a solution for the MCF-LP, considering only the flow variables in X_0 , contains enough information to derive all the needed dual variable values to properly construct the pricing problem.

5.2.3.0.3 Branching and Search: Since the flow variables are not required to be integral, standard branching rules can be applied to non-integral relay and edge variables without any change in the structure of the pricing problem. In our implementations, we give priority to relay variables and branch on edge variables only when all relay variables are integral. For the search of the branch-and-bound tree we use depth first strategy that speeds up the re-optimization after adding a branching cut.

5.2.3.0.4 Heuristic: As our computational experiments in Chapter 4 the integral solution obtained by solving MCF with only the columns generated in the root node can give heuristic solutions of high quality and can speed up the branch-and-price algorithm significantly. We use the same strategy here and for solving the MCF we resort to this heuristic every 300 nodes if more than 500 columns have been added after our last heuristic attempt. We give a time limit of 1800 seconds to our mixed integer linear program (MIP) solver and consider the best incumbent solution if it reaches to this time limit without a proof of optimality.

5.2.3.0.5 Variable Pool Management: In most column generation/branch-and-price implementations managing the variable pool is crucial. Our preliminary studies have indicated that keeping all the columns in the model and not taking

them out based on the frequency they appear in the optimal bases gave the best performance so we abide by this strategy during all our computational experiments.

We present the results of our computational experiments with this branch-and-price algorithm in Section 5.4.

5.3 Solving NDR-S

Obviously our MCF formulation can be used directly to solve NDR-S. However, as a general solution approach it ignores the special structure of the NDR-S problem, which can be exploited to improve computational efficiency. Facilitated by this special problem structure, we present a more efficient solution approach for NDR-S in this section.

5.3.1 Properties of NDR-S Solutions

Let D be the set of *terminal* nodes, i.e., $D = \{i \in V | \mathcal{O}(k) = i \text{ or } \mathcal{D}(k) = i, k \in K\}$ for a given NDR instance. In the variation NDR-S, there exists a special source node $s \in V$ such that $s = \mathcal{O}(k)$ for all $k \in K$.

NDR-S requires that node s reaches all terminal nodes in $D \setminus \{s\}$. This necessitates the solution to induce a connected graph in the physical network and a rooted connected directed graph in the virtual network (ignoring isolated nodes and edges with zero costs). In fact, we can even show that, there exists an optimal solution inducing a tree in the physical layer and a rooted tree in the virtual layer.

Theorem 9. *Given an NDR-S instance, there exists an optimal solution $\langle R^*, T^* \rangle$ such that T^* is a tree.*

Proof. Proof: Let $T^i \subseteq T^*$ for each $i \in K$ be the set of edges used in the route

P^i from s to i . By Lemma 5, we may assume without loss of generality that T^i is a tree for each $i \in K$. Now we shall constructively show that we can add these trees one by one without creating cycles. Assume $\tilde{T} = \cup_{i \leq m} T^i$ for some $1 \leq m < |K|$ is free of cycles. We shall show that $\tilde{T} \cup T^{m+1}$ can be made cycle free without losing from optimality.

Assume to the contrary that when the edge sets in $\tilde{T}$ and T^{m+1} are combined, we necessarily create a cycle. In other words, visualizing $\tilde{T}$ rooted at s , there exist two nodes u and v such that P^{m+1} touches these two nodes but cannot follow the unique path from u to v in $\tilde{T}$. Among all potential such node pairs u and v , pick one for which the unique paths from u to v in $\tilde{T}$ and T^{m+1} only intersect at these nodes. Without loss of generality we assume that the unique $s - u$ paths in $\tilde{T}$ and T^{m+1} are identical (possibly void corresponding to $s = u$ case).

We know that each node in $\tilde{T}$ (similarly in T^{m+1}), has a feasible route from s otherwise we can remove it along with all its incident edges without violating feasibility. Consider the subtree $\tilde{T}_{uv}$ consisting of the unique $u - v$ path in $\tilde{T}$ as well as any subtree hanging from the nodes on this path. Now, there may be some nodes on $\tilde{T}_{uv}$ that are reachable from s only through routes touching node v . Their routes first have to traverse from u to v , possibly get regenerated through a subtree rooted at v and then come back. Note that since v and all its nodes in its subtree have this property, such nodes do exist. Let W be the set of all such nodes along with all the nodes on their unique subtrees hanging from $\tilde{T}_{uv}$ and $S = \tilde{T}_{uv} \setminus W$. In other words, if node i is a node on the unique $u - v$ path, and if j is a node in its subtree which is reachable from s through v only, any node on the subtree rooted at i (including i itself) will belong to set W . By our construction, (S, W) properly partitions the node set of $\tilde{T}_{uv}$ and there exists a unique edge, say $\tilde{e} \in \tilde{T}_{uv}$, with one endpoint in S and the other endpoint in W .

We now know that s has a route to each $w \in W$ visiting node v . Let $P^w = (P_1^w, P_2^w)$ denote this route and assume v appears once on P_1^w as a terminal node. Let $p(w)$ be the portion of the last path-segment on P_1^w terminating at node v . Similarly, for $\mathcal{D}(m+1)$, consider the route P^{m+1} which passes through v and let q be the portion of the path-segment on this route while visiting v for the first

time. Note that $l_q < l_{p(w)}$ for each $w \in W$ otherwise it would be possible for s to reach $\mathcal{D}(m+1)$ without using the edges on the unique path from u to v on T^{m+1} . But then, s can reach each $w \in W$ by first reaching v following the route P^{m+1} on T^{m+1} and then following the route P_2^w from v to w on $\tilde{T}_{uv}$. So if edge $\tilde{e}$ is deleted no node will lose its reachability from node s and one cycle in $\tilde{T} \cup T^{m+1}$ can be removed. One can repeat these arguments to eliminate all cycles. $\square$

Theorem 10. *Given an NDR-S instance, there exists an optimal solution $\langle R^*, T^* \rangle$ such that the union of path-segments used in the design form a rooted tree in the virtual network Γ with root s spanning nodes in $R^* \cup D$.*

Proof. For each $i \in D \setminus \{s\}$, let P^i be a feasible route from s to i . Consider the subgraph of the virtual network Γ induced by the path-segments used by the routes $P^i, i \in D \setminus \{s\}$. Let $\bar{\Gamma} = (\bar{V}, \bar{\mathcal{A}})$ be this subgraph. By definition of feasible routes, $\bar{V} = D \cup R^*$ so $\bar{\Gamma}$ will be the desired rooted tree if no node in $\bar{V}$ has two incoming arcs from $\bar{\mathcal{A}}$. Assume to the contrary that node $i \in \bar{V}$ has two incoming arcs a_1 and a_2 from $\bar{\mathcal{A}}$. Then node i is necessarily in R^* and any one of the arcs a_1 or a_2 could be removed without violating feasibility. $\square$

In the proofs for Theorems 9 and 10, the optimality of the initial solution will be violated if all the edges have positive costs. In other words, we can say that:

Corollary 3. *Given an NDR-S instance, if $c_e > 0$ for each $e \in E$, then every optimal solution induces a tree in the physical network and a rooted tree in the virtual network.*

5.3.2 Tree Formulation (TF)

Now we are ready to present our Tree Formulation (TF) for NDR-S. Using the results of Theorem 10, we can provide a single-commodity formulation. TF uses the same relay and edge design variables of the MCF formulation but considers

flow variables without commodity superscripts. These variables are called *single-flow variables*:

$$x_p = \begin{cases} 1, & \text{if path-segment } p \in \mathcal{P} \text{ is established,} \\ 0, & \text{otherwise.} \end{cases}$$

For $p \in \mathcal{P}$, let $\mathcal{E}(p) = \{e \in E : e = \epsilon(a) \text{ for } a \in p\}$. We let $x(P') = \sum_{p \in P'} x_p$ and $r(V') = \sum_{i \in V'} r_i$. As defined in Chapter 2, we will use V_s to denote $V \setminus s$.

The tree formulation (TF) is given as :

$$\min \sum_{i \in V} h_i r_i + \sum_{e \in E} c_e w_e \quad (5.13)$$

$$\text{s.t. } x(\mathcal{P}) = |D| + r(V \setminus D) - 1, \quad (5.14)$$

$$x(\mathcal{P}(S)) \leq |S \cap D| + r(S \setminus D) - 1 \quad \begin{matrix} S \subset V: \\ S \cap D \neq \emptyset, \end{matrix} \quad (5.15)$$

$$x(\mathcal{P}(S)) \leq r(S \setminus \{j\}) \quad \begin{matrix} S \subset V: \\ S \cap D = \emptyset, j \in S, \end{matrix} \quad (5.16)$$

$$x(\delta^-(i)) = \begin{cases} 1 & \text{if } i \in D \\ r_i & \text{if } i \in V \setminus D \end{cases} \quad \forall i \in V_s, \quad (5.17)$$

$$r_i \geq \begin{cases} \sum_{p \in \mathcal{P}: (i,j,p) \in \mathcal{A}} x_p + \sum_{p \in \mathcal{P}: (j,i,p) \in \mathcal{A}} x_p & \text{if } i \notin D \\ \sum_{p \in \mathcal{P}: (i,j,p) \in \mathcal{A}} x_p & \text{otherwise} \end{cases} \quad (i,j) \in A^c, \quad (5.18)$$

$$\sum_{p \in \delta^-(i): e \in \mathcal{E}(p)} x_p \leq w_e \quad i \in V, e \in E, \quad (5.19)$$

$$x_p \geq 0 \quad p \in \mathcal{P}, \quad (5.20)$$

$$r_i \in \{0, 1\} \quad i \in V, \quad (5.21)$$

$$w_e \in \{0, 1\} \quad e \in E. \quad (5.22)$$

Similar to the MCF formulation, the objective function is to minimize the network design cost. Note that, by Theorem 10, we know that there exists an optimal solution that induces a rooted tree in the virtual network spanning all terminal nodes and relay nodes. Constraints (5.14)-(5.16) ensure that the union of path-segments form a tree in the virtual network. Constraints (5.17) force that all the terminal nodes are visited in this rooted tree and that a non-terminal node is visited if and only if it is enhanced as a relay. Recall that $\mathcal{A}$ is the virtual

network arc set and A^c is the set of plausible node pairs. Constraints (5.18) are regeneration constraints. They make sure that all the internal nodes of the virtual rooted tree are enhanced with relay capabilities. Note that, these constraints are stronger for the non-terminal nodes for which there can be no incoming or outgoing arcs if they are not chosen as relays. Since all routes start in s , for the sake of simplicity and without loss of generality, we assume that $h_s = 0$ and allow for source node to be chosen as a relay point. Constraints (5.19) are the edge design constraints enforcing all the edges used by established path-segments to be included in the design. Note that instead of writing this constraint separately for each path-segment we can sum over all path-segments sharing the same destination and obtain a stronger inequality. The validity of this inequality is also guaranteed by constraints (5.17). Finally (5.20)-(5.22) are the variable domain restrictions.

5.3.3 Strengthening TF

5.3.3.1 Connectivity cuts for the relays

We seek for an optimal solution that is a rooted tree at node s in the virtual network having relay locations as internal nodes and terminal nodes as leaves. Any terminal or relay location that is not in the relay free communication range of s needs to reach to a relay location within its range. Using this knowledge, we get a tighter formulation by adding the following cuts to TF:

$$\sum_{j:(i,j) \in A^c} r_j \geq \begin{cases} 1 & \text{if } i \in D \\ r_i & \text{if } i \in V \setminus D \end{cases} \quad i \in V_s : (s, i) \notin A^c. \quad (5.23)$$

5.3.3.2 Steiner tree cuts in the physical network

TF seeks an optimal solution where the arcs of the virtual network corresponding to single-flow variables induce a rooted tree and where the design cost of the edge and relay variables is minimum. The correctness of this formulation is ensured by Theorem 10. Due to Theorem 9, we can also restrict the design to induce a tree in the physical network. Since all terminal nodes need to be in this tree for connectivity requirements, this tree is actually a Steiner tree that spans terminal nodes in D . Using this fact, we can generate further optimality cuts to eliminate some fractional solutions.

We obtain a strengthened tree formulation (STF) by adding the relay connectivity cuts (5.23) and the following Steiner tree constraints to TF:

$$w(E) = u(V) - 1, \tag{5.24}$$

$$w(E(S)) \leq u(S \setminus \{j\}) \quad S \subset V, j \in S, \tag{5.25}$$

$$u_i \geq r_i \quad i \in V \setminus D, \tag{5.26}$$

$$u_i = 1 \quad i \in D, \tag{5.27}$$

where we define the *Steiner variables* as:

$$u_i = \begin{cases} 1, & \text{if node } i \in V \text{ is included in the design,} \\ 0, & \text{otherwise.} \end{cases}$$

Constraints (5.24) and (5.25) are the well known Steiner tree constraints where for $S \subset V$, $E(S) = \{\{i, j\} \in E : i, j \in S\}$. Since these constraints provide the complete characterization of the related spanning tree polytope in the case of binary u_i variables [38], they are strengthened by tightening the bounds on the Steiner variables. For this reason we add constraints (5.26) and (5.27) that force all terminal nodes and all non-terminal relay locations to be included in the resulting Steiner tree.

5.3.4 Solving STF

Since STF contains a large number of variables and constraints which exponentially grow with the problem size it is not possible to solve STF directly for realistic size problems and a simultaneous row and column generation is needed. We denote the linear relaxation of STF with STF-LP.

To solve STF-LP, we start with a subset of single-flow variables and add the remaining variables iteratively in a column generation phase. Once there is no variable to add we check for violated inequalities. If we find one we add it to the model and return to the column generation phase again.

During the column generation phase we solve the pricing problem to detect any negative reduced cost variables. For the row generation phase we solve a separation problem to detect any violated inequalities. Here note that, constraints (5.15) and (5.16) could be violated by a solution with integral relay and edge design variables and to be able to properly calculate the reduced costs of the single-flow variables we need to add any violated inequalities after each column generation iteration. Below, we discuss this issue in more detail, explain how we solve the pricing and the separation problems, how we choose the initial subset of flow variables, the branching rule, the search rule and other implementation details.

5.3.4.0.1 Initial Variable and Constraint Pool: For the initial variable pool we use the trivial single-flow variables each associated with an arc in the input network, i.e., we have $X_0 = \{x_{p_{ij}} | (i, j) \in A\}$ as the initial set of columns. For the initial constraints we consider Constraints (5.14), (5.17), (5.18), (5.19), (5.23), (5.24), (5.26) and (5.27) and the relaxed domain restrictions.

5.3.4.0.2 Separation Problem: After solving the initial problem with a subset of variables and constraints, we look for violated inequalities by solving the separation problem. We use the separation procedure proposed by Lee et al. [87], which is designed for a closely related Steiner tree problem, for both the physical and virtual network cycle cancelation constraints. For the sake of brevity, we only explain this separation procedure for the physical network, i.e. separation of constraints (5.25), but it is very easy to see that the same procedure can be readily applied to the virtual network cycles as well.

Let (r^*, w^*, u^*, v^*) be the optimal solution for STF-LP where $V^* = \{i \in V | u_i^* > 0\}$, $E^* = \{e \in E | w_e^* > 0\}$ are the sets of Steiner and edge design variables with positive values. We first generate the separation graph $\hat{G} = (\hat{V}, \hat{A})$, where:

- node set $\hat{V} = V^* \cup E^* \cup \{\hat{s}, \hat{t}\}$,
- arc set $\hat{A} = \{(\hat{s}, e) | e \in E^*\} \cup \{(i, \hat{t}) | i \in V^*\} \cup \{(e, i), (e, j) | e = \{i, j\} \in E^*\}$,
- arc capacities are infinite except for those arcs:
 - leaving from $\hat{s}$, with capacities $\hat{w}_{(\hat{s}, e)} = w_e^*$,
 - reaching to $\hat{t}$, with capacities $\hat{w}_{(i, \hat{t})} = u_i^*$.

To solve the separation problem, for each $j \in V^*$, we change the arc capacity for $(j, \hat{t})$ to zero and solve a maximum flow problem from $\hat{s}$ to $\hat{t}$. If the solution value is less than $w(E)$, the minimum capacity cut gives a violated inequality where S is the set of indices of Steiner variables in the minimum cut and j is the index of the Steiner variable that is excluded in the first sum in (5.25). For more details about the validity of this separation procedure we refer the reader to Lee et al. [87].

Note that the above exact separation procedure requires solving several maximum flow problems. Indeed, imitating the approach we propose to solve separation problem in Chapter 3, we can do better than this by first trying a simple yet quite efficient heuristic before resorting to the exact solution for the separation problem. The heuristic algorithm which we call as the *connectivity heuristic*

(H_C) depends on the following observation. If the separation graph $\hat{G}$ is not connected and none of its connected components contains all the terminal nodes (including s), then we can detect violations as follows. Let $\mathcal{C}$ be the collection of connected components of $\hat{G}$ each including a terminal node. For $S \subset V$, let $\delta(S) = \{e \in E : |e \cap S| = 1\}$. Then the following cuts can be added to the model to remove this infeasible solution.

$$w(\delta(C)) \geq 1 \quad C \in \mathcal{C}. \quad (5.28)$$

It is obvious that Theorem 1 establishes the validity of these cuts for the physical layer and Theorem 2 enables a very similar procedure to be applied in the virtual layer. Thus, using the heuristic H_C , we have the chance to solve the separation problem with a simple graph search in many cases especially at the beginning of the algorithm where a very limited number of constraints are in the model.

5.3.4.0.3 Pricing Problem: After the row generation (cutting plane) phase ends, we look for columns (single-flow variables) with negative reduced costs to add to the model. We solve the pricing problem to find such a column or conclude that none exists. Let $-\alpha, -\beta, -\gamma, -\lambda, \theta, -\mu$ be the dual variables associated with constraints (5.14) -(5.19), respectively. Since we are looking for a tree rooted at s , the reduced cost $\bar{x}_p$ of a flow variable x_p is infinity if $d(p) = s$ and if this is not the case it can be calculated as follows:

$$\bar{x}_p = \begin{cases} \alpha + \sum_{\substack{S \subset V, \\ S \cap D \neq \emptyset \\ o(p), d(p) \in S}} \beta_S + \sum_{\substack{S \subset V, \\ S \cap D = \emptyset \\ o(p), d(p) \in S}} \sum_{j \in S} \gamma_S^j + \lambda_{d(p)} + \theta_{o(p), d(p)} + \sum_{e \in \mathcal{E}(p)} \mu_{d(p)}^e, & \text{if } d(p) \in D, \\ \alpha + \sum_{\substack{S \subset V, \\ S \cap D \neq \emptyset \\ o(p), d(p) \in S}} \beta_S + \sum_{\substack{S \subset V, \\ S \cap D = \emptyset \\ o(p), d(p) \in S}} \sum_{j \in S} \gamma_S^j + \lambda_{d(p)} + \theta_{o(p), d(p)} + \theta_{d(p), o(p)} + \sum_{e \in \mathcal{E}(p)} \mu_{d(p)}^e, & \text{otherwise.} \end{cases} \quad (5.29)$$

Note that, as a complicating factor, the above calculation involves some dual variables associated with constraints that are not in the current model. Fortunately, this does not cause a problem since we can set the values of the dual variables associated with the constraints that are not in the model to zero. Here note that, since we start with all the trivial single-flow variables and do not remove any of them throughout the algorithm, addition of the Steiner cuts does not cause infeasibility and reduced cost calculations are not adversely affected.

Recall that we are looking for negative reduced cost variables and we have $\mu_{d(p)}^e \geq 0$, for each edge $e \in E$ and path-segment $p \in \mathcal{P}$. Thus, the pricing problem for each plausible pair $(i, j) \in A^e$ is actually a CSP instance from node i to node j on original network $G = (V, A)$ in which the resource for each arc $a \in A$ is l_a , the cost for each arc $a \in A$ is μ_j^e and the resource limit is d_{max} . As we have discussed during the presentation of the branch-and-price algorithm for the MCF formulation in subsection 5.2.3, we first try the H_q heuristic to detect any negative reduced cost variables and resort to the exact solution only when H_q cannot find one.

5.3.4.0.4 Implementation Details: For the branch-and-price-and-cut, we use essentially the same branching and search strategies previously discussed for the branch-and-price in subsection 5.2.3 and for the sake of brevity we do not repeat the same explanations here. For the variable and constraint pool management our preliminary studies have shown that our branch-and-price-and-cut algorithm performs best when we do not remove any column or row. During the branch-and-price-and-cut algorithm we search for violated inequalities in the root node, in the last branch-and-bound node left in the queue and when we find an integral solution. For the cut generation, except for the case we have an integer solution at hand, we terminate it when the increase in the objective function value is less than 0.01 after the inclusion of the last set of cuts.

5.3.4.0.5 Heuristic: Mimicking the heuristic solution approach we have for the branch-and-price algorithm, at the root node of the branch-and-bound tree, at each iteration of the column generation phase we check whether the solution for the current restricted STF-LP is integral or not. If it is integral, there is no violated constraint and this solution has a lower cost than the best integer found so far then we store it as the best integer solution. If no such solution is found, we solve STF as an integer program with only the columns generated at the root node. To solve this restricted STF more efficiently, we employ a branch-and-cut approach in which the violated constraints are added iteratively using the same separation procedure we presented above. Our numerical experiments have shown

that the integral solution obtained by this procedure is often of high quality and can speed up the branch-and-price-and-cut algorithm significantly.

Note that we can also solve NDR-S by solving the TF formulation with a branch-and-price-and-cut algorithm that is similar to the one presented for STF. With an aim to assess the benefits of using the information that the solution of the problem is a tree in the input graph, we consider this simpler algorithm in our computational experiments as well.

5.3.5 Enhancing MCF for NDR-S problems

When we consider NDR-S problems, we can add the Steiner tree cuts to the MCF formulation as well and obtain a stronger formulation. Let SMCF be the MCF formulation strengthened with the Steiner tree cuts. We need a branch-and-price-and-cut algorithm to solve this new formulation. Since the Steiner tree cuts for MCF do not include the flow variables, the pricing problem is not affected by the inclusion of these cuts. As a result, with a slight modification of the branch-and-price algorithm explained in detail in Section 5.2.3, it is straightforward to implement this new branch-and-price-and-cut algorithm and for the sake of brevity we do not provide the details here.

5.4 Numerical Experiments

Comprehensive numerical experiments are conducted to test the performances of the algorithms proposed in this chapter. Two common network topologies from the literature are considered: grid networks by Cabral et al. [17] and Steiner problem instances B and C from the OR-Library [13]. We implemented all the algorithms using Java under Linux and CPLEX 12.5 and all experiments are done on the same machine: Intel 2 Xeon E5-2609 4C 2.4GHz CPU and 8GB RAM.

5.4.1 Grid Networks

Since NDR problem is first introduced by Cabral et al. [17], most of the literature on this problem considers the instances presented in this study. To test our solution algorithms on these commonly used NDR-S instances we imitated the random problem generation procedure described in Cabral et al. [17]. These instances have a grid structure with a rows and b columns. The costs and lengths of the edges are randomly generated from a uniform distribution $U[10, 30]$. The randomly chosen OD pairs all share the same origin node that is chosen randomly, i.e., Cabral instances are NDR-S instances. For these problem instances $d_{max} = 70$ and the relay costs are randomly chosen from a uniform distribution $U[70, 140]$.

Before proceeding with the computational results, we first look at NDR-S solutions on some small size network instances to show the complexity of the problem and to illustrate the correspondence between the solutions in the physical and virtual networks. These examples clearly show that solving NDR in a sequential manner, i.e., first solving a Steiner tree problem and then placing regenerators in the resulting design, may not produce high quality solutions. In Figure 5.4, we consider a 5×10 grid network with 10 terminal nodes. Figures (5.4a)-(5.4c) depict the NDR-S solutions on the input graphs for d_{max} values of 30, 50 and infinity, respectively. The dashed lines show the edges present in the grid graph, whereas the thick(red) lines show those edges included in the optimal solution. The terminal nodes are shown with double circles and nodes enhanced with relay capabilities are highlighted (depicted in turquoise color). The dashed circles show the nodes in the grid graph that are not included in the design. Figures (5.4d)-(5.4f) illustrate the respective solutions in the virtual networks as rooted trees. These instances clearly show that the optimal edge design may heavily depend on the locations of the relays. Note that when $d_{max} = \infty$ our problem reduces to a Steiner tree problem and Figure (5.4c) shows the minimum cost Steiner tree solution. This design is quite different than the design for the instance with $d_{max} = 30$. As we see in this small example, the tree structure generated in the physical network layer is significantly affected by the relay placement decisions.

Therefore, when the relay costs are not negligible and d_{max} values are restrictive, solving NDR problem by jointly solving relay placement and network design problems can provide much different results than a sequential solution approach. As a result, depending on the cost structure of the given problem instance, the difference between such an heuristic solution's objective function value and the true optimum can get arbitrarily large.

We first compare the computational performances of the solution approaches we propose in this part of the study. Table 5.1 shows the results of numerical experiments for instances with up to 60 nodes and 40 OD pairs. In the table, a and b are the grid dimensions and $|K|$ is the number of OD pairs. For each setting, five random instances are created and the average solution times are reported for each solution method, which are characterised by their formulation of the problem. The first four columns show the run times in seconds for the respective algorithms that solve MCF, SMCF, TF and STF formulations and the last three columns report the ratios of the solution times of algorithms for MCF, SMCF and TF to those of STF. A time limit of 7200 seconds is given to all the algorithms. For all instances, H_6 heuristic is used to solve the related pricing problem in the branch-and-price and branch-and-price-and-cut algorithms.

Looking at Table 5.1, we immediately notice that including Steiner cuts drastically improves the performances of the solution procedures both for the multi-commodity flow and tree formulations. For the multi-commodity flow formulations, solution times with SMCF can on the average improve more than 20 times, whereas this ratio increases to 100 when we consider tree formulations. Comparing the computational performances all these solution methods, the results presented in Table 5.1 clearly show that the branch-and-price-and-cut algorithm of STF is superior to other algorithms. The last three columns present that, on the average, this algorithm can be more than 461, 67 and 136 times faster than the algorithms that solve MCF, SMCF and TF, respectively. Note that these ratios could be much larger if we had not have the time limit and waited all algorithms to finish. Such a drastic performance superiority can be explained by two major advantages STF has over other algorithms. The first one, that it shares with TF, is the single flow approach for which the increase in the number

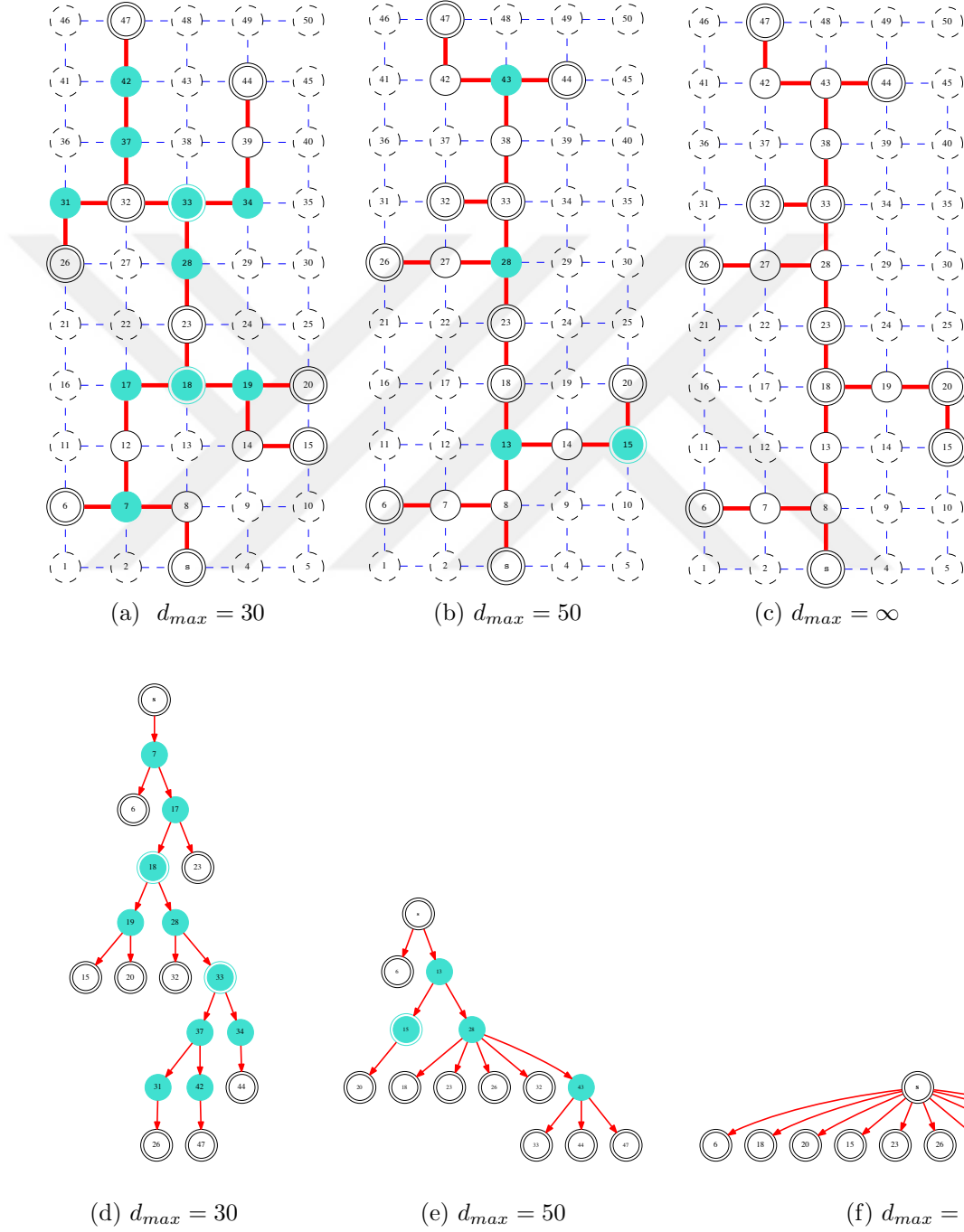


Figure 5.4: Grid graph solution illustrations

of OD pairs in the problem does not increase the number of variables whereas, number of variables in the multi-commodity flow formulations depends directly on the number of OD pairs. As a result, while the number of OD pairs grows, the run times for STF do not get as much adversely affected as those for the multi-commodity flow formulations. The second advantage is the inclusion of the Steiner cuts, which it shares with the SMCF. The inclusion of these cuts results in a much tighter formulation that can be solved more efficiently as we have discussed at the beginning of this paragraph.

As Table 5.1 indicates, the performance of the branch-and-price-and-cut algorithm for STF is quite satisfactory for small and medium size problems. To test its performance on larger size problems we considered grid networks with up to 100 nodes and 80 OD pairs. For each problem setting we have five random instances. Since none of the other formulations are able to scale up to these larger problem instances, we only report the solutions for STF in Table 5.2. In this table, the column *Sol.* is the objective function value of the best integer solution found by the algorithm whereas *H.Sol* is the heuristic solution we obtain at the root node. *Root.Lp* is the lower bound we obtain at the root node at the moment we start branching. *RT* is the run time of the algorithm in seconds until it finds an optimal solution or the time limit of 7200 seconds is reached. The column *#Opt* shows the number of instances solved to optimality within the given time limit and $Gap = \frac{Sol - BestBound}{Sol}$ is the optimality gap calculated for each instance, where *BestBound* denotes the best lower bound determined by the STF algorithm when it stops. The column *BBN* shows the number of branch-and-bound nodes explored by the algorithm whereas *Cols* and *Cuts* columns show the total number of columns and rows generated, respectively, throughout the solution procedure.

As we see in Table 5.2, for most of the instances, the branch-and-price-and-cut algorithm solves STF to optimality and the optimality gaps are quite small for the instances for which the algorithm stopped due to the time limit. As we see in the table, the linear relaxation of the problem provides a quite tight lower bound which indicates the strength of the STF and accounts for the high computational efficacy. We can also see that the heuristic continues to provide very high quality

Table 5.1: Solution time comparisons for the proposed algorithms

a	b	$ K $	Solution Time (sec)				Solution Time Ratios		
			MCF	SMCF	TF	STF	MCF	SMCF	TF
5	5	5	0.9	0.6	1.0	0.7	1.3	0.9	1.4
		10	2.6	1.3	1.2	0.7	3.7	1.8	1.7
		20	115.7	15.4	19.3	1.4	81.8	10.9	13.7
6	5	5	1.2	1.0	2.0	1.1	1.1	1.0	1.9
		10	8.9	3.5	5.9	1.6	5.7	2.3	3.8
		20	285.7	12.8	35.8	1.7	170.3	7.6	21.3
7	5	5	2.0	2.3	3.8	2.4	0.8	0.9	1.6
		10	18.3	6.4	10.1	2.3	7.9	2.8	4.4
		20	448.5	31.1	55.1	3.0	148.7	10.3	18.2
		30	4213.5	230.5	387.8	9.1	461.2	25.2	42.4
8	5	5	2.2	1.9	5.8	3.5	0.6	0.5	1.7
		10	51.9	14.6	45.0	7.0	7.4	2.1	6.4
		20	735.5	39.8	145.5	6.6	110.7	6.0	21.9
		30	> 7200.0	859.7	2376.7	17.9	> 402.9	48.1	133.0
9	5	5	3.1	5.6	20.3	6.1	0.5	0.9	3.3
		10	53.1	23.5	65.9	8.4	6.3	2.8	7.8
		20	1786.8	136.2	682.2	17.1	104.8	8.0	40.0
		30	> 7200.0	220.8	2148.6	19.1	> 376.2	11.5	112.3
10	5	5	3.1	3.3	58.5	40.6	0.1	0.1	1.4
		10	54.2	15.2	149.5	44.4	1.2	0.3	3.4
		20	1483.0	65.7	1011.4	70.3	21.1	0.9	14.4
		30	> 7200.0	317.9	3564.4	43.4	> 165.9	7.3	82.1
		40	> 7200.0	851.9	> 7200.0	63.7	> 113.1	13.4	> 113.1
11	5	5	11.6	18.6	675.8	42.9	0.3	0.4	15.8
		10	98.1	28.6	484.0	43.9	2.2	0.7	11.0
		20	2072.6	108.9	1881.9	28.7	72.1	3.8	65.5
		30	> 7200.0	1104.0	5819.3	122.0	> 59.0	9.0	47.7
		40	> 7200.0	3968.6	> 7200.0	226.3	> 31.8	17.5	> 31.8
12	5	5	5.0	8.9	294.2	40.6	0.1	0.2	7.2
		10	58.6	32.1	712.0	44.4	1.3	0.7	16.0
		20	2684.6	166.1	3187.7	70.3	38.2	2.4	45.3
		30	> 7200.0	2926.0	5938.2	43.4	165.9	67.4	136.8
		40	> 7200.0	802.4	> 7200.0	63.7	> 113.1	12.6	> 113.1

solutions with these larger size problems. From Table 5.2 we can infer that the problem gets harder when the number of OD pairs increases. This is expected since larger network designs emerge with the increasing number of OD pairs opening room for more alternative solutions to consider. It is interesting to see that while the number of explored branch-and-bound nodes drastically increases with the number of OD pairs, the number of columns and rows generated do not increase as much. This is mainly due to the fact that when the number of OD pairs increases the number of alternative relay placements increases and finding an integral solution becomes much harder even though solution times for STF-LP do not increase so much.

Table 5.2: Results summary for STF solutions on larger size problem instances

a	b	K	Sol.	H.Sol	Root.Lp	RT	#Opt	Gap	BBN	Cols	Cuts
8	8	5	446.40	446.40	397.96	23.57	5	0.00	44.60	2590.40	561.60
		10	597.20	597.20	550.02	19.52	5	0.00	42.60	2838.20	535.80
		20	841.80	841.80	781.11	56.10	5	0.00	100.20	3115.60	875.20
		30	979.00	981.20	896.45	99.84	5	0.00	223.00	3216.20	1002.80
		40	1168.80	1170.40	1074.40	494.94	5	0.00	1410.20	3275.60	1450.40
		60	1428.00	1432.00	1326.44	3110.17	4	0.02	13643.80	3343.40	1514.60
9	9	5	593.20	593.20	546.53	100.98	5	0.00	44.60	3782.60	1474.20
		10	755.00	755.40	678.49	219.75	5	0.00	181.00	4145.80	1286.80
		20	998.20	1002.40	884.98	689.04	5	0.00	647.80	5026.00	1329.00
		30	1192.20	1192.80	1065.39	1135.57	5	0.00	1130.60	4939.00	1820.60
		40	1323.00	1323.00	1196.43	2149.40	5	0.00	2488.20	4882.00	1938.00
		60	1633.20	1634.80	1486.86	6680.92	1	0.06	11118.00	4914.20	1507.60
10	10	5	513.80	513.80	454.61	208.85	5	0.00	71.80	4109.00	1952.00
		10	807.00	816.40	720.54	936.79	5	0.00	277.80	5368.80	2280.00
		20	1078.40	1086.60	960.46	2259.04	5	0.00	1402.20	5922.00	2264.40
		30	1361.25	1362.50	1218.53	4198.15	4	0.02	1882.25	5620.75	2434.25
		40	1548.40	1553.00	1366.52	6373.17	1	0.09	3885.40	5998.40	1729.80
		60	1786.00	1786.00	1605.69	6053.07	1	0.09	5050.60	6022.80	1452.40
		80	2068.60	2068.60	1823.37	7201.20	0	0.12	10563.80	5969.60	967.00

5.4.2 Steiner Problems form OR-Library

As we have already argued, NDR problems are closely related with the Steiner tree problems, which are extensively studied in the literature. In order to further

test and evaluate the computational performance of our branch-and-price-and-cut algorithm for STF on a quite different network topology than we have for the grid networks, we also consider the Steiner problem instances B and C from the OR-Library [13]. These network topologies are considered for other extensions to the basic Steiner tree problem such as the Steiner tree problem with revenues, budget and hop constraints [32].

We abide by the cost structures and reach limitations of the Cabral instances when we generate random NDR-S instances from these network topologies. The root node s is chosen as the smallest number node in the list of terminals provided for each problem instance. In solving the pricing problems, we use H_{15} and H_5 for the group of problems B and C, respectively.

Table 5.3 summarizes the STF solutions with the branch-and-price-and-cut algorithm for these instances. For each problem five random instances are considered. As we see in the table, STF can solve problem instances with up to 500 nodes and 1000 edges. Most of the problems are solved to optimality and optimality gaps are quite low for the unsolved problems. Confirming the results we get with the grid networks solutions of Table 5.3, we can see that the linear relaxation of the problem provides quite tight bounds for these large size problems. Similarly we see that the number of OD pairs is a significant dimension of the problem that affects the performance of the algorithm. It is also interesting to see that solving the problem with just the columns generated at the root node can give quite good heuristic solutions even for these large size instances except for the ones we have in SteinC number 5. In that problem setting, for two random instances, no integral solution is detected during the column generation and CPLEX was not able to find a feasible solution for the respective MIP model in the given time limit of 1800 seconds. So we read NA in the table for the average of heuristic solution values for this set of instances.

Table 5.3: TSF Algorithm results for OR library Steiner tree instances

Prob.	Num	V	E	K	Sol.	H.Sol	Root.Lp	RT	#Opt	Gap	BBN	Cols	Cuts
SteinB	1	50	63	9	524.80	524.80	510.28	1.49	5	0.00	3.80	1217.80	79.80
	2			13	576.80	576.80	561.49	1.87	5	0.00	6.60	1249.40	90.00
	3			25	858.80	876.00	855.47	1.41	5	0.00	4.20	1424.60	49.80
	4		100	9	355.60	355.60	343.77	6.94	5	0.00	7.40	5307.20	224.80
	5			13	410.80	410.80	406.13	2.84	5	0.00	4.20	4158.20	43.60
	6			25	641.80	645.80	636.60	4.46	5	0.00	5.40	4544.40	139.20
	7	75	94	13	622.00	622.00	607.57	2.90	5	0.00	9.80	1992.60	85.60
	8			19	689.00	689.00	665.97	4.21	5	0.00	7.40	1951.80	151.00
	9			38	1239.40	1259.20	1209.75	12.58	5	0.00	99.00	1837.20	253.80
	10		150	13	490.60	490.60	482.99	10.05	5	0.00	6.20	6735.60	235.00
	11			19	597.40	597.40	581.97	24.87	5	0.00	11.40	7018.60	365.20
	12			38	868.20	872.00	853.45	24.82	5	0.00	11.80	8441.00	348.80
	13	100	125	17	847.80	847.80	845.45	2.57	5	0.00	3.40	2028.80	59.00
	14			25	1178.20	1178.20	1160.37	5.60	5	0.00	9.40	2110.60	320.60
	15			50	1665.60	1682.00	1637.44	41.14	5	0.00	206.60	2874.80	667.60
	16		200	17	660.00	673.00	632.39	31.15	5	0.00	18.20	7228.00	459.40
	17			25	829.40	830.20	791.34	92.17	5	0.00	31.40	10879.00	783.20
	18			50	1221.20	1230.60	1182.73	458.77	5	0.00	86.60	11470.80	1215.20
SteinC	1	500	625	5	697.80	697.80	609.30	960.05	5	0.00	44.60	14707.00	1282.20
	2			10	905.40	908.20	831.49	3851.64	3	0.02	91.80	17618.80	1690.00
	3			83	4579.40	4582.00	4214.08	7200.00	0	0.08	772.20	15461.20	1193.80
	4			125	5904.40	5907.00	5595.56	7200.00	0	0.05	418.80	16529.80	2543.80
	5			250	9247.00	NA	8574.24	7200.00	0	0.07	433.60	18392.00	3417.20
	6	500	1000	5	337.00	337.00	329.01	2743.11	5	0.00	5.00	50299.80	2791.60
	7			10	642.20	651.80	631.37	2786.11	5	0.00	5.40	58781.00	1795.20
	8			83	3198.60	3198.60	2955.33	7200.00	0	0.08	21.60	42982.40	2617.80

5.5 Summary

In this chapter, we study the network design problem with the relay placement perspective and introduce efficient solution algorithms to solve this relatively new and challenging network design problem. The literature on network design problems is quite extensive but even though relays are common in many telecommunication and transportation applications, there are very few studies that consider relays as part of the network design. This part of our dissertation attempts to fill this gap in the literature.

Network design problems are challenging problems and incorporating relay placement decisions complicates them further by allowing paths with cycles to connect OD pairs in the optimal solutions. In order to address this challenging problem, we propose a multi-commodity flow based formulation and a branch-and-price algorithm to solve it by using the path-segment notion we present in Chapter 4 and introduce to the literature by Yildiz and Karasan [153] and Yildiz

et al. [155]. Our path-segment formulations enable quite efficient representation of the reach constraints in the model and provide opportunities to devise graph transformations and novel algorithmic approaches to improve the performance of our branch-and-price procedure. As a result, we succeeded in devising a general algorithm whose computational performance is drastically better than the exact solution approaches that exist in the literature.

With both practical and theoretical motivations, we pay a special attention to NDR-S. As one of the main contributions of this chapter, we show special properties satisfied by a subset of optimal solutions for this interesting problem. Then we propose a novel tree formulation and a branch-and-price-and-cut algorithm that exploits this special structure. Considering the special properties of NDR-S exclusively and tailoring our algorithm respectively, we achieve a computational efficiency that outperforms the state of the art heuristics in the literature. The flexible structure of the novel tree formulation and the impressive performance of the branch-and-price-and-cut algorithm we propose to solve it suggest a good potential to adapt this approach to address other location and routing problems that emerge in quite different contexts. One such example is the refueling station location problem for the alternative fuel vehicles which is the topic of the next chapter.

Chapter 6

Refueling Station Location Problem for Alternative Fuel Vehicles

Different then the relay location problems studied in the previous chapters, the total length of the routes that connect each origin destination pair is an important consideration when it comes to vehicle trips instead of transmitting signals. In this chapter we extend our solution approaches to capture this significant aspect and present efficient solution algorithms to solve Refueling Station Location Problem for Alternative Fuel Vehicles. The results of our studies presented in this chapter has been contributed to the literature by Yildiz et al. [155].

In the existing literature, different modeling approaches are used to locate the refueling stations. Early studies in this area [47, 102, 103] utilized the p-median model to minimize the sum of the travel times from the demand sites (i.e. homes) to the nearest refueling facilities. The motivation behind p-median models is that the vehicle owners usually prefer to refuel close to their homes [74, 134]. The p-median approach assumes that the demand is located at nodes. A different approach to the refueling station location problem considers path-based demand. This idea is initially presented in flow capturing location model (FCLM)

by Hodgson [53] and in flow intercepting location model (FILM) independently by Berman et al. [14]. A path-based demand is considered to be ‘captured’ if the path contains a node with an open facility. In other words, a single facility is assumed to be enough to cover the whole flow on the path. The objective is to locate p facilities while covering as many path flows as possible. This flow capturing logic cannot directly be applied to the refueling station location problem of the alternative fuel vehicles since a vehicle might need to be refueled more than once to travel between an origin-destination (O-D) pair. To handle this, Kuby and Lim [80] introduce Flow Refueling Location Model (FRLM) that locates p refueling stations to maximize the total refueled flow volume while making sure that the vehicles never run out of fuel. Similar to FCLM, the demand is defined as a flow on the shortest path between an O-D pair. But this time, rather than a single facility, a certain set of stations enabling the round trip of the vehicle between an O-D pair is required. In other words, a ‘combination of facilities’ is needed to serve the demand so that the vehicles do not run out of fuel while traveling. In the initial phase of the two-stage solution methodology, feasible minimal combinations that can refuel the shortest path between each O-D pair are determined by a preprocessing algorithm. These combinations are given as input to a mixed integer linear programming (MILP) formulation in the second stage. In the FRLM, at least a half-full tank of fuel is required at the final destination with no refueling station [25, 80–82, 99]. This enables the vehicle to have enough fuel to complete a round trip. If a refueling station is located at the destination node, the half-full tank requirement is relaxed. This is a very realistic assumption since no AFV driver would like to reach the destination without enough fuel to visit a refueling station on the return trip. With the same reasoning, a similar assumption is made for the origin nodes. This basic FRLM formulation is extended from different aspects and some assumptions are relaxed in further studies. The objective function is modified to maximize the total vehicle-miles traveled [82]. The feasible set of candidate sites for refueling stations is extended from the node set to include the points on the arcs as possible location points by Kuby and Lim [81].

The FRLM requires the generation of all combinations for all the path-based

demands. Thus, building the model for even medium-sized networks requires excessive time. In order to overcome this drawback, Lim and Kuby [91] propose three heuristic algorithms: greedy-adding, greedy-adding with substitution and genetic algorithm. In a similar line of efforts, a different refueling logic is embedded into the MILP model by Capar et al. [25]. The authors propose a simple, yet powerful formulation that solves the FRLM to optimality in reasonable amount of time.

Different approaches such as set covering are also studied in the recent literature [99, 143–145]. Rather than locating p facilities to serve the demand, a set covering approach finds the minimum-cost combination of facilities to serve all of the O-D demand pairs. MirHassani and Ebrazi [99] approach this problem from a different perspective to increase the size of the problems that can be solved to optimality. Initially building an expanded network, the problem is solved by a mixed-integer linear programming (MILP) formulation. They do not consider flow deviation (driver preferences) and assume a fixed simple path, namely, the shortest path, between each O-D pair. With the fixed path assumption, the resulting MILP formulation can be directly solved by any commercial solver for realistic problem instances.

All of the aforementioned studies consider only a fixed number of simple paths to connect O-D pairs. Although these assumptions make problems computationally tractable, they unnecessarily restrict the solution space. It is clear that considering only a small portion of all possible paths can result in a suboptimal solution. For the simple path case, consider the example depicted in Figure 6.1. In order to cover both demands between O-D pairs o_1 - d_1 and o_2 - d_2 , two stations located at nodes A and C are required if we only consider simple paths. However, if non-simple paths are viable, a single refueling station located at node B would cover both demands.

The effects of driver preferences such as deviating from the shortest paths is a significant factor on travel costs [7]. However, there are very few studies in the refueling station location literature that incorporate this significant factor into the location decisions. In this context, Kim and Kuby [71] study simple-path

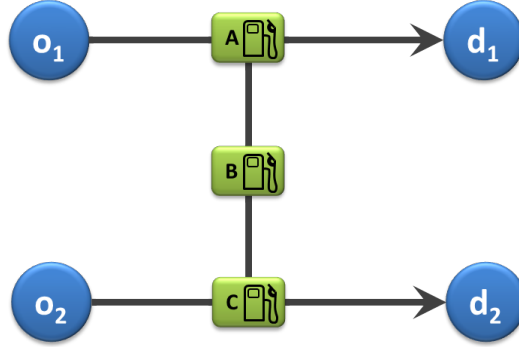


Figure 6.1: Non-simple path example

deviations (i.e. cycles are excluded) from the shortest paths. The deviations are calculated by a k -shortest path algorithm before the model is solved until a predefined user tolerance deviation is reached. The deviation is defined as the percentage difference of the selected route and the shortest path. Similar to FRLM, the preprocessing time in this Deviation Flow Refueling Location Model (DFRLM) is excessive when deviations are considered. Therefore Kim and Kuby [72] propose a network transformation heuristic to solve realistic-sized problems.

The DFRLM assumes that the demand decays by increasing deviation from the shortest distance. In their study, Kim and Kuby [71] defines the decay as a function of the deviation. Observe that the function of the demand is hard to obtain and “the exact shape and steepness may need to be calibrated to empirical survey data” [71]. Empirical studies usually consider bins that specific values fall into. Rather than fitting a function from an empirical data, one can easily define different demands for the same O-D pair with different deviation tolerances. This increases the flexibility of the solution technique. This way, there is no need to define a function and the empirical data can directly be used without sacrificing from data quality due to fitting errors. As we present next, our solution approach is quite flexible to incorporate this approach to model driver preferences.

In a similar context, routing is considered in a recent study by Kang and Recker [62]. In order to account for the routing decisions of the drivers, household activity pattern problem (HAPP) [118] is used, which is a variation of the pickup

and delivery problem with time windows. The authors consider the routing decisions of the individuals in a metropolitan area and simultaneously optimize the scheduling and routing decisions of the households as well as the location of the refueling stations. The limited range of the vehicles is not considered in this study. Instead, it is presumed that each household visits a refueling station once in a day either on the way to another activity or as a single trip.

6.1 Mathematical Model

In this section we formally define the refueling station location problem with routing (RSLP-R) and present the chapter specific notation.

6.1.1 Definitions and Notation

An AFV trip has three components: vehicle, O-D pair and driver. For each trip, the fuel range (the maximum distance to be covered with a full fuel tank) is a function of vehicle specifications, the O-D pair indicates where the trip starts and ends and the driver preference determines how much extra driving can be tolerated by this driver. From a macroscopic view, those trips with the same vehicle, O-D pair and driver preference can be considered a single group which we call as a demand. The flow volume of a demand is given proportional to the amount of AFV trips. For each demand there is an associated traffic volume which is a function of the number of AFV trips in the considered time interval. Following the convention established in the literature, we assume that all the alternative fuel vehicle trips start with half full tank. A trip is considered to be feasible for a given demand if it satisfies the following three conditions:

- It starts from the origin and ends in the destination node,
- There are enough refueling stations positioned on the path such that it is possible to travel without running out of fuel and arrive to the destination

with at least half full fuel tank,

- Its length is not more than the threshold value that the AFV driver can tolerate.

A given demand is considered to be refueled if the designed road network contains a feasible trip for it. In RSLP-R, the objective is to find the locations of a fixed number of refueling stations in the network such that the total volume of the refueled demand is maximized.

We now present the notation we use in the remainder of this chapter. For this chapter, the weighted graph $G = (V, E, l)$ represents a road network instance. Let $S, T \subseteq V$ be the sets of origin and destination nodes, respectively. We define the expanded network $\overline{G} = (\overline{V}, \overline{A})$ where:

- $\overline{V}$ contains nodes $\bar{i}$ for all $i \in S$ and $\bar{j}$ for all $j \in T$ in addition to the original set of nodes V .
- $\overline{A}$ consists of all the arcs in A plus the zero-length arcs $a_i = (\bar{i}, i)$ for all $i \in S$ and $a_j = (j, \bar{j})$ for all $j \in T$.

Between two nodes $s, t \in \overline{V}$, the shortest distance in $\overline{G}$ is denoted by $\delta_{s,t}^*$.

We define M as the set of vehicle types. Similar to the case with the different modulation levels with different optical reach in Chapter 4 different vehicle types have different ranges which determines the refueling free distance limit. The range of a vehicle $\mu \in M$ is denoted by Δ^μ . A demand $k \in K$ is a five tuple $\langle m^k, \mathcal{O}(k), \mathcal{D}(k), \lambda^k, f^k \rangle$, where $m^k \in M$ is the vehicle type and $\mathcal{O}(k) = \bar{i}$ and $\mathcal{D}(k) = \bar{j}$ are the artificial origin and destination nodes associated with the O-D pair $i \in S, j \in T$. $\lambda^k \geq 0$ represents the maximum distance that the driver would accept to travel and f^k is the flow volume. The set of demands is denoted by K .

In this chapter we consider a path-segment p as a directed simple path with an associated demand $t(p) \in K$. In our formulations, we only consider path-segments

with total length less than the range of the vehicle type associated with it and call such path-segments feasible. More formally, a path-segment p is feasible if $l(p) \leq \Delta^{m^{t(p)}}$. We define $\mathcal{P}^k$ as the set of all those feasible path-segments for a demand $k \in K$ and denote the set of all the feasible path segments as $\mathcal{P}$ as usual.

To make it a better fit for the problem context we call routes as *trips* in this chapter and say that a trip is feasible for a demand $k \in K$, if $o(p^1) = \mathcal{O}(k)$, $d(p^q) = \mathcal{D}(k)$, $l(P) = \sum_{i=1, \dots, q} l(p_i) \leq \lambda^k$ and a refueling station is located at $d(p^i)$, $\forall i = 1, \dots, q - 1$.

For a given node set $R \subseteq V$, let $\overline{K}_R \subseteq K$ be the set of demands for which there exists a feasible trip in $\overline{G}$ when a refueling station is located at every node in R . Then, RSLP-R can be formally stated as follows:

Definition 4. *The Refueling Station Location Problem with Routing (RSLP-R) is defined as finding a set $R^* \subseteq V$ with cardinality at most ρ such that the total amount of flow refueled $\sum_{k \in \overline{K}_{R^*}} f^k$ is maximized.*

Proposition 3. *RSLP-R is NP-Complete.*

Proof. Hodgson [53] established the fact that the FCLM is NP-Complete. With the following obvious relationships:

$$\text{FCLM} \prec \text{FRLM} \prec \text{DFRLM} \prec \text{RSLP-R}$$

where $\prec$ is used as a ‘special case’ operator, RSLP-R is also NP-Complete. $\square$

6.1.2 Path-Segment Formulation for RSLP-R (*PSF*)

In this subsection we present the path-segment formulation *PSF* for RLSP-R and provide the details of the proposed branch-and-price algorithm to solve it. Recall that refueling a demand k requires to find a trip $P = (p^1 \dots, p^q)$ such that a refueling station is located at the end of each path-segment $p \in P \setminus \{p^q\}$ where $d(p^q) = \mathcal{D}(k)$. Since there is no refueling at the interval nodes of a path-segment, it is always best to choose the shortest path among all the path segments between two nodes for the RSLP-R problem. Thus, we only need to consider the

shortest path between two nodes as a path-segment. This core property is also considered by MirHassani and Ebrazi [99] to represent refueling constraints for a vehicle traveling on a fixed path. Our methodology generalizes this approach to the whole network to relax the fixed simple path assumption.

We define the following decision variables.

$$\begin{aligned} y^k &= \begin{cases} 1, & \text{if a feasible trip is built for the demand } k \in K \\ 0, & \text{otherwise,} \end{cases} \\ r_i &= \begin{cases} 1, & \text{if there is a refueling station located at node } i \in V \\ 0, & \text{otherwise,} \end{cases} \\ x_p^k &= \begin{cases} 1, & \text{if demand } k \in K \text{ uses path-segment } p \\ 0, & \text{otherwise,} \end{cases} \end{aligned}$$

We call $y^k, k \in K$ as the *cover* variables, $r_i, i \in V$ as the *refueling* variables and $x_p^k, k \in K, p \in \mathcal{P}$ as the *arc flow* variables. With these decision variables, *PSF* can be stated as follows:

$$\max \sum_{k \in K} f^k y^k \quad (6.1)$$

$$\text{s.t. } x^k(\delta^+(i)) - x^k(\delta^-(i)) = \begin{cases} y^k, & \text{if } i = \mathcal{O}(k) \\ -y^k, & \text{if } i = \mathcal{D}(k) \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in \bar{V}, k \in K, \quad (6.2)$$

$$\sum_{p \in \mathcal{P}^k} l(p) x_p^k \leq \lambda^k \quad \forall k \in K, \quad (6.3)$$

$$x^k(\delta^-(i)) \leq r_i \quad \forall k \in K, i \in V \quad (6.4)$$

$$r(V) \leq \rho \quad (6.5)$$

$$y^k \in \{0, 1\} \quad \forall k \in K, \quad (6.6)$$

$$r_i \in \{0, 1\} \quad \forall i \in V, \quad (6.7)$$

$$x_p^k \in \{0, 1\} \quad \forall k \in K, p \in \mathcal{P} \quad (6.8)$$

The objective function (6.1) is the total amount of the AFV flow volume to be captured. Constraints (6.2) are the flow balance equations that force a chosen demand to be carried from its source to its destination (covered) by the concatenation of feasible path-segments. Constraints (6.3) are the driver distance

tolerance constraints which ensure that the total length of any AFV trip is not longer than the maximum allowed. Constraints (6.4) enforce fuel range requirements by ensuring refueling at the end of each feasible path-segment that does not end in the destination node of the associated demand. Constraint (6.5) makes sure that not more than ρ refueling stations are located. Constraints (6.6-6.8) are the domain restrictions.

In order to strengthen the given formulation we can replace constraints (6.3) with the following constraints:

$$\sum_{p \in \mathcal{P}^k} l(p) x_p^k \leq \lambda^k y^k \quad \forall k \in K \quad (6.9)$$

This cut is very useful when solving the PSF formulation. Indeed, as we will more formally present below, integrality of the refueling variables is sufficient to guarantee the integrality of the cover and flow variables with the inclusion of this cut in the model. We will call this stronger formulation as $\overline{PSF}$. We now present our branch-and-price algorithm to solve $\overline{PSF}$. During the solution procedure column generation technique is employed to solve the linear relaxation of $\overline{PSF}$, say $\overline{PSF}$ -LP and obtain an upper bound for each node of the branch-and-bound tree.

6.1.3 LP Solution

6.1.3.1 Pricing Problem:

Let $\overline{RPSF}$ be the restricted $\overline{PSF}$ formulation with a subset of flow variables x_p^k . At every iteration we determine whether there exists a column with positive reduced cost such that including it to the $\overline{RPSF}$ might improve the objective function. If such columns are detected, we add them to the $\overline{RPSF}$ and repeat the procedure until there is no column left with a positive reduced cost.

Let π_i^k represent the unrestricted dual variables associated with Constraints (6.2), and κ^k and γ_i^k be the nonnegative dual variables associated with constraints

(6.3) and (6.4), respectively. For a flow variable x_p^k , the reduced cost $\bar{x}_p^k$ is given as:

$$\bar{x}_p^q = \begin{cases} \pi_{d(p)}^k - \pi_{o(p)}^k - l(p)\kappa^k, & \text{if } d(p) = \mathcal{D}(k) \\ \pi_{d(p)}^k - \pi_{o(p)}^k - l(p)\kappa^k - \gamma_{d(p)}^k, & \text{o.w.} \end{cases} \quad (6.10)$$

Considering the specific structure of the RLSP, in this chapter we consider the following definition for the *plausible-pairs*.

Definition 5. An ordered node pair $(i, j) \in (\bar{V} \times V) \cup (V \times \bar{V})$ is called a *plausible-pair* for a demand k if it satisfies the following conditions:

- It is possible to transit from node i to node j without any refueling. More formally:

$$\delta_{i,j}^* \leq \begin{cases} \Delta m^k, & \text{if } i \neq \mathcal{O}(k) \text{ and } j \neq \mathcal{D}(k) \\ \Delta m^k/2, & \text{o.w.} \end{cases} \quad (6.11)$$

- It is possible to visit nodes i and j without violating driver tolerance constraints. i.e.,

$$\delta_{\mathcal{O}(k),i}^* + \delta_{i,j}^* + \delta_{j,\mathcal{D}(k)}^* \leq \lambda^k \quad (6.12)$$

The set of all the plausible-pairs for a demand k is denoted by Π^k .

In order to identify flow variables that price out, it is only required to check *plausible-pairs* for each demand $k \in K$ and see if there is a pair $(i, j) \in \Pi^k$ such that, the shortest path $p_{i,j}^*$ from node i to j satisfies the following condition:

$$l(p_{i,j}^*)\kappa^k < \begin{cases} \pi_j^k - \pi_i^k, & \text{if } j = \mathcal{D}(k) \\ \pi_j^k - \pi_i^k - \gamma_j^k, & \text{o.w.} \end{cases} \quad (6.13)$$

Note that if the shortest path between a plausible pair (i, j) does not satisfy the above condition, none of the other paths connecting node i to node j can. Thus, for a plausible pair $(i, j) \in \Pi^k$, it is sufficient to check whether (6.13) is satisfied for the path segment $p_{i,j}^*$ and declare the variable $x_{p_{i,j}^*}^k$ as a positive reduced cost variable if this is the case.

6.1.3.2 Determining an Initial Set of Columns:

We define the initial variable pool as $X_0 = \{x_{\{p_{i,j}^*\}}^k | k \in K, (i, j) \in \Pi^k, (i, j) \in \bar{A}\}$. Note that, a solution for the $\overline{RPSF} - LP$, considering only the flow variables in X_0 contains enough information to derive all the needed dual variable values to properly construct the pricing problem as discussed in Chapter 4.

6.1.4 IP solution

In $\overline{PSF}$, all the decision variables are defined as binary. However, due to (6.9), requiring only the refueling variables as binary is sufficient to obtain a solution in which both cover and arc flow variables are also binary. Before proceeding with the formal propositions and their proofs, we need the following definition:

Definition 6. For a given solution (y, r, x) of $\overline{PSF} - LP$, we call $G_x^k = (\bar{V}, A_x^k)$ as the reduced graph of demand $k \in K$, where $A_x^k := \{a \in \bar{A} | a \in p, x_p^k > 0\}$.

Proposition 4. Let $(\hat{y}, \hat{r}, \hat{x})$ be an optimal solution for the $\overline{PSF} - LP$ where refueling variables $\hat{r}$ are all binary. Then, the cover variables $\hat{y}$ necessarily assume integral values.

Proof. Let $(\hat{y}, \hat{r}, \hat{x})$ be an optimal solution of $\overline{PSF} - LP$, where $\hat{r}_i \in \{0, 1\}, \forall i \in V$ and $\hat{z}$ is the optimal solution value. Assume there exists $\hat{k} \in K$ such that $0 < \hat{y}^{\hat{k}} < 1$. Let $U^{\hat{k}}$ be the set of trips that connect $\mathcal{O}(\hat{k})$ to $\mathcal{D}(\hat{k})$ in $G_{\hat{x}}^{\hat{k}}$. Note that $U^{\hat{k}}$ is not empty since $\hat{y}^{\hat{k}} > 0$ and the solution $(\hat{y}, \hat{r}, \hat{x})$ is feasible. Let u^* be the shortest path in $U^{\hat{k}}$. Now consider the solution $\bar{y}, \hat{r}, \bar{x}$ where

$$\bar{y}^k = \begin{cases} 1, & \text{if } k = \hat{k} \\ \hat{y}^k, & \text{o.w.} \end{cases} \quad \bar{x}_p^k = \begin{cases} 1, & \text{if } k = \hat{k} \text{ and } p \in u^* \\ 0, & \text{if } k = \hat{k} \text{ and } p \notin u^* \\ \hat{x}_p^k, & \text{o.w.} \end{cases} \quad (6.14)$$

Observe that this new solution $(\bar{y}, \hat{r}, \bar{x})$ is feasible since refueling variables are all integral and $u^* \leq \lambda^k$. Let $\bar{z}$ be the objective function value for the solution $(\bar{y}, \hat{r}, \bar{x})$. Then, $\bar{z} - \hat{z} = f^k(1 - \hat{y}^k) > 0$. This concludes the proof. $\square$

Proposition 5. *Let $\hat{z}$ be the optimal solution value for $\overline{PSF}$ -LP obtained by the solution $(\hat{y}, \hat{r}, \hat{x})$, where refueling variables $\hat{r}$ and cover variables $\hat{y}$ are all binary. Then the optimal solution value for $\overline{PSF}$ is equal to $\hat{z}$.*

Proof. Let $(\hat{y}, \hat{r}, \hat{x})$ be the optimal solution for $\overline{PSF}$ -LP where refueling variables $\hat{r}$ and cover variables $\hat{y}$ are all binary and assume that there exists a demand $\hat{k} \in K$ with a positive cover variable $\hat{x}_p^{\hat{k}} < 1$ (if there is no such flow variable, then the assertion is vacuously true). $U^{\hat{k}}$ and u^* definitions are the same as their definitions in the previous proof. Now consider the solution $(\hat{y}, \hat{r}, \bar{x})$ where:

$$\bar{x}_p^k = \begin{cases} 1, & \text{if } k = \hat{k} \text{ and } p \in u^* \\ 0, & \text{if } k = \hat{k} \text{ and } p \notin u^* \\ \hat{x}_p^k, & \text{o.w.} \end{cases} \quad (6.15)$$

Observe that the integrality of refueling and cover variables and u^* being the shortest path in $U^{\hat{k}}$ ensure that the new solution $(\hat{y}, \hat{r}, \bar{x})$ is feasible with the same objective function value and strictly less number of fractional flow variables than the starting solution $(\hat{y}, \hat{r}, \hat{x})$. Since one can repeat this procedure as much as needed to obtain an integral solution, the proof is complete. $\square$

Due to the Propositions 2 and 3, we only need to consider the refueling variables in the branching phase.

6.1.4.0.1 Branching on refueling variables Comparing the refueling variables $r_i, i \in V$ by the degrees of the associated node $i \in V$, we sort them in a descending order and obtain a priority list. Encountering a fractional solution, we select the fractional refueling variable highest in the list as the branching variable. Let r_i be the fractional refueling variable we chose to branch on:

- **Branching-cut-1** $r_i = 0$: In this case the set of arc flow variables $\underline{X}_i = \{x_p^k | k \in K, p \in \mathcal{P} \text{ and } d(p) = i\}$ are implicitly set to 0. Thus, we must make sure that in the pricing problem any path-segment $x_p^k \in \underline{X}_i$ should not appear as a positive reduced cost column. This can be easily done by setting $\gamma_i^k = \infty \quad \forall k \in K, \mathcal{D}(k) \neq i$.

- **Branching-cut-2** $r_i = 1$: In this case the flow variables are not affected by the branching cut and the pricing problem stays the same except for the possible change in the value of the dual variables γ_i^k .

6.2 Numerical Experiments

Comprehensive numerical experiments are conducted to test the performance of the branch-and-price algorithm (B&P). Two particular network topologies are considered: Arizona (AZ) network in Figure 6.2a and California (CA) network in Figure 6.2b. Being a commonly used network in the literature [25, 53, 71, 80, 91, 99, 143], AZ network constitutes a suitable test bed for us to compare the performance of B&P with the benchmark studies in the literature. The CA network [7] on the other hand is a close representation of the actual California State road network and allows us to test B&P in realistic large problem instances. The main parameters of these networks are presented in Table 6.1.

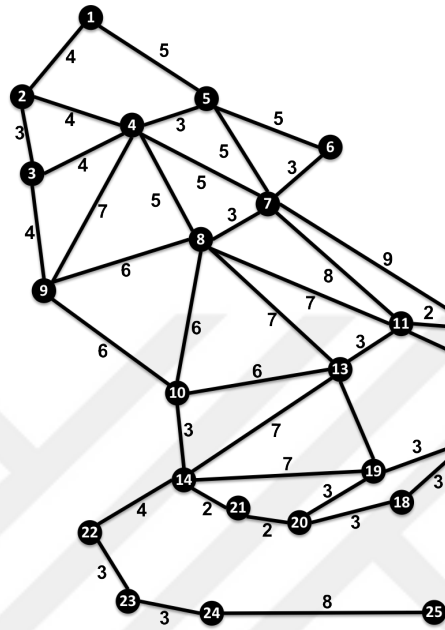
Table 6.1: Network and related O-D pair parameters

Network	#nodes	#edges	Node Degree			Count	O-D pairs		
			min	max	mean		min.dist	max.dist	mean.dist
AZ	25	42	1	6	3.36	300	2	38	14.23
CA	339	617	1	7	3.64	1167	30.06	463.50	153.37

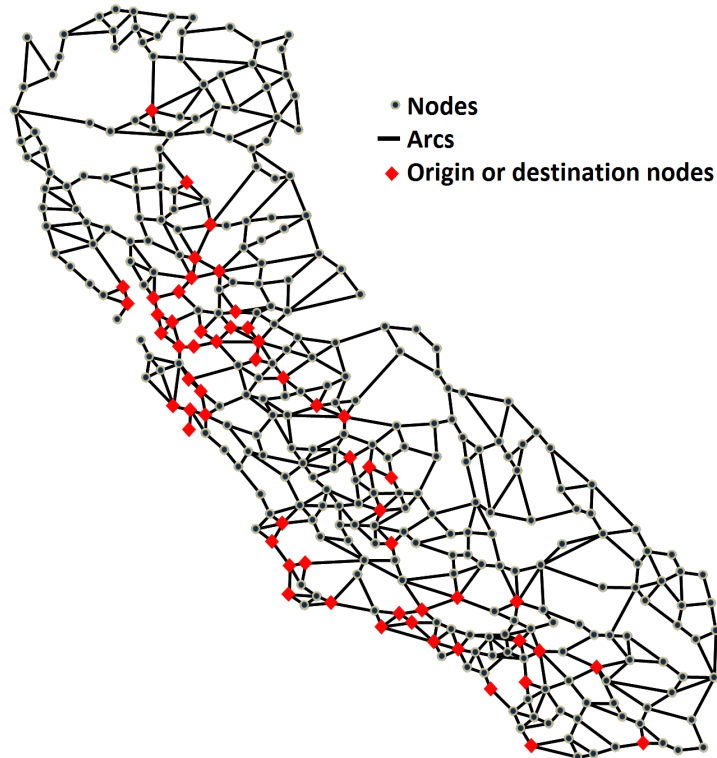
We implemented all the algorithms using Java under Linux and CPLEX 12.5 and all experiments are done on the same machine: AMD Opteron(tm) Processor 6282 SE with 2GB RAM.

6.2.1 AZ Network

For AZ network experiments, we generated the same test problems studied by Kim and Kuby [71]. All 25 nodes of the network are considered as O-D nodes and all the possible pairings between them are considered as O-D pairs. The flow is calculated by the gravity model proposed by Hodgson [53]. A total of 225



(a) Arizona state network



(b) California state network

Figure 6.2: Network instances

problem instances are obtained by considering

- 3 vehicle ranges: 4, 8 and 12
- 3 levels of driver tolerance: 0, 10 and 50%
- 25 different refueling station numbers: $1, \dots, 25$

Table 6.2 depicts the average CPU times of 25 runs ($\rho = 1, \dots, 25$) for different range and tolerance settings. The preprocessing time for DFRLM is the time it takes to generate the paths (by solving consecutive k -shortest path algorithms) and the minimal combination sets for these paths. The preprocessing time of B&P is for generating the plausible pairs for each demand. The right-most column, the solution time, shows the model solution time combined with the preprocessing time.

Table 6.2: Solution time comparisons of DFRLM and B&P algorithms.

Range	Tol. (%)	Preprocessing Time		Solution Time (total)	
		DFRLM	B&P	DFRLM	B&P
4	0	3.85	0.17	4.14	1.51
	10	5.03	0.17	5.4	1.54
	50	54.52	0.20	54.98	1.88
8	0	3.89	0.19	4.3	2.25
	10	4.91	0.20	5.37	2.22
	50	57.68	0.23	72.22	3.72
12	0	3.97	0.21	4.46	2.51
	10	5.12	0.22	5.77	2.39
	50	82.38	0.23	130.2	4.70

Even though branch-and-price algorithms are not as famous for their speed as their capability to handle very large problem instances, it is interesting to observe that the run times of B&P are much faster than DFRLM. Apparently, problems with longer vehicle range and higher driver tolerance take longer solution times. In those cases, the number of alternative feasible paths between O-D pairs increases which makes these problems harder to tackle. Notice that the

Table 6.3: B&P solutions for the AZ network

Range	ρ	No Tolerance					10 % Tolerance					50 % Tolerance				
		Opt.Sol	Lp.sol	#BBN	#Col.	Time	Opt.Sol	Lp.sol	#BBN	#Col.	Time	Opt.Sol	Lp.sol	#BBN	#Col.	Time
4	1	4.92	5.96	5	368	1.31	4.92	5.96	5	367	1.44	4.92	5.96	5	367	1.63
	2	6.31	11.91	35	379	3.08	6.31	11.91	35	377	2.98	6.31	11.91	35	377	3.84
	3	12.49	17.87	27	377	2.87	12.49	17.87	27	376	2.61	12.49	17.87	27	376	3.22
	4	20.38	23.82	19	372	2.16	20.38	23.82	19	376	2.35	20.38	23.82	19	376	2.68
	5	27.54	29.78	9	371	1.59	27.54	29.78	9	369	1.59	27.54	29.78	9	369	1.84
	6	34.01	35.73	7	370	1.58	34.01	35.73	7	373	1.52	34.01	35.73	7	373	1.75
	7	41.41	41.69	3	375	1.28	41.41	41.69	3	379	1.31	41.41	41.69	3	379	1.47
	8	45.26	47.64	9	431	2.12	45.26	47.64	9	403	2.05	45.26	47.64	9	403	2.52
	9	53.6	53.6	1	407	1.18	53.6	53.6	1	403	1.22	53.6	53.6	1	403	1.37
	10	55.97	56.71	3	473	1.39	55.97	56.71	3	441	1.32	56.08	57.98	3	441	2.47
	11	59.82	59.82	1	458	1.31	59.82	59.82	1	453	1.28	62.36	62.36	1	453	1.4
	12	61.51	61.84	5	500	1.49	61.69	61.84	5	503	1.55	64.41	64.41	3	503	1.62
	13	62.72	63.86	9	476	2.04	62.72	63.86	9	523	1.92	65.26	66.43	11	523	2.4
	14	65.12	65.88	5	488	1.62	65.12	65.88	5	502	1.7	67.66	68.45	7	502	2.28
	15	67.89	67.89	1	462	1.27	67.89	67.89	1	497	1.14	70.44	70.47	1	497	1.65
	16	69.58	69.58	1	481	1.26	69.77	69.77	1	512	1.3	72.48	72.48	1	512	1.44
	17	71.12	71.12	1	508	1.15	71.3	71.3	1	474	1.27	74.02	74.02	1	474	1.32
	18	71.81	72.23	3	470	1.4	71.99	72.42	3	474	1.44	74.84	74.84	3	474	1.34
	19	73.34	73.34	1	523	1.18	73.53	73.53	1	500	1.23	75.47	75.56	1	500	2.01
	20	73.98	73.98	1	575	1.08	74.22	74.22	1	581	1.15	76.28	76.28	1	581	1.38
	21	73.98	74.21	3	596	1.29	74.22	74.45	3	515	1.44	76.28	76.52	3	515	2.21
	22	74.45	74.45	1	598	1.12	74.68	74.68	1	517	1.42	76.75	76.75	1	517	1.41
	23	74.54	74.54	1	586	1.12	74.78	74.78	1	586	1.16	76.84	76.84	1	586	1.37
	24	74.54	74.54	1	370	1.05	74.78	74.78	1	357	1	76.84	76.84	1	357	1.23
	25	74.54	74.54	1	351	0.91	74.78	74.78	1	357	1.03	76.84	76.84	1	357	1.23
8	1	17.13	17.13	1	776	1.2	17.13	17.13	1	823	1.22	17.13	17.13	1	823	1.53
	2	32.58	32.58	1	778	1.18	32.58	32.58	1	873	1.3	32.58	32.58	1	873	2.08
	3	44.41	44.41	1	845	1.36	44.41	44.41	1	895	1.36	44.41	44.41	1	895	1.74
	4	55.97	55.97	1	892	1.38	55.97	55.97	1	958	1.43	56.08	56.08	1	958	2.06
	5	63.52	63.52	1	906	1.4	63.52	63.52	1	926	1.49	64.06	64.06	1	926	3.64
	6	68.08	68.74	5	1073	2.21	68.08	68.88	5	1093	2.33	71.61	71.61	5	1093	3.49
	7	72.32	73.95	9	1094	2.62	72.32	74.24	9	1137	2.69	75.32	79.08	7	1137	7.81
	8	75.39	79.16	17	1193	4.25	77.87	79.54	17	1144	2.63	84.56	85.64	5	1144	5.22
	9	82.35	84.25	5	1120	2.74	82.77	84.8	5	1245	3.18	92.18	92.18	9	1245	4.12
	10	87.58	89.33	9	1216	3.38	90.06	90.06	9	1189	1.88	95.99	95.99	1	1189	2.84
	11	94.41	94.41	1	1226	2.01	94.41	94.41	1	1200	1.95	98.25	98.25	1	1200	3.01
	12	96.8	96.8	1	1291	2.02	96.8	96.8	1	1318	2.04	98.76	98.76	1	1318	4.33
	13	97.78	98.07	7	1406	3.51	97.78	98.1	7	1421	3.41	99.03	99.11	7	1421	5.37
	14	98.36	98.57	9	1494	3.04	98.43	98.75	9	1494	3.41	99.45	99.45	9	1494	3.78
	15	98.48	98.97	5	1492	3.12	98.74	99.39	5	1512	3.47	99.72	99.76	7	1512	5.74
	16	99.17	99.21	3	1465	2.84	99.71	99.75	3	1496	3.05	99.81	99.85	3	1496	6.12
	17	99.24	99.29	9	1495	3.58	99.77	99.82	9	1513	3.89	99.87	99.93	15	1513	8.41
	18	99.33	99.36	3	1577	2.69	99.86	99.89	3	1463	3.17	99.97	99.98	3	1463	6.77
	19	99.39	99.39	2	1533	2.65	99.92	99.92	2	1461	2.7	100	100	2	1461	3.25
	20	99.39	99.39	2	1568	1.83	99.92	99.92	2	1424	1.81	100	100	1	1424	2.66
	21	99.39	99.39	2	1567	1.75	99.92	99.92	2	1539	1.73	100	100	1	1539	1.91
	22	99.39	99.39	1	1689	1.53	99.92	99.92	1	1250	1.4	100	100	1	1250	2.03
	23	99.39	99.39	1	1565	1.44	99.92	99.92	1	1604	1.54	100	100	1	1604	1.92
	24	99.39	99.39	1	1648	1.49	99.92	99.92	1	1339	1.35	100	100	1	1339	1.93
	25	99.39	99.39	1	481	1.01	99.92	99.92	1	619	1.08	100	100	1	619	1.24
12	1	18.23	18.23	1	1342	1.17	18.23	18.23	1	1475	1.27	18.23	18.23	1	1475	1.64
	2	34.34	34.75	3	1333	1.5	34.34	34.75	3	1562	1.91	34.34	34.75	3	1562	2.48
	3	47.9	47.9	1	1490	1.44	47.9	47.9	1	1618	1.48	49.04	49.04	1	1618	3.03
	4	57.47	57.47	1	1572	1.56	58.14	58.14	1	1833	2.16	62.64	62.64	1	1833	3.09
	5	66.18	66.18	1	1619	1.72	67.7	67.7	1	1818	1.91	72.46	72.77	1	1818	5.59
	6	72.53	74.11	9	1918	4.26	75	76	9	1992	2.83	82.15	82.5	3	1992	5.33
	7	80.88	81.57	9	1921	4.36	83.35	83.35	9	1941	2.27	91.78	91.78	1	1941	4.75
	8	87.33	87.4	7	1945	4.57	88.83	88.83	7	2009	2.48	95.95	95.95	1	2009	5.06
	9	92.71	92.71	1	1834	2.86	92.93	92.98	1	2027	2.85	97.59	97.75	3	2027	6.71
	10	96.83	96.83	1	1889	2.84	96.83	96.83	1	2037	2.83	98.97	99.18	1	2037	7.39
	11	97.81	98.03	5	1924	3.51	97.81	98.03	5	1994	3.45	99.54	99.55	5	1994	6.7
	12	98.66	99.16	13	1928	4.5	98.66	99.16	13	2049	4.71	99.8	99.82	13	2049	6.85
	13	99.3	99.57	5	2048	4.22	99.3	99.72	5	2187	4.7	99.89	99.94	13	2187	12.69
	14	99.85	99.85	1	1857	2.97	99.85	99.85	1	2096	3.61	99.95	100	3	2096	14.02
	15	99.93	99.93	1	1965	3.24	99.93	99.93	1	2163	3.51	100	100	1	2163	4.94
	16	100	100	1	2012	2.96	100	100	1	2062	3.07	100	100	1	2062	5.3
	17	100	100	1	1951	2.58	100	100	1	2052	2.42	100	100	2	2052	4.22
	18	100	100	2	1827	2.04	100	100	2	1901	2.05	100	100	1	1901	4.29
	19	100	100	1	1901	2.02	100	100	1	1919	1.88	100	100	1	1919	2.38
	20	100	100	4	2006	2.04	100	100	4	1863	1.66	100	100	1	1863	2.41
	21	100	100	1	1876	1.54	100	100	1	1863	1.63	100	100	1	1863	2.23
	22	100	100	1	1878	1.41	100	100	1	1725	1.37	100	100	1	1725	1.98
	23	100	100	1	1866	1.35	100	100	1	1773	1.42	100	100	1	1773	1.63
	24	100	100	1	1819	1.18	100	100	1	1669	1.33	100	100	1	1669	1.5
	25	100	100	1	640	0.99	100	100	1	984	1.05	100	100	1	984	1.18

computational performance of DFRLM quickly degrade as problem gets harder whereas the solution times for B&P are more stable.

Both algorithms DFRLM and B&P are run on all problem instances. Table 6.3 shows the solutions obtained by the B&P algorithm. In the table, ρ stands for the number of refueling stations, Opt.Sol shows the maximum percent of the flow that could be refueled in the optimal solution, LP.sol indicates the solution value for the linear relaxation of the problem, BBN is the number of branch-and-bound nodes explored by the B&P algorithm and #Col. indicates the total number of columns generated. Table 6.3 shows that optimal values are quite close to those of the linear relaxation solutions. This indicates the strength of the path-segment formulation which helps to make B&P a competitive alternative to the state-of-the-art models in the literature. Our results also show that the computational performance of the B&P algorithm does not vary significantly across different problem instances.

All solution values for the DFRLM are the same with those resulting from B&P except for the three cases highlighted in Table 6.3. For those instances, B&P is able to generate a better solution by utilizing non-simple paths. One example is when range is 12, the tolerance is 50% and $\rho = 6$. The refueling stations are located at nodes $\{4, 10, 12, 17, 20, 22\}$ in the optimal solution. Even though there does not exist a feasible simple path between nodes 10 and 11, a non-simple path can connect these two nodes and cover the flow in between. When traveling from node 11 to node 10 on a non-simple path, the vehicle first visits node 12, refuels there, and travels to node 10 by visiting node 11 again. The travel distance in total is 13 which is just less than the tolerable maximum 13.5. This is an example of a non-simple path occurrence. It is no surprise that all these three highlighted cases share the same high range and tolerance (range=12, tolerance= 50%) parameters. This is because, for a non-simple path to be feasible, the range of the vehicle should be long enough to traverse two consecutive arcs without refueling and driver tolerance should be high enough to compensate for the extra mileage of such a detour. Emergence of non-simple paths even in a quite aggregate network such as AZ is an interesting result which indicates that neglecting them could result in sub-optimal solutions especially in

less aggregate and more realistic network instances.

6.2.2 CA Network

In order to study more realistic problem instances, CA network test problems are used. For this set of experiments, all the urban population centers in the California are considered as O-D nodes. There are total of 57 such centers with population more than 50,000 according to recent reports [135]. All possible pairings of these population centers that are not closer than 30 km are considered as O-D pairs (1167 in total) and the volume of the flow on each pair is calculated using the gravity model [53]. Our experimental design contains 64 problem instances:

- 2 vehicle ranges: 100 and 150 km,
- 4 levels of driver tolerance: 0, 5, 10 and 20%,
- 8 different refueling station limits: 1, 5, 10, $\dots$, 35.

We consider driver tolerances up to 20% in this realistic case study since higher tolerance values are hard to justify with economic or environmental concerns of the drivers. For these problem instances, DFRLM fail to solve the problem for even minor driver tolerances. This model cannot even keep the problem in the memory in these problem instances. This is due to the exponential growth in the number of paths as the driver tolerance level increases. Illustrating this fact, Figure 6.3 shows the total number of alternative paths that connect O-D pairs for a given tolerance level. There are more than 3.5 million alternative paths for 5% driver tolerance and this number grows almost thirty times larger when the tolerance is increased to %8. However, the B&P algorithm does not get overwhelmed by these large problem instances since it does not require the inclusion of all those paths, only a tiny fraction of which actually appear in the optimal solution.

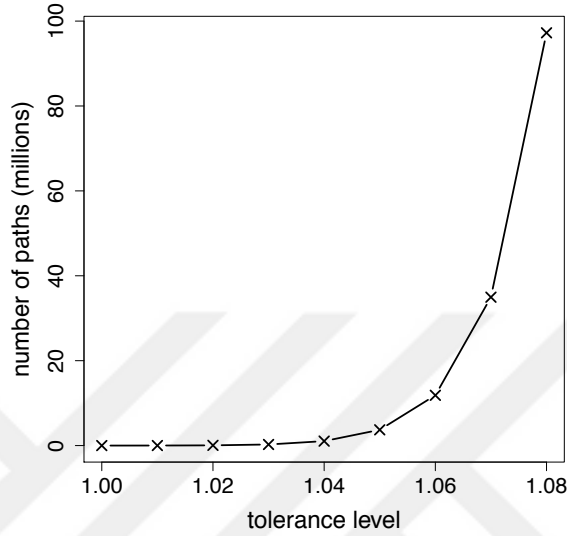


Figure 6.3: Number of paths generated vs tolerance Level

Table 6.4 shows the results of the computational experiments with B&P on the CA network. In the table, ρ stands for the number of refueling stations, Sol. is the maximum percent of the flow that could be refueled by the B&P algorithm solution, %Gap is the percent of the optimality gap, BBN is the number of branch-and-bound nodes explored by the B&P algorithm and #Col. indicates the total number of columns generated and RT is the solution time in seconds. The sign ‘-’ indicates the problem is not solved when 100% coverage is already established for less number of refueling stations.

As seen in the table, B&P is able to solve approximately the 75% of the problems to optimality. For those instances, where B&P did not converge to the optimal solution in the given time limit of 3 hours, the maximum optimality gap is 0.506% and the average is below 0.007%. The results show that problems with small or large number of refueling stations are easier to solve and harder problems arise in between. Also problems with high tolerance values are naturally hard to solve since a higher number of columns are generated and considered in the solutions. The same claims and arguments are obviously true for the higher driving ranges.

Also note that higher driver tolerances make more significant differences for the medium values of p . For example consider the problem instances with $\rho = 5$, $range = 100$ km in Table 6.4. For this set of problems just a 5% driver tolerance results in 9 % increase in the percentage total refuelable flow. However, the gain for the same driver tolerance decreases when p grows or decreases. This is expected since, for small ρ , driver tolerance can not compensate the scarcity of the refueling stations to generate a feasible path whereas in the case where the refueling stations are widespread, there is not so much need for a long refueling detour.

6.3 Summary

In this final part of the dissertation, we present the refueling station location problem with routing (RSLP-R) for locating a given number of refueling stations for alternative fuel vehicles in a road network so as to maximize the total flow covered. Driver deviations from the shortest path up to a certain tolerance value are considered as alternative paths including non-simple ones. The problem is practically important due to fact that the adoption of AFVs strongly depends on the availability of the refueling infrastructure and the high cost of this initial investment motivates the efforts for the best use of limited resources. It is theoretically challenging because the problem is NP-Complete and previous formulations of similar problems failed to handle large networks due to their modeling structures. Regarding to refueling station location problem, the most important contribution of our dissertation study to the literature is extending the size of the solvable problems. Rather than pregenerating all the path alternatives before solving the model, we apply a branch-and-price solution algorithm which enables us to handle problems that were not of manageable size by previous works on similar contexts. Our algorithm also decreased the solution times with respect to previous studies which is another major contribution to the literature. The efficiency of the solution technique is mainly due to the path-segment definition in our formulation.

Table 6.4: B&P Solutions for the CA network

ρ	λ	Range=100 km					Range=150 km				
		Sol.	%Gap	BBN	#Col.	RT	Sol.	%Gap	BBN	#Col.	RT
1	1	30.545	0	1	71403	155	33.953	0	1	95141	200
	1.05	32.979	0	1	228320	146	34.439	0	1	312759	199
	1.1	33.285	0	1	372845	183	34.618	0	1	556776	221
	1.2	36.457	0	1	664187	202	36.828	0	1	1097566	306
5	1	67.084	0	1	70532	191	79.944	0	1	93654	217
	1.05	76.002	0	19	265154	811	84.136	0	3	337780	423
	1.1	79.573	0	3	409219	740	85.907	0	11	625782	1142
	1.2	82.861	0	29	877962	6556	89.078	0	109	1334565	10236
10	1	87.98	0	3	74059	239	92.984	0	15	104253	793
	1.05	91.977	0	7	246228	758	95.859	0	7	343821	878
	1.1	93.469	0	165	476869	6585	97.403	0	1	605160	989
	1.2	94.609	0.506	19	793718	10800	98.286	0	7	1270512	7262
15	1	95.008	0	1	86957	462	98.348	0	83	110644	2105
	1.05	97.793	0	27	276550	1655	99.435	0	37	354104	2204
	1.1	98.885	0	1	449125	910	99.793	0	31	633476	2497
	1.2	99.208	0.124	9	880796	10800	99.917	0.044	24	1267006	10800
20	1	98.407	0	69	91136	1980	99.89	0	619	114941	9165
	1.05	99.525	0	283	293126	10691	99.982	0.007	435	364327	10800
	1.1	99.82	0	229	499775	9885	99.969	0.028	193	666414	10800
	1.2	99.974	0.01	2	823954	10800	100	0	7	1128846	8820
25	1	99.776	0.134	331	91596	10800	100	0	365	108527	6240
	1.05	99.936	0.05	475	287709	10800	100	0	10	315231	1132
	1.1	99.996	0.004	48	452787	10800	100	0	131	595547	3620
	1.2	99.982	0.018	12	783728	10800	100	-	-	-	-
30	1	99.964	0.036	1041	91583	10800	100	-	-	-	-
	1.05	99.991	0.009	579	273906	10800	100	-	-	-	-
	1.1	99.988	0.011	211	470354	10800	100	-	-	-	-
	1.2	99.997	0.003	19	762189	10800	100	-	-	-	-
35	1	100	0	23	77066	581	100	-	-	-	-
	1.05	100	0	30	237368	1192	100	-	-	-	-
	1.1	100	0	34	413932	2146	100	-	-	-	-
	1.2	100	0	79	728364	6539	100	-	-	-	-

Chapter 7

Conclusion

7.1 Remarks

Extending the reach of a commodity/entity or providing capability to switch mode of transmission/transportation relays constitute crucial parts of several transportation and telecommunication networks. Despite their practical significance, because of the diversity of the settings in which they arise, relay location problems have not received a unifying treatment in the literature and this dissertation is meant to be a first attempt to close this gap.

In this dissertation we study four relay location problems with different practical settings. Although each of these problems has different aspects that distinguish it from the others we present a common framework to approach all these problems with the help of hub location perspective and path-segment based formulations that can easily capture the reach limitations of the commodities and transportation mode switching capabilities of the relays. As a result we are able to devise exact solution methods which can compete with the state of the art heuristic approaches available in the literature.

The contribution of our dissertation is not confined to extending the computational efficiency. Approaching relay location problems from different angles: network topologies, network survivability, capacities, costs, and demands we provide significant theoretical results and managerial insights. Our studies shed light onto the questions such as “what makes relay location problems challenging” and “how does problem data affect the problem difficulty in various aspects”. We present significant results about the structure of some optimal solutions and suggest new formulation approaches that exploit these special structures. Our extensive computational tests with realistic problem instances do not just provide good test beds for the computational efficiency of our solution methods but also render quite significant practical insights for the network managers, investors and governments.

7.2 Future Study

Relating hub location and relay location problems, proposing new application areas for the relay location problems, proposing novel formulation approaches with path-segments this study presents various new research directions to study several classic problems with a new perspective or to develop efficient solution methods for the new emerging problems in transportation and telecommunications networks. In particular, path-segment based formulations, with their flexible nature and capability to easily incorporate side constraints on the routes to connect OD pairs, have significant potential to produce efficient solution methods for challenging location and routing problems that arise in quite diverse practical settings.

One such example is the dynamic vehicle routing problem which began to attract significant research effort due to the new technological advances in mobile communication and global positioning systems [113]. Path segment based formulations can be easily employed to approach these problems in which a corrective action needs to be taken or some state change occurs at the end of each path segment. As stated by Pillac et al. [113], besides dynamic routing problems, where the solution is a set of routes with sequence of customer visits/vehicle

stops, there are also well known vehicle dispatching problems, such as emergency response fleet location and relocation problems [16, 45] or the dynamic allocation problems in the area of long haul truckload trucking [101, 115, 127], for which hub location perspective we present along with the path segment based formulations can be very useful to develop efficient solution approaches.

In multi modal freight transportation, stations/facilities where mode of transportation is switched can be modeled as relay points. As such, this growing area of research constitutes another significant application area for the path segment based formulations where each path segment can be associated with a transportation mode as we present in Chapter 4. Such an approach can also be easily generalized for the closely related Intermodal transportation, Co-modal transportation and Synchromodal transportation problems that recently gained more popularity as a result of increasing volumes of containerized transport (15% annual growth) that enhances safety of cargo, reduction of handling costs, standardization, and accessibility to multiple modes of transportation [128].

In a completely different setting, relay functionalities and hub location perspective can be quite useful to model prorogation of influence on social networks. Similar to optical signals, influences lose their strength by the *distance* they travel and might need relays/regenerators to extend their reach. Thus, the solution approaches presented in this study could be extended to address problems which aim to maximize the spread of influence on a social network [65].

Bibliography

- [1] G. Agrawal. *Fiber-optic communication systems*, volume 222. Wiley, 2012.
- [2] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, 1993.
- [3] T. Ali, S. Radhakrishnan, S. Pulat, and N. Gaddipati. Relay network design in freight transportation systems. *Transportation Research Part E: Logistics and Transportation Review*, 38(0):405 – 422, 2002.
- [4] S. Alumur and B. Kara. Network hub location problems: The state of the art. *European Journal of Operational Research*, 190(1):1–21, 2008.
- [5] S. A. Alumur, B. Y. Kara, and O. E. Karasan. The design of single allocation incomplete hub networks. *Transportation Research Part B: Methodological*, 43(10):936 – 951, 2009. ISSN 0191-2615.
- [6] Y. An, Y. Zhang, and B. Zeng. The reliable hub-and-spoke design problem: Models and algorithms. *optimization-online.org*, 2011. URL <http://www.optimization-online.org/DB-HTML/2011/05/3043.html>.
- [7] O. Arslan, B. Yıldız, and O. E. Karasan. Impacts of battery characteristics, driver preferences and road network features on travel costs of a plug-in hybrid electric vehicle (phev) for long-distance trips. *Energy Policy*, 2014.
- [8] N. Azizi, S. Chauhan, S. Salhi, and N. Vidyarthi. The impact of hub failure in hub-and-spoke networks: Mathematical formulations and solution techniques. *Computers & Operations Research*, [http : //dx.doi.org/10.1016/j.cor.2014.05.012](http://dx.doi.org/10.1016/j.cor.2014.05.012) (to appear in print), 2014.

- [9] R. Bapna, L. S. Thakur, and S. K. Nair. Infrastructure development for conversion to environmentally friendly fuel. *European Journal of Operational Research*, 142(3):480–496, 2002.
- [10] C. Barnhart, C. A. Hane, E. L. Johnson, and G. Sigismondi. A column generation and partitioning approach for multi-commodity flow problems. *Telecommunication Systems*, 3(3):239–258, 1994.
- [11] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329, 1998.
- [12] C. Barnhart, C. Hane, and P. Vance. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48(2):318–326, 2000.
- [13] J. E. Beasley. Or-library: distributing test problems by electronic mail. *Journal of the operational research society*, pages 1069–1072, 1990.
- [14] O. Berman, R. C. Larson, and N. Fouska. Optimal location of discretionary service facilities. *Transportation Science*, 26(3):201–211, 1992.
- [15] G. Bosco, V. Curri, A. Carena, P. Poggiolini, and F. Forghieri. On the performance of nyquist-wdm terabit superchannels based on pm-bpsk, pm-qpsk, pm-8qam or pm-16qam subcarriers. *Journal of Lightwave Technology*, 29(1):53–61, 2011.
- [16] L. Brotcorne, G. Laporte, and F. Semet. Ambulance location and relocation models. *European journal of operational research*, 147(3):451–463, 2003.
- [17] E. A. Cabral, E. Erkut, G. Laporte, and R. A. Patterson. The network design problem with relays. *European Journal of Operational Research*, 180(2):834–844, 2007.
- [18] H. Calik, S. Alumur, B. Kara, and O. Karasan. A tabu-based heuristic for the hub covering problem over incomplete hub networks. *Computers & Operations Research*, 36(12):3088–3096, 2009.

- [19] J. Campbell. Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, 72(2):387–405, 1994.
- [20] J. Campbell. A survey of network hub location. *Studies in Locational Analysis*, 6:31–49, 1994.
- [21] J. Campbell and M. O’Kelly. Twenty-five years of hub location research. *Transportation Science*, 46(2):153–169, 2012.
- [22] J. Campbell, A. Ernst, and M. Krishnamoorthy. Hub arc location problems: Part i- introduction and results. *Management Science*, 51(10):1540–1555, 2005.
- [23] J. Campbell, A. Ernst, and M. Krishnamoorthy. Hub arc location problems: Part ii- formulations and optimal algorithms. *Management Science*, 51(10):1556–1571, 2005.
- [24] J. F. Campbell. Continuous and discrete demand hub location problems. *Transportation Research Part B: Methodological*, 27(6):473 – 482, 1993. ISSN 0191-2615.
- [25] I. Capar, M. Kuby, V. J. Leon, and Y.-J. Tsai. An arc coverpath-cover formulation and strategic analysis of alternative-fuel station locations. *European Journal of Operational Research*, 227(1):142 – 151, 2013.
- [26] R. Cardillo, V. Curri, and M. Mellia. Considering transmission impairments in configuring wavelength routed optical networks. In *Optical Fiber Communication Conference*, page OFG6. Optical Society of America, 2006.
- [27] S. Chen, I. Ljubić, and S. Raghavan. The regenerator location problem. *Networks*, 55(3):205–220, 2009.
- [28] S. Chen, I. Ljubić, and S. Raghavan. The generalized regenerator location problem. *Journal On Computing*, 27(2), 2015.
- [29] M. Chimani, M. Kandyba, I. Ljubić, and P. Mutzel. Strong formulations for 2-node-connected steiner network problems. *Combinatorial Optimization and Applications*, pages 190–200, 2008.

- [30] A. Cohn and C. Barnhart. Improving crew scheduling by incorporating key maintenance routing decisions. *Operations Research*, 51(3):387–396, 2003.
- [31] I. Correia, S. Nickel, and F. S. da Gama. Single-assignment hub location problems with multiple capacity levels. *Transportation Research Part B: Methodological*, 44(89):1047 – 1066, 2010. ISSN 0191-2615.
- [32] A. Costa, J.-F. Cordeau, and G. Laporte. Models and branch-and-cut algorithms for the steiner tree problem with revenues, budget and hop constraints. *Networks*, 2009. doi: 10.1002/net.
- [33] S. Davari, M. Zarandi, and I. Turksen. The fuzzy reliable hub location problem. In *Fuzzy Information Processing Society (NAFIPS), 2010 Annual Meeting of the North American*, pages 1–6, 2010.
- [34] J. de Azevedo, J. Silvestre Madeira, E. Vieira Martins, and F. Pires. A computational improvement for a shortest paths ranking algorithm. *European Journal of Operational Research*, 73(1):188–191, 1994.
- [35] Z. Degraeve and R. Jans. A new dantzig-wolfe reformulation and branch-and-price algorithm for the capacitated lot-sizing problem with setup times. *Operations Research*, 55(5):909–920, 2007.
- [36] G. Desaulniers. Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Operations research*, 58(1):179–192, 2010.
- [37] A. Duarte, R. Martí, M. Resende, and R. Silva. Improved heuristics for the regenerator location problem. *International Transactions in Operational Research*, 21(4):541–558, 2014.
- [38] J. Edmonds. Submodular functions, matroids, and certain polyhedra. *Combinatorial structures and their applications*, pages 69–87, 1970.
- [39] R. Essiambre, G. Kramer, P. Winzer, G. Foschini, and B. Goebel. Capacity limits of optical fiber networks. *Lightwave Technology, Journal of*, 28(4): 662–701, 2010.

- [40] M. Flammini, A. Marchetti-Spaccamela, G. Monaco, L. Moscardelli, and S. Zaks. On the complexity of the regenerator placement problem in optical networks. *IEEE/ACM Transactions on Networking (TON)*, 19(2):498–511, 2011.
- [41] B. Fortz, M. Labbé, and F. Maffioli. Solving the two-connected network with bounded meshes problem. *Operations Research*, 48(6):866–877, 2000.
- [42] P. Foulhoux, O. Karasan, A. Mahjoub, and H. O. Özkök. Survivability in hierarchical telecommunications networks. *Networks*, 59(1):37–58, 2012.
- [43] R. Francis, J. L. McGinnis, and J. White. *Facility layout and location: An analytical approach*. Prentice Hall, 1992.
- [44] M. Garey and D. Johnson. *Computers and intractability: A guide to the theory of NP-completeness*. W.H. Freeman & Co., 1979.
- [45] M. Gendreau, G. Laporte, and F. Semet. A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. *Parallel computing*, 27(12):1641–1653, 2001.
- [46] B. Gendron, A. Lucena, A. Cunha, and L. Simonetti. Benders Decomposition, Branch-and-Cut and Hybrid Algorithms for the Minimum Connected Dominating Set Problem. Technical report, Interuniversity REsearch Centre on Enterprise Networks, Logistics and Transportation (CIRRELT, 10 2012.
- [47] M. F. Goodchild and V. T. Noronha. Location-allocation and impulsive shopping: the case of gasoline retailing. *Spatial analysis and location-allocation models*, pages 121–136, 1987.
- [48] M. Grotschel, C. Monma, and M. Stoer. Design of survivable networks. *Handbooks in Operations Research and Management Science*, 7:617–672, 1995.
- [49] H. Hamacher and T. Meyer. Hub cover and hub center problems. *Department of Mathematics, University of Kaiserslautern, Germany*, 2006.

- [50] M. Hamidi, M. Gholamian, and K. Shahanaghi. Developing prevention reliability in hub location models. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 2014. doi: 10.1177/1748006X13519247 (to appear in print).
- [51] F. He, D. Wu, Y. Yin, and Y. Guan. Optimal deployment of public charging stations for plug-in hybrid electric vehicles. *Transportation Research Part B: Methodological*, 47(0):87 – 101, 2013.
- [52] J. He, M. Brandt-Pearce, Y. Pointurier, C. L. Brown, and S. Subramaniam. Adaptive wavelength assignment using wavelength spectrum separation for distributed optical networks. In *Communications, 2007. ICC'07. IEEE International Conference on*, pages 2406–2411. IEEE, 2007.
- [53] M. J. Hodgson. A flow-capturing location-allocation model. *Geographical Analysis*, 22(3):270–279, 1990.
- [54] Y. Huang, J. P. Heritage, and B. Mukherjee. Connection provisioning with transmission impairment consideration in optical wdm networks with high-speed channels. *Lightwave Technology, Journal of*, 23(3):982–993, 2005.
- [55] R. Hulsermann, A. Betker, M. Jager, S. Bodamer, M. Barry, J. Spath, C. Gauger, and M. Kohn. A set of typical transport network scenarios for network modeling. *ITG FACHBERICHT*, (182):65–72, 2004.
- [56] F. K. Hwang, D. S. Richards, and P. Winter. *The Steiner tree problem*. Elsevier, 1992.
- [57] C. Index. Global mobile data traffic forecast update, 2011–2016, white paper, 2012.
- [58] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka. Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies. *Communications Magazine, IEEE*, 47(11):66–73, 2009.

- [59] M. Jinno, B. Kozicki, H. Takara, A. Watanabe, Y. Sone, T. Tanaka, and A. Hirano. Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network [topics in optical communications]. *Communications Magazine, IEEE*, 48(8):138–145, 2010.
- [60] M. Jinno, T. Takagi, and K. Kiyokawa. Minimal virtualized-elastic-regenerator placement and least congestion resources assignment for translucent elastic optical networks. In *Optical Fiber Communication Conference*, page Th3J.2. Optical Society of America, 2015. doi: 10.1364/OFC.2015.Th3J.2.
- [61] A. Kahya. *Routing Spectrum Allocation and Regenerator Placement in Flexible-Grid Optical Networks*. PhD thesis, Bilkent university, 2013.
- [62] J. E. Kang and W. Recker. Strategic hydrogen refueling station locations with scheduling and routing considerations of individual vehicles. *Transportation Science*, 2014. doi: 10.1287/trsc.2014.0519.
- [63] B. Kara and B. Tansel. On the single-assignment p- hub center problem: Models and linearizations. *European Journal of Operational Reserach*, 125: 648–655, 2000.
- [64] O. E. Karasan, A. R. Mahjoub, O. Özkök, and H. Yaman. Survivability in hierarchical telecommunications networks under dual homing. *INFORMS Journal on Computing*, 26(1):1–15, 2013.
- [65] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.
- [66] H. Kerivin and A. Mahjoub. Design of survivable networks: A survey. *Networks*, 46(1):1–21, 2005.
- [67] P. Kewcharoenwong and H. Uster. Benders decomposition algorithms for the fixed-charge relay network design in telecommunications. *Telecommunication Systems*, 56(4):441–453, 2014.

- [68] H. Kim. P-hub protection models for survivable hub network design. *Journal of Geographical Systems*, 14(4):437–461, 2012. ISSN 1435-5930.
- [69] H. Kim and M. O’Kelly. Reliable p-hub location problems in telecommunication networks. *Geographical Analysis*, 41(0):283 – 306, 2009.
- [70] H. Kim and M. E. O’Kelly. *Reliable p-hub location problems and protection models for hub network design*. PhD thesis, Ohio State University, 2008.
- [71] J.-G. Kim and M. Kubry. The deviation-flow refueling location model for optimizing a network of refueling stations. *International Journal of Hydrogen Energy*, 37(6):5406 – 5420, 2012.
- [72] J.-G. Kim and M. Kubry. A network transformation heuristic approach for the deviation flow refueling location model. *Computers & Operations Research*, 40(4):1122 – 1131, 2013.
- [73] S. Kim and S. Seo. Regenerator placement algorithms for connection establishment in all-optical networks. In *Communications, IEE Proceedings-*, volume 148, pages 25–30. IET, 2001.
- [74] R. Kitamura and D. Sperling. Refueling behavior of automobile drivers. *Transportation Research Part A: General*, 21(3):235 – 245, 1987.
- [75] M. Klinkowski. On the effect of regenerator placement on spectrum usage in translucent elastic optical networks. In *Transparent Optical Networks (ICTON), 2012 14th International Conference on*, pages 1–6. IEEE, 2012.
- [76] M. Klinkowski. On the effect of regenerator placement on spectrum usage in translucent elastic optical networks. In *Transparent Optical Networks (ICTON), 2012 14th International Conference on*, pages 1–6. IEEE, 2012.
- [77] A. Konak. Network design problem with relays: A genetic algorithm with a path-based crossover and a set covering formulation. *European Journal of Operational Research*, 218(0):829–837, 2012.
- [78] A. Konak. Network design problem with relays: A genetic algorithm with a path-based crossover and a set covering formulation. *European Journal of Operational Research*, 218(3):829–837, 2012.

- [79] A. Konak. Two-edge disjoint survivable network design problem with relays: a hybrid genetic algorithm and lagrangian heuristic approach. *Engineering Optimization*, 46(1):130–145, 2014.
- [80] M. Kubry and S. Lim. The flow-refueling location problem for alternative-fuel vehicles. *Socio-Economic Planning Sciences*, 39(2):125–145, 2005.
- [81] M. Kubry and S. Lim. Location of alternative-fuel stations using the flow-refueling location model and dispersion of candidate sites on arcs. *Networks and Spatial Economics*, 7(2):129–152, 2007.
- [82] M. Kubry, L. Lines, R. Schultz, Z. Xie, J.-G. Kim, and S. Lim. Optimization of hydrogen stations in florida using the flow-refueling location model. *International Journal of Hydrogen Energy*, 34(15):6045 – 6064, 2009.
- [83] S. Kulturel-Konak and A. Konak. A local search hybrid genetic algorithm approach to the network design problem with relay stations. In *Telecommunications Modeling, Policy, and Technology*, pages 311–324. Springer, 2008.
- [84] M. Labbé, G. Laporte, I. Martin, and J. Gonzalez. The ring star problem: polyhedral analysis and exact algorithm. *Networks*, 43:177–189, 2004.
- [85] G. Laporte and M. M. Pascoal. Minimum cost path problems with relays. *Computers & Operations Research*, 38(1):165–173, 2011.
- [86] L. Lau, J. S. Naor, M. Salavatipour, and M. Singh. Survivable network design with degree or order constraints. *SIAM Journal on Computing*, 39(3):1062–1087, 2009.
- [87] Y. Lee, S. Y. Chiu, and J. Ryan. A branch and cut algorithm for a steiner tree-star problem. *INFORMS Journal on Computing*, 8(3):194–201, 1996.
- [88] T. Lei. Identifying critical facilities in hub-and-spoke networks: A hub interdiction median problem. *Geographical Analysis*, 45(2):105–122, 2013.
- [89] M. Leitner, I. Ljubić, M. Riedler, and M. AC Ruthmair. On optimal design of charging stations for electric vehicles. Technical report, Institute of Computer Graphics and Algorithms, TU Wien, 2015.

- [90] X. Li, Y. Aneja, and J. Huo. Using branch-and-price approach to solve the directed network design problem with relays. *Omega*, 40(5):672–679, 2012.
- [91] S. Lim and M. Kuby. Heuristic algorithms for siting alternative-fuel stations using the flow-refueling location model. *European Journal of Operational Research*, 204(1):51 – 61, 2010.
- [92] M. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.
- [93] A. Lucena, N. Maculan, and L. Simonetti. Reformulations and solution algorithms for the maximum leaf spanning tree problem. *Computational Management Science*, 7:289–311, 2010.
- [94] K. Manousakis, K. Christodouloupoulos, E. Kamitsas, I. Tomkos, and E. A. Varvarigos. Offline impairment-aware routing and wavelength assignment algorithms in translucent wdm optical networks. *Journal of Lightwave Technology*, 27(12):1866–1877, 2009.
- [95] M. Melaina and J. Bremson. Refueling availability for alternative fuel vehicle markets: Sufficient urban station coverage. *Energy Policy*, 36(8): 3233–3241, 2008.
- [96] M. W. Melaina. Initiating hydrogen infrastructures: preliminary analysis of a sufficient number of initial hydrogen stations in the us. *International Journal of Hydrogen Energy*, 28(7):743–755, 2003.
- [97] M. W. Melaina. Turn of the century refueling: A review of innovations in early gasoline refueling methods and analogies for hydrogen. *Energy Policy*, 35(10):4919–4934, 2007.
- [98] Q. Meng and X. Wang. Intermodal hub-and-spoke network design: Incorporating multiple stakeholders and multi-type containers. *Transportation Research Part B: Methodological*, 45(4):724 – 742, 2011. ISSN 0191-2615.
- [99] S. A. MirHassani and R. Ebrazi. A flexible reformulation of the refueling station location problem. *Transportation Science*, 47(4):617–628, 2013.

- [100] C. Monma and D. Shallcross. Methods for designing communications networks with certain two-connected survivability constraints. *Operations Research*, 37(4):531–541, 1989.
- [101] I. D. Nagtegaal and J. H. J. M. van Krieken. The role of pathologists in the quality control of diagnosis and treatment of rectal canceran overview. *European Journal of Cancer*, 38(7):964 – 972, 2002. ISSN 0959-8049. doi: 10.1016/S0959-8049(02)00056-4.
- [102] M. A. Nicholas and J. Ogden. Detailed analysis of urban station siting for california hydrogen highway network. *Transportation Research Record: Journal of the Transportation Research Board*, 1983(1):121–128, 2006.
- [103] M. A. Nicholas, S. L. Handy, and D. Sperling. Using geographic information systems to evaluate siting and networks of hydrogen stations. *Transportation Research Record: Journal of the Transportation Research Board*, 1880 (1):126–134, 2004.
- [104] Y. M. Nie and M. Ghamami. A corridor-centric approach to planning electric vehicle charging infrastructure. *Transportation Research Part B: Methodological*, 57(0):172–190, 2013.
- [105] M. O’Kelly. Activity levels at hub facilities in interacting networks. *Geographical Analysis*, 18(4):343–356, 1986.
- [106] M. O’Kelly. The location of interacting hub facilities. *Transportation Science*, 20(2):92–106, 1986.
- [107] M. O’Kelly. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32(3):393–404, 1987.
- [108] Y. Ouyang, Q. Zeng, and W. Wei. Dynamic lightpath provisioning with signal quality guarantees in survivable translucent optical networks. *Optics Express*, 13(26):10457–10468, 2005.
- [109] O. Ozkok. Private communication. *Operations Research Scientist*, page Amazon, June 2014.

- [110] S. Pachnicke, T. Paschenda, and P. M. Krummrich. Physical impairment based regenerator placement and routing in translucent optical networks. In *Optical Fiber Communication Conference*, page OWA2. Optical Society of America, 2008.
- [111] K. Park, S. Kang, and S. Park. An integer programming approach to the bandwidth packing problem. *Management science*, 42(9):1277–1291, 1996.
- [112] M. Parker and J. Ryan. A column generation algorithm for bandwidth packing. *Telecommunication Systems*, 2(1):185–195, 1993.
- [113] V. Pillac, M. Gendreau, C. Guret, and A. L. Medaglia. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1 – 11, 2013. ISSN 0377-2217.
- [114] G. A. Plan. An inefficient truth. *Global Action Plan Report*, <http://globalactionplan.org.uk>, 2007.
- [115] W. B. Powell, J. A. Shapiro, and H. P. Simão. An adaptive dynamic programming algorithm for the heterogeneous resource allocation problem. *Transportation Science*, 36(2):231–249, 2002.
- [116] I. Racunica and L. Wynter. Optimal location of intermodal freight hubs. *Transportation Research Part B: Methodological*, 39(5):453 – 477, 2005. ISSN 0191-2615.
- [117] B. Ramamurthy, D. Datta, H. Feng, J. P. Heritage, and B. Mukherjee. Impact of transmission impairments on the teletraffic performance of wavelength-routed optical networks. *Journal of Lightwave Technology*, 17(10):1713, 1999.
- [118] W. W. Recker. The household activity pattern problem: General formulation and solution. *Transportation Research Part B: Methodological*, 29(1): 61–77, 1995.
- [119] G. Rizzelli, A. Morea, M. Tornatore, and O. Rival. Energy efficient traffic-aware design of on-off multi-layer translucent optical networks. *Computer Networks*, 56(10):2443–2455, 2012.

- [120] J. Romm. The car and fuel of the future. *Energy Policy*, 34(17):2609–2614, 2006.
- [121] L. Santos, J. Coutinho-Rodrigues, and J. Current. An improved solution algorithm for the constrained shortest path problem. *Transportation Research Part B: Methodological*, 41(7):756–771, 2007.
- [122] C. Saradhi and S. Subramaniam. Physical layer impairment aware routing (pliar) in wdm optical networks: issues and challenges. *Communications Surveys & Tutorials, IEEE*, 11(4):109–130, 2009.
- [123] M. Savasini, P. Monti, M. Tacca, A. Fumagalli, and H. Waldman. Regenerator placement with guaranteed connectivity in optical networks. *Optical Network Design and Modeling*, pages 438–447, 2007.
- [124] A. Segev. The node-weighted steiner tree problem. *Networks*, 17(1):1–17, 1987.
- [125] A. Sen, S. Murthy, and S. Bandyopadhyay. On sparse placement of regenerator nodes in translucent optical network. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–6. IEEE, 2008.
- [126] G. Shen and R. Tucker. Energy-minimized design for ip over wdm networks. *Optical Communications and Networking, IEEE/OSA Journal of*, 1(1):176–186, 2009.
- [127] M. Z. Spivey and W. B. Powell. The dynamic assignment problem. *Transportation Science*, 38(4):399–419, 2004.
- [128] M. SteadieSeifi, N. P. Dellaert, W. Nuijten, T. Van Woensel, and R. Raoufi. Multimodal freight transportation planning: A literature review. *European journal of operational research*, 233(1):1–15, 2014.
- [129] K. Steiglitz, P. Weiner, and D. Kleitman. The design of minimum-cost survivable networks. *Circuit Theory, IEEE Transactions on*, 16(4):455–460, 1969.
- [130] M. Stoer and F. Wagner. A simple min-cut algorithm. *Journal of the ACM*, 44(4):585–591, 1997.

- [131] T. Taha and G. Taylor. An integrated modeling framework for evaluating hub-and-spoke networks in truckload trucking. *Logistics and Transportation Review*, 30(2):141–166, 1994.
- [132] G. Taylor, T. Meinert, R. Killian, and G. Whicker. Development and analysis of alternative dispatching methods in truckload trucking. *Transportation Research Part E: Logistics and Transportation Review*, 35(0):191 – 205, 1999.
- [133] I. Tomkos, E. Palkopoulou, and M. Angelou. A survey of recent developments on flexible/elastic optical networking. In *Transparent Optical Networks (ICTON), 2012 14th International Conference on*, pages 1–6. IEEE, 2012.
- [134] C. Upchurch and M. Kuby. Comparing the p-median and flow-refueling models for locating alternative-fuel stations. *Journal of Transport Geography*, 18(6):750 – 758, 2010. Special Section on Alternative Fuels and Vehicles.
- [135] U.S. Census Bureau. Urban, urbanized area, urban cluster, and rural population, 2010 and 2000: United states, 2010. URL <http://www.census.gov/geo/reference/ua/urban-rural-2010.html>. accessed May 18, 2014.
- [136] H. Üster and P. Kewcharoenwong. Strategic design and analysis of a relay network in truckload transportation. *Transportation Science*, 45(4):505–523, 2011.
- [137] H. Üster and N. Maheshwari. Strategic network design for multi-zone truckload shipments. *IEEE Transactions*, 39(2):177–189, 2007.
- [138] R. Vachani, A. Shulman, and P. Kubat. Multicommodity flows in ring networks. *INFORMS Journal on Computing*, 8(3):235, 1996. ISSN 10919856.
- [139] H. A. Vergara and S. Root. Mixed fleet dispatching in truckload relay network design optimization. *Transportation Research Part E: Logistics and Transportation Review*, 54:32–49, 2013.

- [140] S. Voß. The steiner tree problem with hop constraints. *Annals of Operations Research*, 86:321–345, 1999.
- [141] B. Wagner. Model formulations for hub covering problems. *Journal of the Operational Research Society*, 59(7):932–938, 2007.
- [142] X. Wang, M. Brandt-Pearce, and S. Subramaniam. Impact of wavelength and modulation conversion on translucent elastic optical networks using milp. *arXiv preprint arXiv:1412.8086*, 2014.
- [143] Y. Wang and C. Wang. Locating passenger vehicle refueling stations. *Transportation Research Part E: Logistics and Transportation Review*, 46(5):791–801, 2010.
- [144] Y.-W. Wang and C.-C. Lin. Locating road-vehicle refueling stations. *Transportation Research Part E: Logistics and Transportation Review*, 45(5):821 – 829, 2009.
- [145] Y.-W. Wang and C.-C. Lin. Locating multiple types of recharging stations for battery-powered electric vehicle transport. *Transportation Research Part E: Logistics and Transportation Review*, 58(0):76 – 87, 2013.
- [146] X. Xi, R. Sioshansi, and V. Marano. Simulationoptimization model for location of a public electric vehicle charging infrastructure. *Transportation Research Part D: Transport and Environment*, 22(0):60 – 69, 2013.
- [147] H. Yaman. The hierarchical hub median problem with single assignment. *Transportation Research Part B: Methodological*, 43(6):643 – 658, 2009. ISSN 0191-2615.
- [148] H. Yaman, B. Y. Kara, and B. Tansel. The latest arrival hub location problem for cargo delivery systems with stopovers. *Transportation Research Part B: Methodological*, 41(8):906 – 919, 2007. ISSN 0191-2615.
- [149] X. Yang and B. Ramamurthy. Dynamic routing in translucent wdm optical networks: the intradomain case. *Lightwave Technology, Journal of*, 23(3): 955–971, 2005.

- [150] X. Yang and B. Ramamurthy. Sparse regeneration in translucent wavelength-routed optical networks: Architecture, network design and wavelength routing. *Photonic network communications*, 10(1):39–53, 2005.
- [151] J. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1971.
- [152] E. Yetginer and E. Karasan. Regenerator placement and traffic engineering with restoration in gmpls networks. *Photonic Network Communications*, 6(2):139–149, 2003.
- [153] B. Yildiz and O. E. Karasan. Regenerator location problem in flexible optical networks. *Operations Research Submitted-Under Review*, 2014.
- [154] B. Yildiz and O. E. Karasan. Regenerator location problem and survivable extensions: A hub covering location perspective. *Transportation Research Part B: Methodological*, 71:32–55, 2015.
- [155] B. Yildiz, O. Arslan, and O. E. Karasan. A branch and price approach for routing and refueling station location model. *European Journal of Operational Research*, 2015.
- [156] M. Zarandi and S. Davari. Design of a reliable hub-and-spoke network using an interactive fuzzy goal programming. In *2011 IEEE International Conference on Fuzzy Systems*, pages 2955–2959. IEEE, 2011.
- [157] B. Zeng, Y. An, Y. Zhang, and H. Kim. A reliable hub-spoke model in transportation systems. In *In: Proceedings of the 4th International Symposium on Transportation Network Reliability, Minneapolis, Minnesota, USA, 22-23 Jul 2010*, 2010.
- [158] S. Zsigmond, G. Németh, and T. Cinkler. Mutual impact of physical impairments and grooming in multilayer networks. In *Optical Network Design and Modeling*, pages 38–47. Springer, 2007.