

# **SURVIVABLE FIBER OPTICAL NETWORK DESIGN**

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE  
OF BILKENT UNIVERSITY  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR  
THE DEGREE OF  
MASTER OF SCIENCE  
IN  
INDUSTRIAL ENGINEERING

By  
Seçil Sözüer  
September, 2015

SURVIVABLE FIBER OPTICAL NETWORK DESIGN

By Seil Sözüer

September, 2015

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

---

Assoc. Prof. Dr. Oya Karaşan(Advisor)

---

Assoc. Prof. Dr. Osman Oğuz

---

Prof. Dr. Serpil Erol

Approved for the Graduate School of Engineering and Science:

---

Prof. Dr. Levent Onural  
Director of the Graduate School

# ABSTRACT

## SURVIVABLE FIBER OPTICAL NETWORK DESIGN

Seçil Sözüer  
M.S. in Industrial Engineering  
Advisor: Assoc. Prof. Dr. Oya Karaşan  
September, 2015

This thesis presents a study on a survivable extension of a network design problem of one of the largest Internet service providers operating in Turkey. In a previous study, this problem is defined as the “Green Field Network Design Problem” where the aim is to design a cost effective fiber optical network that will provide high speed and high quality Internet access from a prelocated central station to a set of aggregated demand nodes. In order to attain a required service level, insertion loss, speed level and distance limitations are considered simultaneously. The Internet access from the central station to the demand nodes can be provided either directly by installing fiber optical wires or indirectly by utilizing special telecommunication devices called “Passive Splitters”. Passive splitters copy and split the data into the output ports, and they can be considered as hubs since they consolidate and disseminate the data. In this study, we consider the survivable version of this problem: “Survivable Green Field Network Design Problem”. In order to ensure survivability, we seek to find 2-node disjoint paths for every demand node such that the fixed costs of installing passive splitters and the fiber wiring costs are minimized. A mathematical model is constructed. In order to solve problems with higher dimensions, heuristic algorithms are also proposed. A data set belonging to Kartal district of İstanbul is used to test the performances of mathematical model and the heuristics, and the results of the computational study are reported.

*Keywords:* Fiber optical telecommunication network, Survivable network design, Primary and secondary paths, Hub-and-spoke networks.

# ÖZET

## GÜVENİLİR FİBER OPTİK AĞ TASARIMI

Seçil Sözüer

Endüstri Mühendisliği, Yüksek Lisans

Tez Danışmanı: Doç. Dr. Oya Karaşan

Eylül, 2015

Bu tez, Türkiye’de faaliyet gösteren en büyük internet sağlayıcısının ağ tasarımı probleminin güvenilir versiyonu hakkındadır. Önceki bir çalışmada, “Yeşil Bölge Ağ Tasarımı Problemi” olarak adlandırılan problemdeki amaç, en az maliyetle fiber optik ağ tasarlamaktır ve merkez istasyonundan, bütünleşik talep düğüm noktalarına, yüksek hızda ve kalitede internet erişimi sağlamaktır. Merkez istasyonunun yerleşkesi, önceden tayin edilmiştir. Mecburi servis düzeyinin sağlanması için, ağ içi hatlardaki iletim kaybı, hız seviyesi ve mesafe kısıtlamaları eş zamanlı olarak düşünülmüştür. Merkez istasyonundan talep noktalarına internet erişimi, iki yolla sağlanabilmektedir: (i) Direkt olarak fiber bağlantılarının kurulmasıyla (ii) Dolaylı olarak “pasif bölücü” adı verilen spesifik telekomünikasyon cihazlarının kullanılmasıyla. Pasif Bölücülerin görevi, gelen verileri kopyalama ve dağıtmaktır ve bu cihazlar ana dağıtım üssü olarak ele alınabilir. Bu çalışmada, bahsedilen problemin güvenilir bir versiyonu olan “Güvenilir Yeşil Bölge Ağ Tasarımı Problemi” incelenmiştir. Güvenilirliği sağlamak amacıyla, her bir kaynak ve hedef ikilisi için düğüm noktaları temelli 2-ayrıt yol temin edilmiştir. Aynı zamanda, pasif bölücü kurulum sabit giderlerini ve fiber kablolama giderlerini en aza indirgeyecek şekilde ağ tasarımı oluşturulmuştur. Matematiksel bir model geliştirilmiştir. Ayrıca, büyük ölçekli problemleri çözebilmek için sezgisel algoritmalar sunulmuştur. Matematiksel modellerin ve sezgisel yaklaşımların performansını test etmek amacıyla İstanbul’un Kartal ilçesine ait veri seti kullanılmıştır. Hesaplama sonuçları rapor edilmiştir.

*Anahtar sözcükler:* Fiber optik telekomünikasyon ağı, Güvenilir ağların tasarımı, Birincil ve ikincil yollar, Ana dağıtım üssü.

## Acknowledgement

Foremost, I would like to express my gratitude to my advisor, Assoc. Prof. Dr. Oya Ekin Karařan, who has guided me with her knowledge and patience throughout this research. She provided many insightful ideas and discussions about the problem. It was a nice experience to work in this project under her supervision.

I am also grateful to my committee members, Assoc. Prof. Dr. Osman Ođuz and Prof. Dr. Serpil Erol, for accepting to read this thesis

I owe a lot to my dear family for their love, care and encouragement in every stage of my life. They have always been on my side with their support and love. It is a blessing that I have a wonderful sister, Sibel Sözüer Zorlu. She inspired me in so many ways. My father was so loving, caring and attentive about me. My mother supported and helped me as much as she can. My favorite aunts, Gülay and Nebahat, were also very caring and loving. I feel so lucky to have them all. I managed to overcome the difficulties I faced thanks to them. Words alone cannot explain my love and gratitude for them.

I am grateful to Sevgican, Nil, Ramez and Nihal for their valuable friendship and support. We shared a lot with Sevgican especially during my masters, and I appreciate all of Nil's, Ramez's and Nihal's help and support.

Special thanks to Ahmet Dünder Sezer and Semih Kaldırım for being there when I needed them most. I feel lucky to be their friend.

I wish to thank Sinejan for her friendship and motivation. Although I have not seen her in person for a long time, I feel the warmth of her friendship despite of the distances. I miss the enjoyable and fun times we spent together during our senior year. She always managed to put a smile on my face.

I have met a lot of amazing and sweet people during my last year. Dođan,

Hüsna, Ömer and Gökhan, thank you for your friendship. You have made this period special and enjoyable.

I want to thank Sibel and Tuğçe for their continual friendship that started in high school. I am very happy that we managed to stay close despite the passing years and distances.

I would like to thank The Scientific and Technological Research Council of Turkey (TÜBİTAK) for providing financial support during my graduate study. I am also grateful to my university for providing high quality education and the opportunity for me to participate in Erasmus program in Brussels.

During seven years at Bilkent, I learned and developed a lot, and I had many great experiences that I will reminiscence for the rest of my life.

# Contents

- 1 Introduction and Problem Definition** **1**
  
- 2 Literature Review** **6**
  - 2.1 Hub Location Problems . . . . . 6
  - 2.2 Telecommunication Network Design . . . . . 8
  - 2.3 Survivable Network Design . . . . . 12
  - 2.4 Clustering Methods . . . . . 15
  
- 3 Mathematical Model** **19**
  - 3.1 MIP Model for “Survivable Green Field Network Design Problem” 20
  - 3.2 The variants of the problem . . . . . 30
  
- 4 Heuristic Algorithms** **33**
  - 4.1 Ring Creation Algorithm . . . . . 34
  - 4.2 Clustering Algorithm . . . . . 51

- 5 Data and Computational Analysis** **61**
- 5.1 Data . . . . . 62
- 5.2 Computational Analysis . . . . . 66
  - 5.2.1 Performance of MIP Start Option . . . . . 67
  - 5.2.2 The effect of the Location of Central Station . . . . . 68
  - 5.2.3 The outputs of Ring Creation Algorithm . . . . . 71
  - 5.2.4 The outputs of Clustering Algorithm . . . . . 75
  - 5.2.5 Performance of Heuristic Algorithms . . . . . 78
  
- 6 Conclusion** **81**

# List of Figures

1.1	2-node disjoint paths . . . . .	4
2.1	FTTX Services [1] . . . . .	10
2.2	Fully Connected / Tree Network [2] . . . . .	11
2.3	Star / Star Network [2] . . . . .	12
2.4	Exclusive and Nonexclusive Clustering . . . . .	16
3.1	Example for Insertion Loss and Speed Level . . . . .	25
4.1	Ring Structure and 2 Node-Disjoint Paths . . . . .	35
4.2	Set E generation . . . . .	36
4.3	Ring Creation Algorithm- Expanding the Ring . . . . .	40
4.4	Ring Creation Algorithm- Paths of Node $i$ and Node $k$ . . . . .	41
4.5	Ring Algorithm - Serving nonPS node $h$ . . . . .	44
4.6	Ring Example Network . . . . .	48
4.7	Clustering Example Network- $\beta_1 = 4$ and $\beta_3 = 6$ . . . . .	55

5.1	The Map of Kartal . . . . .	63
5.2	The Map of Kartal with 7 and 14 demand nodes . . . . .	64
5.3	Optimum Designs for 7 demand node network and different central locations . . . . .	70
5.4	Ring Algorithm- Output of instance K15C . . . . .	74
5.5	Clustering Algorithm- Output of instance K15C . . . . .	77

# List of Tables

5.1	Features of the data set . . . . .	65
5.2	Comparison based on MIP Start . . . . .	67
5.3	Optimal Results of instances with 8 node cardinality . . . . .	69
5.4	Best Integer Result of instances with 15 node cardinality . . . . .	69
5.5	Ring Algorithm- Results of instances with 8 node cardinality . . .	71
5.6	Ring Algorithm- Results of instances with 15 node cardinality . .	72
5.7	Ring Algorithm- Results of instances with 45 node cardinality . .	73
5.8	Clustering Algorithm- Results of instances with 15 node cardinality	75
5.9	Clustering Algorithm- Results of instances with 45 node cardinality	76
5.10	Comparison for Instances with 8 and 15 node cardinality . . . . .	79
5.11	Percentage Comparison for Instances with 8 and 15 node cardinality	79
5.12	Comparison for Instances with 45 node cardinality . . . . .	80

# Chapter 1

## Introduction and Problem Definition

Communication is to exchange information between two entities, and it has always been a very crucial issue since it enables knowledge expansion. Wide variety of ways and products have been invented for communication. For instance, during the early ages, some of these various communication ways are audio messages (such as coded drumbeats and lung-blown horns) and visual signals (such as beacons, smoke signals and signal flags). Due to the developments in the field of electrical and electromagnetic technologies, the communication structure has evolved into telecommunication which is the exchange of information by electronic means. For instance, telegraph, telephone, teleprinter, radio, fiber optics, and communications satellites are some of the telecommunication devices.

Networks are used for modeling purposes in many settings and telecommunication is one of them. A basic telecommunications networks has three basic units: (i) A transmitter that takes information and converts it to a signal.(ii) A receiver that takes the signal from the transmission medium and converts it back into usable information. (iii) A transmission medium that carries the signal and provides data exchange (voice, video, signals) among network's constituent parts. It acts as a communication channel between a transmitter and a receiver. The medium

of signal transmission can be electrical wires (such as copper cables or coaxial cables), fiber optical cables, radio links or satellite links [3]. In a generic network representation, a transmitter point can be regarded as a source node, a receiver point can be considered as a demand node and a transmission medium can be thought as a collection of edges.

Fiber-optic communication, which is a specific and recently developed version of telecommunication, is a method of transferring data from a transmitter point to a receiver point through fiber optical wires [4]. Data transmission is enabled by sending pulses of light through a fiber optical wire, hence fiber optical wire is utilized as a transmission medium. Fiber-optic communication has become popular and replaced electrical wire communications to a high extent. Because fiber-optic communication is an appealing alternative due to its immense data capacity and high speed. Some application areas of fiber-optic communication are telephone signal transmission, internet communication, and cable television signal transmission. Especially, high speed technologies adopted the usage of fiber optic communication. Modern telecommunication requires more data transfer at higher speed level (especially due to the emergence of smart phones, tablets, computers) and fiber-optic communication has the capability to handle these requirements. For example, France Telecom-Orange decided to convert its copper wire network to a fiber optic network [1]. Although fiber-optic communication offers a high speed data transmission, the signal quality should be taken into consideration since the signals tend to become distorted and weak over long distances. In order to transmit the data, special devices called optical splitters are utilized: Active splitter and Passive splitter. Active splitters distribute the received data without copying. They serve as a regulator for data and prevent the supply rail fluctuations. Hence, active splitters are utilized for data transmission within the stations. They are not used for distribution purpose since they do not copy the signal and their splitting capacity is very small. On the other hand, Passive Splitters are used to carry the optical signal by copying and distributing the data to the ports [3]. These ports will be the specified receiver points and/or another passive splitters. Based on the passive splitters' types, they have different splitting capacities. Passive Splitters are also referred to as fiber optic splitters.

A specific fiber optical network design problem has been studied in [5]. This problem is based on an application of one of the largest Internet providers in Turkey. The authors defined the problem as the “Green Field Network Design Problem” where the aim is to design a cost-effective fiber optical network that will provide high quality Internet access from a prelocated central station (transmitter point) to a set of aggregated demand nodes (receiver points). In order to attain a required service level, the authors took insertion loss, bandwidth level and distance limitations into consideration simultaneously. Bandwidth level corresponds to the number of data bits per unit time, hence bandwidth corresponds to the rate of data transferred. Therefore, high bandwidth Internet access means that the speed of the data transmission is high. The access from central station to the demand nodes can be enabled either directly by installing optical fiber wires or indirectly by utilizing Passive Splitters. There are 4 types of Passive Splitters and they have different data splitting capacity. The problem adopts a hub location perspective since Passive Splitters are regarded as hubs.

Survivability is the ability of a system to be able to continue to function during and/or after a disruption, malfunction or failure in the network. Equipment failures may occur due to various reasons such as construction and destructive natural events (such as earthquakes, tsunamis, tornadoes). Survivability is a major consideration while designing telecommunications networks since the design should hedge against possible malfunctions at the passive splitters and continue to satisfy the service requirements. Considering this aspect, in this study, we focus on the survivable version of this problem: “Survivable Green Field Network Design Problem”. In order to ensure survivability, we seek to find 2-node disjoint paths for every demand node such that the fixed costs of establishing passive splitters and the fiber wiring costs are minimized.

Different schemes exist in regards of survivability. The most general approach is to connect the demand nodes to central station via more than a single path. Thanks to this survivable design, in case a malfunction on a path occurs, another path can be utilized for the required data transmission. In general, the paths are chosen as either node-disjoint or edge-disjoint. Usually, the survivable network requires two disjoint paths since a lot of disjoint paths will result a more dense

(and more expensive) network. Figure 1.1 represents 2-node disjoint paths for a source destination pair,  $s-t$ . The first path is  $s-1-3-5-t$  and the second path is  $s-2-4-t$  as illustrated.

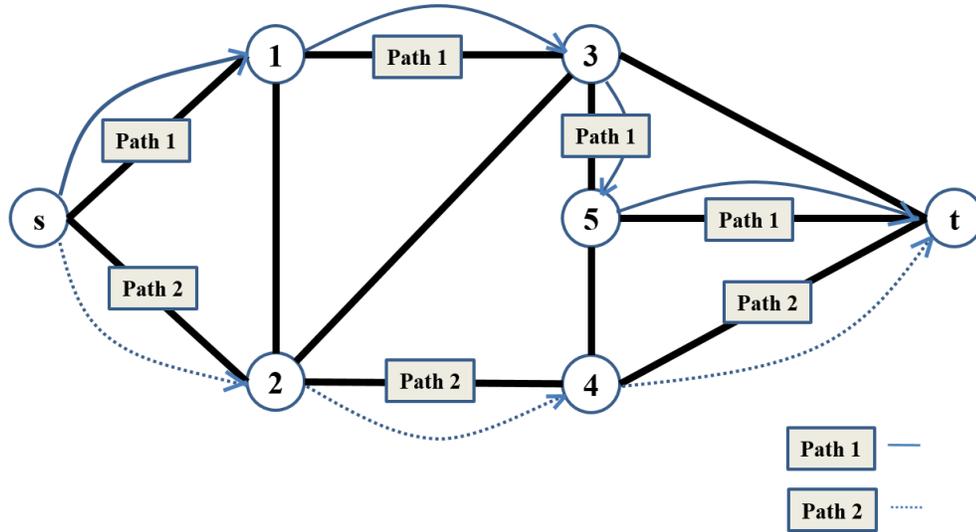


Figure 1.1: 2-node disjoint paths

The main contribution of this study is the consideration of the survivability issue to a real-life telecommunication network design problem and the proposed heuristic algorithms. The first heuristic is called “Ring Creation Algorithm”. We aim to construct fiber links in a ring form for the demand nodes in order to maintain 2-node connectivity. The second heuristic, “Clustering Heuristic Algorithm”, aims to group the given nodes into the clusters, and solves each sub-problem that corresponds to the generated cluster independently. Then, the separate results are combined to give a solution for the original problem. Typically, the exact number of the clusters is given beforehand. According to the “Clustering Heuristic Algorithm” we developed, the number of clusters are decided based on the given network distance data set. This grouping approach is more flexible and data dependent since we utilize the given network’s data set instead of forced input parameters.

The remainder of this thesis is organized as follows: In the next chapter, we expand this research and present literature on hub location, telecommunication

network design, survivability issue and clustering approach. Chapter 3 illustrates the mathematical model for the “Survivable Green Field Network Design Problem”. In Chapter 4, heuristic algorithms are explained. Results of the computational study are provided in Chapter 5. A conclusion and possible future research directions are given in Chapter 6.

# Chapter 2

## Literature Review

Our literature review consists of four parts. The first part will illustrate the hub location problems since the “Survivable Green Field Network Design Problem” is approached from a hub location perspective. Then, telecommunication networks problems will be discussed. In the third part, we will present a review on survivable network design. Lastly, clustering methods will be explained since one of the heuristics algorithms we proposed is utilizing clustering approach.

### 2.1 Hub Location Problems

The cost-effective networks design problem is a major area of interest and a widely-adapted architecture for designing it is to utilize hub facilities. Hub facilities serve as consolidating and disseminating points and they offer indirect connection among the nodes instead of costly direct connections. The utilization of hub facilities provide a fewer number of links compared to point-to-point direct connections. For instance, a hub can be an airport, seaport, warehouse, concentrator or any other facility according to the corresponding context. Due to its advantages about the network construction and operation cost, the hubs are commonly utilized.

The Hub Location Problem is a specific kind of facility location problem. This problem seeks to find the location of hub facilities and allocation of the non-hub nodes to these hubs such that the origin-destination pair demands will be satisfied in a cost effective manner. A hub location perspective can be adopted while designing a network and such a network is called “Hub-and-Spoke network”. The non-hub nodes are referred to as “spokes”. An arc that connects a spoke and a hub is called an “access arc” whereas a “hub arc” provides inter hub connection. The aim of a ‘Hub-and-Spoke network” is to determine the optimal number and location of hubs, and the allocation structure of the spokes to the hubs. In a traditional “hub-and-spoke network”, direct connection between spokes is not allowed and any flow has to pass through the hubs. Generally, inter-hub connection cost is lower than spoke-hub connection due to economies of scale. Usually, paths between origin-destination pairs visit at most two hubs [6]. There are two types of assignment structures between spoke and hub nodes: Single and Multi-Allocation Scheme. In single allocation scheme, each spoke node is assigned to exactly one node. Hence all the incoming and outgoing flows are served via that node. In multi-allocation scheme, a spoke node can be assigned to more than one hub. For a specific hub location problem, there might be other considerations such as the type and capacity of the hub. In order to evaluate a network, many factors are considered in the design such as cost, capacity, reliability and performance.

The first mathematical model for the hub location problem is provided by O-Kelly [7] and five fundamental hub location problems are introduced by Campbell [8]. These variants are p-hub median, hub covering, p-hub center, the multi-allocation uncapacitated hub location and hub arc location problems. (i) p-hub median problem: The aim is to locate p hub facilities so as to minimize the total transportation cost for serving the origin-destination pairs. The fixed costs of opening facilities are ignored in the objective function. Single and multi allocation schemes are presented. (ii) Hub covering problem: The aim is to find the minimum number of hubs needed for covering the demands. Demand nodes are considered as covered if they are within a predefined distance. Different types of coverage criteria are considered in the literature. (iii) p-hub center problem: The aim is to locate p hubs and to allocate non-hub nodes to hub nodes such

that the maximum transportation time between any origin destination pair is minimized. (iv) Multi-allocation uncapacitated hub location problem: The aim is minimize the total cost and multiple allocation scheme is adopted. (v) Hub Arc Location problem: the aim is to locate hub arcs instead of locating hub facilities. The reduced unit flow costs are considered. This problem is the most recently developed model [9]

Hub Location became a major area of research and an important sub-field of location science. It is a widely studied area in the literature and plenty of problem variants, models and algorithms are proposed. Comprehensive survey and review of hub location research has been conducted by Alumur and Kara (2008) [10], Campbell, Ernst and Krisnamoorthy (2002) [11] and Klincewicz (1998) [12].

Hub-and-spoke networks are also utilized for telecommunication network design. Klinewicz's study is about telecommunication network design based on hub location perspective. The predominant telecommunication network design costs are related to the link (fiber optic lines, copper cables etc...) establishment rather than transportation. Hence, the objective function structures of transportation and telecommunication network problems are rather different.

## 2.2 Telecommunication Network Design

With the increase of knowledge and usage about electricity as a means of data transfer, telecommunication has emerged and grown tremendously. Hoesel provided a history of the progress in telecommunications in a nutshell in [13]. He gives information on the most fundamental devices of telecommunications and elaborates on its working principles. The earliest invention of telecommunication is "optical telegraph" which was founded by Claude Chappe in 1793. In order to transmit the data, the optical telegraph uses light signals which act as mirrors. The following invention is "electrical telegraph" which was founded by Samuel Morse in 1831. Graham Bell invented "telephone" in 1876. The working principle of telephone is to utilize electrical signal translation for enabling

voice transfer. Another important invention is “wireless telegraphy” which was founded by Marconi in 1897. Wireless telegraphy works by using electromagnetic waves. Nowadays, all these brilliant inventions are outdated to a high extent.

After 1950s, digital signal transmission became more prevalent. This technology evolved into Internet. Internet is a system of interconnected computer networks and it provides the connection of the devices globally. Internet was initially founded in order to provide communication among different universities. Later, plenty of services are enabled such as e-mail sending/receiving, file transfer, video streaming applications etc... Due to these numerous service applications, Internet traffic has increased tremendously and it became a crucial issue to provide high-quality data transmission in a fast and secure way. Internet has become a vital element of telecommunication.

Due to the increment in demand, fiber optic networks have become important since they provide high bandwidths, high speed and security. Some architectures of fiber optic communication are FTTC (Fiber To The Curb), FTTB (Fiber To The Building) and FTTH (Fiber To The Home). The optical fiber stops in the neighborhood (curb) of the subscribers (FTTC), or at their building (FTTB), or at their home FTTH (Fiber To The Home). FTTH is the most appropriate alternative for a long term objective [1]. Figure 2.1 illustrates these services.

The design of telecommunication networks drew attention of many researchers from many disciplines such as operations research, mathematics and electrical engineering. A telecommunication network is composed of three main parts: (i) A transmitter is the source point for the information. It takes information and changes it into a signal. (ii) A transmission medium acts as a communication channel between a transmitter - receiver pair. It carries the signal and enables signal exchange between a transmitter and a receiver point. (iii) A receiver is the destination point for the information. It takes the signal from the transmission medium and converts it back into usable information. Furthermore, the telecommunication network has a special device that will gather and compress the signals from the transmitters and forward the signals to the receiver points. This device acts as a hub facility since it consolidates and disseminates the flow (in

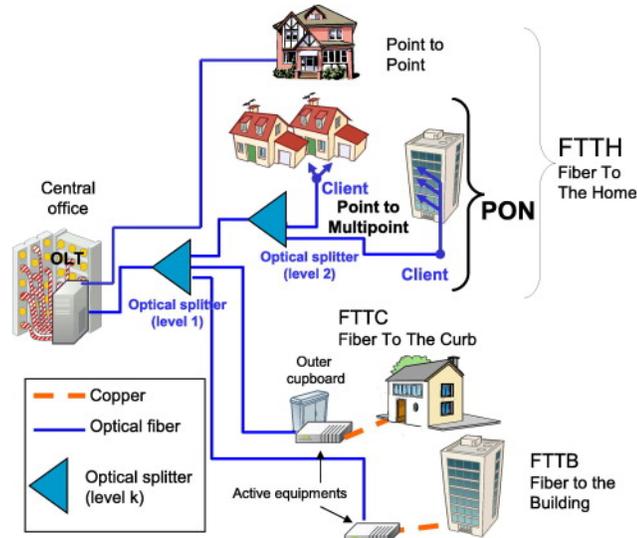


Figure 2.1: FTTX Services [1]

our case, the signals). These devices are often called routers, gateways, switches and multiplexers and concentrators according to their corresponding usage [12]. For instance, gateway nodes are at the highest-level hub nodes. Gateways are connected to switches which are low-level hub nodes. Switches provide access for the spoke nodes (or non-hub nodes). Due to the advantages of hub facility usage, telecommunication networks are often designed and considered as a hub-and-spoke network.

Generally, telecommunication networks have a hierarchical structure. There are many multi-layers associated with it. A generic telecommunication network consists of backbone network and access network (or tributary network). Backbone network interconnects hub nodes and provide connection from central station. Hub arcs constitute hub network. Access network provides connection for spoke nodes and it consists of access arcs. The backbone and access networks might be chosen to be designed independently due to the problem size and application setting. In his review, Klincewicz considered the backbone and access network design problem as an integrated problem. There are many different variants of backbone and tributary network design. Some of them are illustrated by [2] and

can be seen from Figure 2.2 and 2.3. The notation “Backbone Structure / Access Network Structure” is adopted in order to specify the particular design.

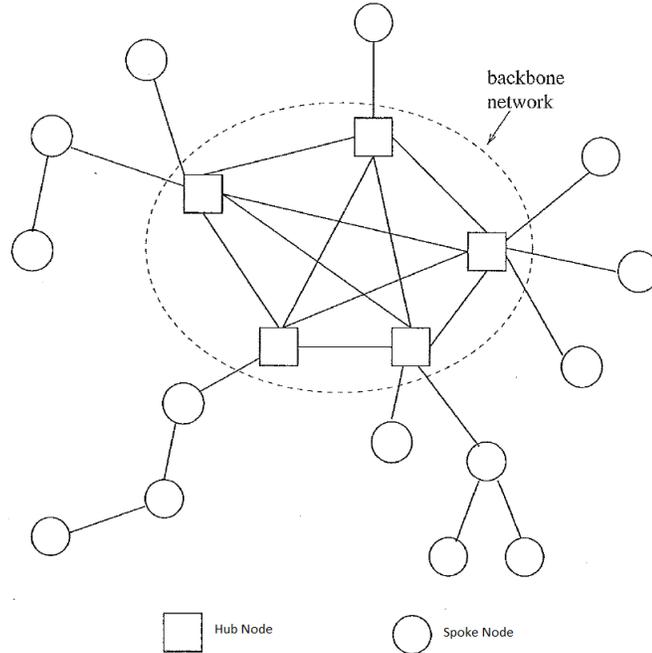


Figure 2.2: Fully Connected / Tree Network [2]

A survey about the significant examples and algorithms of the telecommunication networks is conducted in [2]. The authors considered the backbone and access network design problem as an integrated problem. Since fiber optic communication has been proven as a promising technology, the fiber optic network design problems are also studied.

In the literature, there are also some recent studies which consider the design of fiber optic access networks such as [14], [15] and [1]. In [14], the authors work on the design of the access network on a tree topology. They aim to find the optimal location of the switch and allocation of user nodes to these switches under the constraints such as switch port constraint, switch capacity constraint and routing constraint. They develop a mixed integer program and a tree-partitioning heuristic algorithm. In [15], the aim is to find the optimal location of fiber optical splitters (or passive splitters) such that every demand node will be connected to

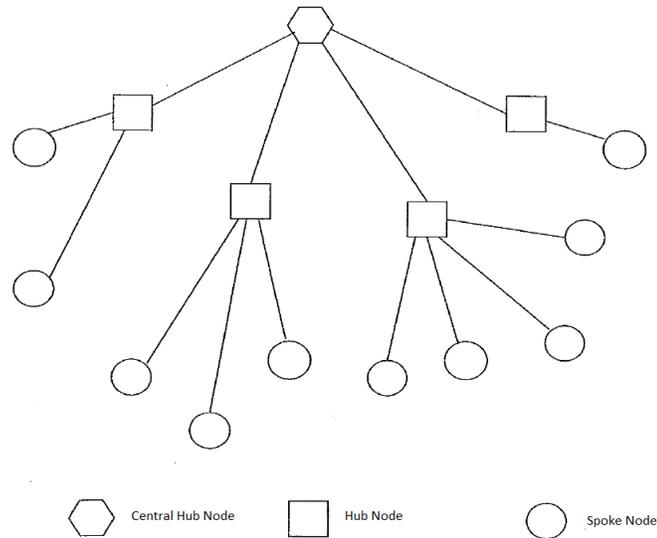


Figure 2.3: Star / Star Network [2]

the central office. As explained in the previous chapter, fiber optical splitters are utilized for splitting and copying the data into several outputs according to a certain ratio. There are two schemes of splitters in this problem: Single and Double Splitters. They also proposed a heuristic algorithm which works in a greedy fashion. In [1], the capacitated version of access network design is considered and FTTH service has been adopted. The authors aim to locate splitters and install fiber wires in an existing network infrastructure while not exceeding edge capacities.

## 2.3 Survivable Network Design

Survivability is a major and critical concern for network performance evaluation. Due to service quality requirements, the network is expected to hedge against failures, disruptions and malfunctions. Survivability and reliability has been studied in the network design literature. Reliable  $p$ -hub location problems are introduced

by O’Kelly and Kim [16]. They developed  $p$ -hub maximum reliability and  $p$ -hub mandatory dispersion models with single and multiple allocation. In their study, the nodes and edges have reliability values which are associated with the probability that the hub node or edge transmits flows within a given time period without failure. Their first model,  $p$ -hub maximum reliability problem, aims to maximize network performance in terms of reliability by locating hubs for transmitting flows. For the second model,  $p$ -hub mandatory dispersion problem, an additional design requirement is considered. The authors seek to design a network where the hub nodes are dispersed from each other based on a specified threshold value. They analyzed the cascading Internet failure from a network reliability perspective. They also mention the earthquake in Taiwan in 2006 in order to point out the significance of dispersion in the hub-and-spoke network design. They also proposed two heuristics based on Tabu Search and Hybrid Search [16].

Different versions of reliable hub-and-spoke network problems in transportation systems are developed by Zhang and Zeng [17]. They claim that in case of malfunctions in equipment, the current strategies are inefficient and costly. Some of these strategies are delaying, canceling and rerouting in air transportation, and network peering in telecommunications systems. They aim to minimize the operating cost during both normal and disruptive situations. They utilize backup hubs and alternative routes in order to avoid vulnerability and attain the required service level. The flow will utilize the primary route in case no hub on the route fails. If a hub on the primary route fails, then backup hubs and alternative routes are utilized.

Typically, survivability requirement is satisfied by enabling two different routes between an origin and a destination pair. In this case, a malfunction will not interrupt the data transfer since a secondary route can be utilized. Therefore, we aim to provide alternative routes for the user nodes. Generally, the failures are assumed to occur at a node or edge. Hence, the typical focus is on the edge disjoint and node disjoint paths while designing networks. Survivability and cost of the network are clashing objectives. Increasing the network survivability too much will result in a dense and expensive network. Likewise, a cost-effective

network is often not survivable. Therefore, both criteria should be considered simultaneously while evaluating a network's performance.

The problem of finding disjoint paths is also studied in the literature. A lot of MIP formulations, heuristic and exact algorithms are developed. The most famous algorithm belongs to Suurballe [18]. He aims to find  $K$  node-disjoint paths while minimizing the total cost. The total cost consists of used arc lengths on paths between every origin-destination pair. He also develops a labeling algorithm for finding  $K$  node-disjoint paths. He also presented his study on edge-disjoint path finding in [19].

Damcı and Karaşan studied a cost-effective survivable telecommunications design [20]. They seek to find 2-edge disjoint paths (primary and secondary paths) for every possible origin destination node pair. Different but relevant routing cost structures are considered for primary and secondary paths along with fixed and variable edge costs. They also proposed heuristic algorithms which utilize Suurballe's and Dijkstra's algorithms for constructing an initial solution. Improvement heuristics are also developed and tested on a large bed of problem instances.

Another survivable telecommunications network design has been proposed by Yıldız and Karaşan [21]. They regard the telecommunications networks as a hub-and-spoke network where the hubs correspond to the regenerators. An optical signal can traverse a certain distance limit without quality degradation and this threshold limit value is called the reach of this optical signal. Regenerators are special telecommunication devices that enable the extension of the reach of an optical signal. They aim to install the minimum number of regenerators while maintaining the communication among every possible node pair. Their objective function only consists of the regenerator installment cost since the link establishment costs are not presented. Within the scope of this study, survivability issue has been analyzed into two dimensions: partial and full survivability. Under partial survivability, the network design hedges against the malfunctions in regenerators. Full survivability ensures that the design maintains the required service in case of a failure in any node.

## 2.4 Clustering Methods

Clustering is the task of grouping and organizing the given objects into subsets based on a specific criterion or pattern. The objects within a cluster are expected to be more similar to each other than the objects in distinct clusters. Clustering is an important technique for data analysis and interpretation since it helps to explore the data structure. Clustering approach has been widely adopted in many fields and the corresponding criteria for the clusters differ according to these application areas. For instance, the criterion might be gender, age range and education level for grouping people in a marketing application. Biological studies utilize clustering in order to classify the plants and animals according to their features. Earthquake studies utilize the clusters for grouping the dangerous zones. Consequently, the criteria change with respect to the context. Another important issue is about the evaluation of the output clusters. There is no single “best” criterion for obtaining a cluster since no precise definition for “cluster” exists [22].

According to the features of the grouping approach, clustering methods can be categorized as follows: (i) Exclusive versus Nonexclusive (ii) Fuzzy versus Non-fuzzy (iii) Intrinsic versus Extrinsic (iv) Hierarchical versus Partitional (v) Heterogeneous versus Homogeneous (vi) Complete versus Partial.

### **Exclusive versus Nonexclusive:**

An exclusive classification is a partition of the set of objects where each object belongs to exactly one cluster. In non-exclusive (or overlapping) clustering, objects may be assigned to multiple clusters. For instance, a grouping of people by age and sex is exclusive whereas a grouping by disease category is nonexclusive since a person can have several diseases at the same time. Figure 2.4 illustrates exclusive and nonexclusive clustering.

### **Fuzzy versus Non-fuzzy:**

Fuzzy (or probabilistic) clustering is a type of nonexclusive classification. In

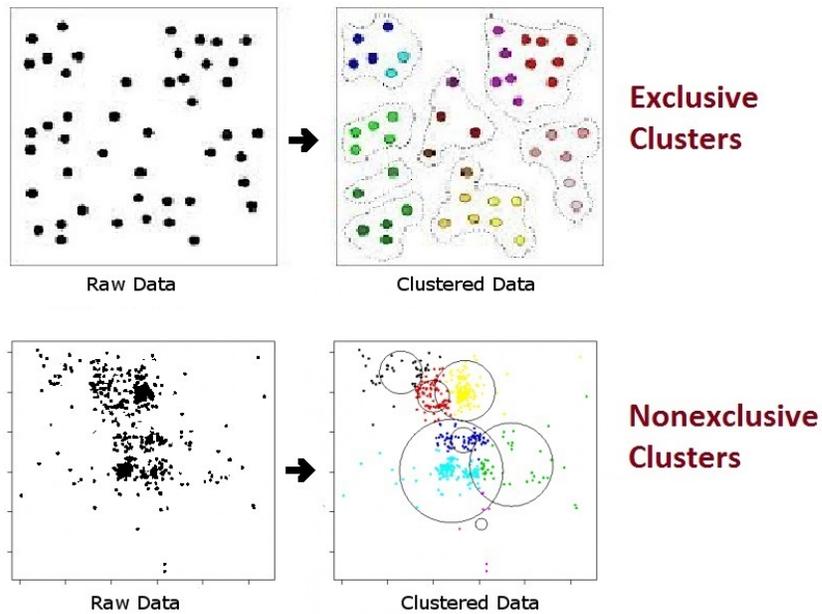


Figure 2.4: Exclusive and Nonexclusive Clustering

fuzzy clustering, an object belongs to every cluster with a membership probability value strictly between 0 and 1. Membership probabilities for every object must sum to 1. Belongingness to each cluster is evaluated based on this membership probability value. In non-fuzzy clustering, an object is either in a cluster or not, and its structure is adaptable for both exclusive or nonexclusive.

### **Intrinsic versus Extrinsic:**

Intrinsic and extrinsic classification both adopt exclusive clustering approach. Intrinsic (or unsupervised) grouping does not use a priori partition of the objects. In intrinsic grouping, we utilize proximity matrix which correspond to the measure for similarity criterion between every pair of objects. For instance, proximity matrix might correspond to the highway distances among a set of cities. Extrinsic (or unsupervised) grouping utilizes category labels on the objects and the proximity matrix. For example, we have data about personal health from smokers and nonsmokers. Intrinsic classification group the people based on the similarities among health level and then try to determine whether smoking should

be considered as a factor for health evaluation. However, extrinsic classification studies ways of discriminating smokers from nonsmokers.

Generally, intrinsic analysis is the main focus since it is the essence of cluster analysis [22]

### **Hierarchical versus Partitional:**

Hierarchical and partitional classification both adopt exclusive and intrinsic clustering approach. In hierarchical classification, the set of clusters are nested and organized as a hierarchical tree. Partitional Clustering corresponds to a division data objects into non-overlapping (ie, exclusive) clusters such that each object is assigned to exactly one subset

### **Heterogeneous versus Homogeneous:**

In heterogeneous clustering approach, the cluster might have different sizes, shapes or densities whereas in the homogeneous case, clusters are required to have identical size.

### **Complete versus Partial:**

In complete classification, we group all the given data whereas under partial clustering, we only cluster certain portion of the data.

Typically, in the network design applications, the specified criterion for cluster formation is based on distance. The objects belong to the same cluster in case they satisfy the closeness criterion according to a given distance measure. However, different closeness criteria exist for the node and cluster distance evaluation. For example, minimum distance to the cluster or minimum average distance etc...

The contribution of this thesis to the survivable telecommunications network design literature is as follows: Survivability issue for a specific fiber optical telecommunications network design problem has been considered. A new 2-node connected fiber optical network design model and two heuristic algorithms are proposed. Generally, the passive optical network problems tend to focus on the design of the access networks structure and consider the backbone and access structure independently, and multi-layer architecture has been widely utilized. Different from the literature, in our mathematical problem, both backbone and access network design have been considered simultaneously and multi-layer hierarchical structure is not enforced. Our first heuristic algorithm, “Ring Creation Algorithm” adopts a greedy perspective and utilizes the benefit of ring formed fiber wires for maintaining the service requirements. The second heuristic algorithm, “Clustering Algorithms”, utilizes distance-based clustering approach in order to form smaller node sets. The clustering algorithm we proposed is an exclusive, intrinsic, partitional, heterogeneous, non-fuzzy and complete clustering algorithm.

In the next chapter, the mathematical model for the “Survivable Green Field Network Design Problem” is presented.

# Chapter 3

## Mathematical Model

In this chapter, the mixed integer programming (MIP) model for the “Survivable Green Field Network Design Problem” (SGFNDP) and some variants of this problem are illustrated. Given a network of demand nodes and a prelocated central station, the task is to design a cost-effective fiber optic network such that there exists 2-node disjoint paths from the central station to every demand node. The demand nodes require certain service criteria about insertion loss and speed level. The access from the central station to the demand nodes can be provided either directly by installing fibers or indirectly by utilizing a special telecommunication equipment called “Passive Splitter” (PS). Passive splitters copy and split the data into the output ports, and there are 4 types of passive splitters with respect to their splitting capacity in this problem setting as chosen in [5]. In our model, the backbone and access network design structures are considered simultaneously. The backbone network consists of the central station and the passive splitters, and it will have a rooted tree structure. Each demand node (or spoke node) will be connected to the backbone network by 2 fiber links.

The network design cost and quality of service are two clashing criteria. Serving the demand nodes with high service quality (ie, low insertion loss values and high speed level values) results in a costly network structure. Likewise, a cost effective networks fails to provide high quality service. A balance should be achieved

through these two clashing factors for the network design.

### 3.1 MIP Model for “Survivable Green Field Network Design Problem”

This section will illustrate the parameters, decision variables and the model for (SGFNPD).

#### Parameters

The parameters are chosen as the same as Yazar’s [5] except minor changes and additions. We are given a set of demand nodes and a central station node which is referred to as *central*. Set  $N$  consists of the demand nodes and *central*. Without loss of generality, we assume that all the demand points are candidate passive splitter locations. There are 4 passive splitter types and  $T$  corresponds to the set of passive splitter types. The passive splitter type specifies the port number and splitting capacity of the corresponding passive splitter. The port number of  $q^{th}$  type passive splitter is  $f_q$ . The splitting capacity (i.e, the maximum number of outputs from a passive splitter) of  $q^{th}$  type is  $2^{f_q}$ . For instance, 3rd type passive splitter will have 3 ports and  $2^3$  splitting capacity. Likewise, 4th type passive splitter will have 4 ports and  $2^4$  splitting capacity. However, central station does not have a splitting capacity.

According to the problem setting, fiber wiring can be applied in the roads including junction points corresponding to corners of streets or entrance of private areas. Hence, highway and street distances are used.  $l_{ij}$  corresponds to the highway distance from node  $i$  to node  $j$  in units of kilometer. This data is symmetric, therefore  $l_{ij} = l_{ji} \quad \forall i, j \in N$ .

$c_{ij}$  is the fiber optical wiring cost between node  $i$  and  $j$ . The fiber wiring cost per kilometer equals to 1, hence  $c_{ij} = l_{ij} \quad \forall i, j \in N$ .

$c_q$  is the cost of  $q^{th}$  type passive splitter  $\forall q \in T$ .  $\alpha$  corresponds to the proportion between the splitter installment and fiber wiring cost and it allows us to generate different cost structures and make comparison. Various  $\alpha$  values are utilized in Yazar’s study. Since passive splitter cost is much lower than fiber wiring cost in real life applications,  $\alpha$  value corresponds to a very small and negligible number. We will consider  $\alpha$  as zero for the computational analysis in order for us to obtain realistic results.

The central station has the signal of the highest quality but the signal will be exposed to the insertion loss. The service quality for insertion is measured in units of dB. The insertion loss of data occurs due to two reasons. The first one is the passive splitter usage and the second one is the traveled distance. *declineps* is the insertion loss in each level of port. It is fixed at 3 dB for each port by the service provider. *declineway* corresponds to the insertion loss per unit distance. It is given as 0.2 dB/km by the company. Due to service quality, there is a limit on the allowed insertion loss. *dBcapacity* is the threshold value loss (ie, the maximum insertion loss allowed). The insertion loss budget was given as 28 dB in “Green Field Network Design Problem” in [5]. We will relax this value by % 25 for SGFNDP. Hence we will choose *dBcapacity* as 35 dB in our model. We assume that a signal starts at the central station with zero dB and each fiber optical wire length traversed augments the dB amount.

Each passive splitter usage causes an insertion loss with respect to its splitting number. For example, if the model uses a type-2 passive splitter, then there will be 2 ports ( $f_2 = 2$ ) and the corresponding insertion loss due the passive splitter of type-2 will be *declineps* \*  $f_2 = 6$  dB.

The insertion loss per distance calculation is more straightforward. Since *declineway* corresponds to the insertion loss per unit distance, the increment insertion loss due the distance from node i to j is *declineps* \*  $l_{ij}$ .

The central station has the fastest speed, it is specified with parameter *mbcentral*. It is given as 2.5 Gb/sec. Speed decreases only due to splitting of passive splitters. More splitting causes more reduction in the speed. However, direct connections

from central station do not cause any speed reduction. Due to service quality requirement, there is a threshold value for speed,  $mbthreshold$ . The service quality for speed level is measured in units of Mb/sec and it was given as 100 Mb/sec in [5]. We will relax this value by % 25 for SGFNDP. Hence we will choose  $mbthreshold$  as 75 dB for our model.

The reason we relax the service requirement threshold limits (ie,  $dBcapacity$  and  $mbthreshold$ ) is that attaining survivability via 2-node disjoint paths converts the problem into a more restricted one and we wish to obtain a balance by relaxing the service quality limitations.

$M$  is a sufficiently big number for the insertion loss and speed values.

### Decision Variables

$W_{ij}$  is the decision variable for keeping track of the installed fiber wire links. If link  $\{i, j\}$  is established, arc (i,j) and arc (j,i) can utilize that link. Another important property is that the model only allows the outgoing links from a passive splitter or *central*. Hence link  $\{i, j\}$  can only be installed in case node i is a passive splitter or *central*. The reason behind this modeling approach is to be able to limit the number of outputs from every established passive splitter since we have to consider the splitting capacity issue.

$y_{jq}$  allows us to find out which type of passive splitter has been established at the nodes.

$x_{ij}^k$  and  $z_{ij}^k$  are the decision variables for keeping track of the used arcs for accessing to node k on primary path and secondary paths, respectively.

$P1_j^k$  and  $P2_j^k$  correspond the dB amount at node j to access node k under the usage of primary path and secondary paths, respectively.

$M1_j^k$  and  $M2_j^k$  indicate the speed value at node j to access node k under the usage of primary path and secondary paths, respectively.

## Parameters

$N$  : the demand node set on a network and the central station which is referred to as *central*

$K$  : the demand node set on a network (ie,  $K = N \setminus \text{central}$ )

$T$  : the set of passive splitters' types.

$L = [l_{ij}]$  : the highway distance from node  $i$  to node  $j$   $\forall i, j \in N$

$C = [c_{ij}]$  : the cost of installing a fiber link from node  $i$  to node  $j$   $\forall i, j \in N$

$c_q$  : the cost of  $q^{th}$  type passive splitter  $\forall q \in T$

$\alpha$  : proportion between the splitter and fiber wiring cost. (This proportion allows us to make a comparison for different cost structures of the fiber optical wiring and passive splitters)

$f_q$  : the port number in the splitter of type  $q$   $\forall q \in T$

$2^{f_q}$  will be the number of maximum outputs from a passive splitter of type  $q$

*declineps* : the insertion loss in each level of port. It is fixed at 3 dB for each port by the service provider

*declineway* : the insertion loss per km. It is given as 0.2 dB/km by the service provider

*dBcapacity* : Threshold value loss (ie, the maximum insertion loss allowed). The insertion loss budget is taken as 35 dB.

*mbcentral* : the out power of the central station. It is specified as 2.5 Gb/sec in this application

*mbthreshold* : the speed threshold level for each demand node. It is chosen as 75 Mb/sec

$M$ : a very big number

### Decision Variables

$$W_{ij} = \begin{cases} 1 & \text{if a fiber link } \{i, j\} \text{ is installed} \\ 0 & \text{otherwise} \end{cases} \quad \forall i, j \in N, i \neq j$$

$$y_{jq} = \begin{cases} 1 & \text{if a passive splitter of } q^{th} \text{ type is located at node } j \\ 0 & \text{otherwise} \end{cases} \quad \forall j \in N, \forall q \in T$$

$$x_{ij}^k = \begin{cases} 1 & \text{if arc } (i, j) \text{ is used for the primary path to access node } k \\ 0 & \text{otherwise} \end{cases} \quad \forall i, j \in N, k \in K$$

$$z_{ij}^k = \begin{cases} 1 & \text{if arc } (i, j) \text{ is used for the secondary path to access node } k \\ 0 & \text{otherwise} \end{cases} \quad \forall i, j \in N, k \in K$$

$P1_j^k$ : the dB amount at node  $j$  under the usage of primary path to access node  $k$ .  
 $\forall j \in N, k \in K$

$P2_j^k$ : the dB amount at node  $j$  under the usage of secondary path to access node  $k$ .  
 $\forall j \in N, k \in K$

$M1_j^k$ : speed value at node  $j$  under the usage of primary path to access node  $k$ .  
 $\forall j \in N, k \in K$

$M2_j^k$ : speed value at node  $j$  under the usage of secondary path to access node  $k$ .  
 $\forall j \in N, k \in K$

### Illustration of Insertion Loss Increment and Speed Level Reduction

Insertion loss increment and speed reduction in the network can be illustrated as follows. Consider that we have a primary path that utilizes arc  $(i, j)$  for accessing node  $k$  as in Figure 3.1. Hence  $x_{ij}^k = 1$ . Assume that node  $i$  is a type- $q$  passive

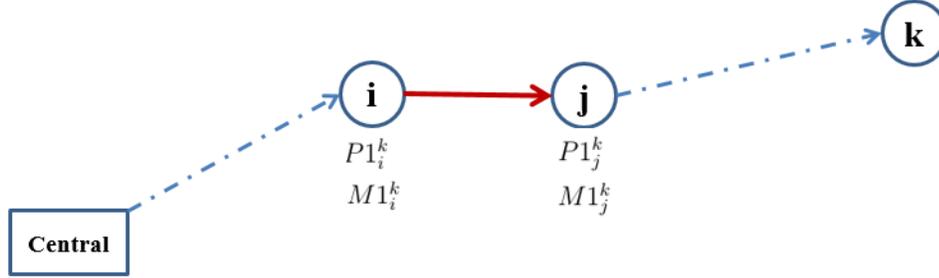


Figure 3.1: Example for Insertion Loss and Speed Level

splitter and node j is a type-r passive splitter.

The insertion loss value for node j increases along the path due to two factors: (1) distance between i and j ( $l_{ij}$ ) and (2) passive splitter assignment of node j. As stated before,  $P1_i^k$  and  $P1_j^k$  are the insertion loss values of node i and node j, respectively on the primary path for accessing node k. Then,  $P1_j^k$  value in Figure (3.1) is calculated as follows:

$$P1_j^k = P1_i^k + \text{declineway} * l_{ij} + \text{declineps} * f_r$$

If node j were not a passive splitter, then insertion value increment would not occur due to passive splitter assignment. In that case, we would only face insertion loss because of the distance.

The speed level reduction for node j occurs due to the output number of node i. As stated before,  $M1_i^k$  and  $M1_j^k$  is the speed level of node i and node j respectively on the primary path for accessing node k. Then,  $M1_j^k$  value in Figure (3.1) is calculated as follows:

$$M1_j^k = \frac{M1_i^k}{2f_q}$$

The same insertion loss and speed level attribute is valid for the secondary path

and the corresponding values as well.

## Mathematical Model

The MIP model for “Survivable Green Field Network Design” is as follows:

$$\max \sum_{i \in N} \sum_{j \in N} c_{ij} * W_{ij} + \sum_{i \in N} \sum_{q \in T} \alpha * c_q * y_{iq} \quad (3.1)$$

s.t.

$$\sum_{j \in N \setminus i} W_{ij} \leq \sum_{q \in T} 2^{f_q} * y_{iq} \quad \forall i \in K \quad (3.2)$$

$$x_{ij}^k + x_{ji}^k + z_{ij}^k + z_{ji}^k \leq W_{ij} + W_{ji} \quad \forall i, j \in N, k \in K, i \leq j \quad (3.3)$$

$$x_{ij}^k \leq \sum_{q \in T} y_{jq} \quad \forall i, j \in N, k \in K, j \neq k \quad (3.4)$$

$$z_{ij}^k \leq \sum_{q \in T} y_{jq} \quad \forall i, j \in N, k \in K, j \neq k \quad (3.5)$$

$$\sum_{q \in T} y_{jq} \leq 1 \quad \forall j \in N \quad (3.6)$$

$$\sum_{j \in N} x_{ij}^k + \sum_{j \in N} z_{ij}^k \leq 1 \quad \forall i, k \in K \quad (3.7)$$

$$x_{central,i}^k + z_{central,i}^k \leq 1 \quad \forall i \in N, k \in K \quad (3.8)$$

$$W_{ij} + W_{ji} \leq 1 \quad \forall i, j \in N, i \leq j \quad (3.9)$$

$$\sum_{j \in N, j \neq i} x_{ij}^k - \sum_{j \in N, j \neq i} x_{ji}^k = \begin{cases} -1 & \text{if } i = k \\ 1 & \text{if } i = \text{central} \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N, k \in K \quad (3.10)$$

$$\sum_{j \in N, j \neq i} z_{ij}^k - \sum_{j \in N, j \neq i} z_{ji}^k = \begin{cases} -1 & \text{if } i = k \\ 1 & \text{if } i = \text{central} \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N, k \in K \quad (3.11)$$

$$P1_{central}^k = 0 \quad \forall k \in K \quad (3.12)$$

$$P1_j^k \geq P1_i^k + \text{declineway} * l_{ij} * x_{ij}^k - M * (1 - x_{ij}^k) + \sum_{q \in T} \text{declineps} * f_q * y_{jq}$$

$$\forall i \in N, j, k \in K \quad (3.13)$$

$$P1_j^k \leq P1_i^k + \text{declineway} * l_{ij} * x_{ij}^k + M * (1 - x_{ij}^k) + \sum_{q \in T} \text{declineps} * f_q * y_{jq}$$

$$\forall i \in N, j, k \in K \quad (3.14)$$

$$P1_j^k \leq \text{dBcapacity} * \sum_{i \in N} x_{ij}^k \quad \forall j \in N, k \in K \quad (3.15)$$

$$P2_{\text{central}}^k = 0 \quad \forall k \in K \quad (3.16)$$

$$P2_j^k \geq P2_i^k + \text{declineway} * l_{ij} * z_{ij}^k - M * (1 - z_{ij}^k) + \sum_{q \in T} \text{declineps} * f_q * y_{jq}$$

$$\forall i \in N, j, k \in K \quad (3.17)$$

$$P2_j^k \leq P2_i^k + \text{declineway} * l_{ij} * z_{ij}^k + M * (1 - z_{ij}^k) + \sum_{q \in T} \text{declineps} * f_q * y_{jq}$$

$$\forall i \in N, j, k \in K \quad (3.18)$$

$$P2_j^k \leq \text{dBcapacity} * \sum_{i \in N} z_{ij}^k \quad \forall j \in N, k \in K \quad (3.19)$$

$$M1_{\text{central}}^k = \text{mbcentral} \quad \forall k \in K \quad (3.20)$$

$$M1_j^k \geq \text{mbcentral} * x_{\text{central},j}^k \quad \forall j \in N, k \in K \quad (3.21)$$

$$M1_j^k \leq \frac{M1_i^k}{2f_q} + \text{mbcentral}(1 - y_{iq}) + \text{mbcentral}(1 - x_{ij}^k)$$

$$\forall i, j, k \in K, q \in T \quad (3.22)$$

$$M1_j^k \geq \frac{M1_i^k}{2f_q} - M * (1 - y_{iq}) - M * (1 - x_{ij}^k)$$

$$\forall i, j, k \in K, q \in T \quad (3.23)$$

$$M1_j^k \geq mbcentral * (1 - \sum_{i \in N} x_{ij}^k) + mbthreshold * \sum_{i \in N} x_{ij}^k \quad \forall j \in N, k \in K \quad (3.24)$$

$$M1_j^k \leq mbcentral \quad \forall j \in N, k \in K \quad (3.25)$$

$$M2_{central}^k = mbcentral \quad \forall k \in K \quad (3.26)$$

$$M2_j^k \geq mbcentral * z_{central,j}^k \quad \forall j \in N, k \in K \quad (3.27)$$

$$M2_j^k \leq \frac{M2_i^k}{2f_q} + mbcentral(1 - y_{iq}) + mbcentral(1 - z_{ij}^k) \quad \forall i, j, k \in K, q \in T \quad (3.28)$$

$$M2_j^k \geq \frac{M2_i^k}{2f_q} - M * (1 - y_{iq}) - M * (1 - z_{ij}^k) \quad \forall i, j, k \in K, q \in T \quad (3.29)$$

$$M2_j^k \geq mbcentral * (1 - \sum_{i \in N} z_{ij}^k) + mbthreshold * \sum_{i \in N} z_{ij}^k \quad \forall j \in N, k \in K \quad (3.30)$$

$$M2_j^k \leq mbcentral \quad \forall j \in N, k \in K \quad (3.31)$$

$$W_{ij}, y_{iq}, x_{ij}^k, z_{ij}^k \in \{0, 1\} \quad \forall i, j \in N, \forall k \in K, \forall q \in T \quad (3.32)$$

$$P1_j^k, P2_j^k, M1_j^k, M2_j^k \geq 0 \quad \forall j \in N, k \in K \quad (3.33)$$

The objective function (3.1) minimizes the total cost of fiber optical wiring and passive splitter installment. Since we consider  $\alpha$  as zero, the PS installment cost disappears. Hence, we only minimize the fiber wiring cost.

Each passive splitter has a splitting capacity depending on its type. The total number of outputs (ie, outgoing links) from a passive splitter of  $q^{th}$  type cannot exceed its splitting capacity,  $2^{f_q}$ . Since *central* does not have a splitting capacity, we exclude it for this constraint. Constraint (3.2) ensures that splitting capacity is not exceeded. It also guarantees that in order to have an outgoing link from any node, a passive splitter has to be located in there.

Constraint (3.3) ensures that the paths can use arc (i,j) only if the fiber link  $\{i, j\}$  is established. It also guarantees that primary and secondary paths will not use the same links. Hence edge disjointness property among primary and secondary paths is ensured. This constraint considers the indexes  $i \leq j$  in order not to write the same constraint again.

We should utilize a path of passive splitters in order to access the demand nodes. Hence, the intermediate nodes on the path must be passive splitters. Constraint (3.4) states that if node j is not the access node (node k) on the path, then node j can only be used on the primary path in case it is a passive splitter. Constraint (3.5) ensures the same restriction for the secondary path.

Constraint (3.6) guarantees that a node can be assigned to at most one type of passive splitter.

Primary and secondary paths cannot use the same intermediate nodes for accessing a demand node. For instance, if the primary path for accessing node k is *central*-  $i_4$  -  $i_7$  -  $k$ , then the secondary path for accessing node  $k$  cannot utilize nodes  $i_4$  and  $i_7$ . The node disjointness property is ensured in Constraint (3.7) and (3.8).

Constraint (3.9) states that there should be at most a single link between two nodes.

Constraints (3.10) and (3.11) are the flow balance equations for the primary and secondary paths respectively.

Constraint (3.12) indicates the central station decibel value characteristics for

primary path usage. In order to evaluate the insertion loss amount and assess the decibel value, Constraints (3.13) and (3.14) are utilized. If node  $j$  is not used on the primary path to access node  $k$ , then its corresponding variable,  $P1_j^k$  will be the same as *central*'s. Hence it will be 0. If node  $j$  is used on the primary path to access node  $k$ , then its dB value increases according to distance and passive splitter usage as explained before. Constraint (3.15) ensures that every node under the usage of primary path is in the certain dB limit.

Constraints (3.16)-(3.19) are analogous to Constraints (3.12)-(3.15). The only difference is the secondary path utilization.

Constraint(3.20) indicates the central station speed value characteristics for primary path usage. Direct connections from central station do not cause speed reduction. This property is ensured by Constraint (3.21). Constraints (3.22)-(3.23) are used for evaluating the speed amount of the nodes on the primary path. If node  $j$  is not used on the primary path to access node  $k$ , then its corresponding variable,  $M1_j^k$  will be the same as *central*'s. Hence it will be *mbcentral*. If node  $j$  is used on the primary path to access node  $k$ , then its speed value decreases according to the passive splitter's type as explained before. Constraint (3.24) ensures that every node under the usage of primary path should satisfy the speed threshold requirement. Constraint (3.24) and (3.25) provide that the nodes that are not utilized for a certain path will have a speed value *mbcentral*.

Constraints (3.26)-(3.31) are analogous to Constraints (3.20)-(3.25). Secondary path consideration is the only difference.

The domains of the decision variables are specified in Constraints (3.32)-(3.33).

## 3.2 The variants of the problem

In this section, the variants of the (SGFNDP) problem definition and the corresponding changes will be examined.

## 2-Edge Disjoint Paths

If our problem were to seek 2-edge disjoint paths instead of 2-node disjoint path, then we would have to delete the Constraints (3.9) and (3.10) We do not need to add any constraint since edge disjointness property among paths is already provided in the model by Constraint (3.3)

## m-Node Disjoint Path Consideration

We may want to construct a network where m node-disjoint paths should exist from the central station to every demand node. In this case, we define additional decision variables and parameters that are analogous to  $x_{ij}^k(z_{ij}^k)$ ,  $P1_j^k(P2_j^k)$  and  $M1_j^k(M2_j^k)$  for the primary path (secondary path). The model can be extendable to m-Node Disjoint Paths by adding the corresponding constraints.

## Differentiating Demand Node Set based on Quality of Service Level

It is a common approach to consider that some demand nodes are more important than the remaining nodes. Hence, a higher quality of service will be required for these nodes. Let us differentiate the demand node set  $K$  as VIP customers  $K1$  and standard customers  $K2$ ,  $K = K1 \cup K2$ . We also differentiate parameter for the speed threshold level accordingly.

Additional parameters:

$K1$  : demand node set for VIP customers

$K2$  : demand node set for standard customers.  $K = K1 \cup K2$

$mbthreshold1$  : speed threshold for each node for VIP customers

$mbthreshold2$  : speed threshold for each node for standard customers.

$mbthreshold1 \geq mbthreshold2$

The modification of the constraints:

For the MIP model, Constraints (3.26)-(3.31) indicate the service requirement about the speed level. Since the demand node set differs, the constraint (3.34)-(3.37) should be considered for both node sets,  $K1$  and  $K2$ . Therefore, Constraint (3.26)-(3.31) should be deleted and Constraints (3.34)-(3.37) should be added to the model instead.

$$M1_j^k \geq mbcentral * (1 - \sum_{i \in N} x_{ij}^k) + mbthreshold1 * \sum_{i \in N} x_{ij}^k \quad \forall j \in K1, k \in K \quad (3.34)$$

$$M1_j^k \geq mbcentral * (1 - \sum_{i \in N} x_{ij}^k) + mbthreshold2 * \sum_{i \in N} x_{ij}^k \quad \forall j \in K2, k \in K \quad (3.35)$$

$$M2_j^k \geq mbcentral * (1 - \sum_{i \in N} z_{ij}^k) + mbthreshold * \sum_{i \in N} z_{ij}^k \quad \forall j \in K1, k \in K \quad (3.36)$$

$$M2_j^k \geq mbcentral * (1 - \sum_{i \in N} z_{ij}^k) + mbthreshold * \sum_{i \in N} z_{ij}^k \quad \forall j \in K2, k \in K \quad (3.37)$$

# Chapter 4

## Heuristic Algorithms

As the problem size increases, obtaining an optimal solution becomes very difficult and takes a lot of time. In order to get feasible solutions in a fast manner, we have developed two constructive heuristic algorithms for SGFNDP.

The first heuristic is called the “Ring Creation Algorithm”. When the fiber links are installed in a ring form, we provide 2 node-disjoint paths for every element of the ring. Using this feature, the algorithm seeks to install fiber wires in a ring structure such that every demand node is an element of a generated ring. While creating a ring structure, new addition to the ring is possible as long as service quality requirements are met for every element of the ring. After the ring generation phase, we may end up with rings with cardinality 1 and this causes problem since there should exist at least 2 node-disjoint paths for every demand node. Special consideration is provided for this case.

The second heuristic algorithm is the “Clustering Algorithm” which adopts a Divide-and-conquer approach. We break down SGFNDP into sub-problems where these sub-problems can be solved easily in a fast manner. For the break-down process, this algorithm divides the demand nodes into different clusters based on closeness criteria. Each cluster’s cardinality has a lower and upper bound. After the grouping operation, we independently solve each sub-problem which is based

on a clustered node set. The separate solutions will be combined to give a solution for the original problem. As mentioned in Chapter 2, our clustering algorithm is exclusive, intrinsic, partitional, heterogeneous, non-fuzzy and complete.

As stated in the previous chapter, Passive Splitter (PS) installment cost is negligible compared to fiber wiring cost. We consider  $\alpha = 0$ , hence we do not take the PS installment cost into consideration at all for the heuristics algorithms.

## 4.1 Ring Creation Algorithm

In this algorithm, we seek to install fiber wires as a ring structure such that every demand node is an element of a generated ring. When the fiber links are constructed in a ring form, we provide 2 node-disjoint paths for every element of the ring. The start and end point of every generated ring is *central* since we are required to provide paths from *central*. For illustration, let us consider Ring R0 and Ring R1 in Figure 4.1. As we can see, these rings both start from and end at *central*. R0 consists of nodes i, m, j and k where i is the first inserted element and m is the second inserted element. Likewise, Ring R1 is composed of nodes t, h and u. Due to the feature of ring structure, each element of these rings is served via 2-node disjoint paths from *central*. The primary and secondary paths of the nodes are also given in Figure 4.1.

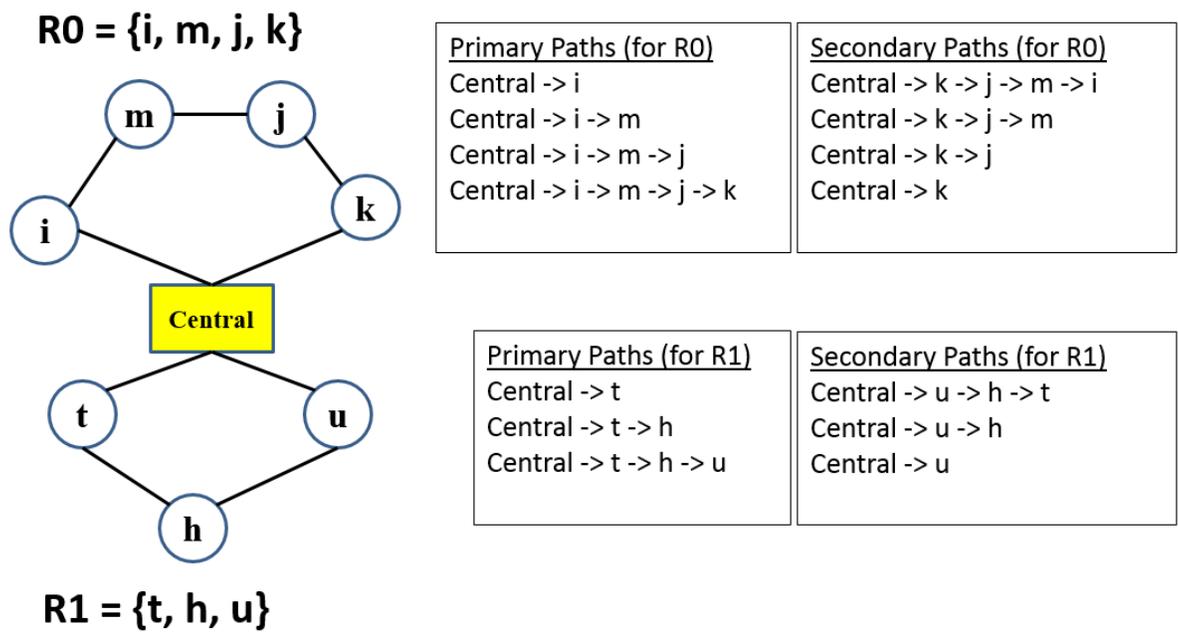


Figure 4.1: Ring Structure and 2 Node-Disjoint Paths

Initially, we assign each demand node as type-1 PS since type-1 PS nodes face the lowest quality of service degradation (ie, insertion loss increment and speed reduction amount is smaller for type-1 PS) compared to the other PS types. Then, we sort the demand nodes into set E by utilizing Nearest Neighbor approach. Set E consists of all demand node elements but the order is based on a closeness criterion. The first element of E is the closest node to *central*, and the second element of E is the closest node to the first element. The remaining elements are organized in a similar manner. An illustration is provided in Figure 4.2. Node 3 is the closest node to *central*, hence it is the first element of set E. Likewise, node 1 is the closest node to node 3 among the remaining node set, hence it is the second element of set E. Set E consists of distinct nodes since a node cannot be added twice.

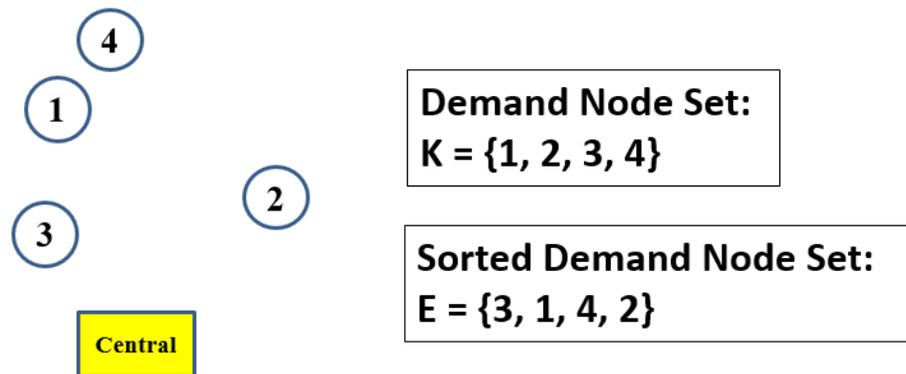


Figure 4.2: Set E generation

The ring creation and node insertion occur based on the order of elements in E. Starting from the first element of E, we keep inserting the nodes to the ring structure. Ring structure we proposed assures the existence of 2 node-disjoint paths from *central*. However, we have to take feasibility into account as well. Therefore, we should make sure that the elements of the generated rings obey the insertion loss limit and speed level threshold (ie, *dBcapacity* and *mbthreshold*). These feasibility requirements are considered during the node addition phase. The algorithm ensures that a new node insertion to the current ring is possible as long as service quality requirements are met for every node in this ring.

Let us assume that  $E = \{i, m, j, k, t, h, u\}$ . The algorithm starts by creating a ring R0 and adding the first element of E (ie, node *i*) to that ring. We do not need to check the feasibility requirements for the first insertion. Because it is safe to assume that direct connection from *central* to any demand node do not cause infeasibility. We delete node *i* from set E. Then, we scan the elements of E in an orderly fashion for further node insertion to the ring R0. Hence, we examine whether we can add node *m* to R0 without causing infeasibility. If *m* does not endanger feasibility, then it is inserted to R0. If node *m* is added to the ring, then we examine whether we can further insert node *j* to the ring (since node *j* is the next element to be scanned). We keep scanning the elements of E in this orderly manner until we reach a node that endangers feasibility. In this case, that node is not inserted to the ring and we complete the ring R0. We delete the elements of R0 from set E. If E is nonempty, we create a new ring R1 and apply the same procedure. We keep creating rings in this fashion until set E is empty. Hence, this algorithm allocates every demand node to a ring.

### **Service Quality Value Features in a Ring Structure**

For feasibility evaluation, we need to calculate the service quality values (ie,  $P1, P2, M1$  and  $M2$  values) of the ring elements. Due to the ring structure feature, we have use the same arc for accessing a node element based on its insertion order. Let us consider R0 in Figure 4.1.  $R0 = \{i, m, j, k\}$  where node *i* is the first inserted element and node *k* is the last inserted element. We use arc (*central*, *i*) for the primary path of nodes *i*, *m*, *j*, *k* and we utilize arc (*i*,*m*) for

the primary paths of nodes  $m, j, k$ . The utilization of the primary path arcs is based on the order of  $R0$ . Likewise, we utilize arc (*central*,  $k$ ) for the secondary path of nodes  $k, j, m, i$ . The utilization of the secondary path arcs is based on the reverse order of  $R0$ . Furthermore, we have the same PS type for every ring element, hence the insertion loss due to the PS usage will be the same. Therefore, we have the following relationship for  $R0$  elements in Figure 4.1:

$$\begin{aligned}
P1_i^i &= P1_i^m = P1_i^j = P1_i^k \\
P1_m^m &= P1_m^j = P1_m^k \\
P1_j^j &= P1_j^k \\
P2_k^k &= P2_k^j = P2_k^m = P2_k^i \\
P2_j^j &= P2_j^m = P2_j^i \\
P2_m^m &= P2_m^i
\end{aligned}$$

Due to the same rationality about arc usage in a ring structure, the relationship for speed value is as follows:

$$\begin{aligned}
M1_i^i &= M1_i^m = M1_i^j = M1_i^k \\
M1_m^m &= M1_m^j = M1_m^k \\
M1_j^j &= M1_j^k \\
M2_k^k &= M2_k^j = M2_k^m = M2_k^i \\
M2_j^j &= M2_j^m = M2_j^i \\
M2_m^m &= M2_m^i
\end{aligned}$$

As mentioned in the previous chapter, the speed level value is only affected by the PS type. Since every ring element has the same PS type, the speed value of a ring element on a path is proportional to the number of nodes traversed at the corresponding path. We see that this number is the same for the first inserted and last inserted node on different paths. The similar rule applies between the second and penultimate node of  $R0$ . Hence, we have the following relationship for the elements of  $R0$ :

$$\begin{aligned}
M1_i^i &= M2_k^k \\
M1_m^m &= M2_j^j \\
M1_k^k &= M2_i^i \\
M1_j^j &= M2_m^m
\end{aligned}$$

### The Illustration of the Node Insertion Phase

Let us assume that current ring  $R0 = \{i, m, j\}$  and we check whether we can add node  $k$  to  $R0$ . Node  $k$  can be inserted to  $R0$  only if the feasibility is still maintained after the insertion of that node. Hence we have to check whether the service quality values (ie,  $P1, P2, M1$  and  $M2$  values) of the elements of  $R0$  obey the corresponding limits (ie,  $dBcapacity, mbthreshold$ ) in case we insert node  $k$  to  $R0$ . For the service quality value evaluation, we find the corresponding values of node  $k$  as if we have already inserted that node to the current ring. Hence, the last inserted element of the ring that we refer corresponds to node  $k$ .

We should note that it is sufficient to check the service quality values of the first inserted and last inserted element of the ring in order to assess the feasibility statues of the overall ring. The intermediate nodes' values ( $P1, P2, M1$  and  $M2$ ) need not to be checked since they will always be between the values of the first inserted and last inserted element of the ring. Therefore, we only check the service quality values of first inserted and last inserted nodes (ie, the first element of  $R0$  and node  $k$ ).

We want to calculate the service quality values of node  $k$  in order to examine whether node  $k$  can be added to  $R0$ . We know the values of  $P1_j^j$  and  $M1_j^j$  since node  $j$  is already in  $R0$ . Due to ring structure feature, we also know that  $P1_k^k = P1_j^j$  and  $M1_j^j = M1_k^k$  as explained above. As illustrated in Figure 4.3, we can calculate  $P1_k^k$  and  $M1_k^k$  as follows:

$$\begin{aligned}
P1_k^k &= P1_j^j + declineway * l_{kj} + declineps * f_1 \\
M1_k^k &= \frac{M1_j^j}{2^{f_1}}
\end{aligned}$$

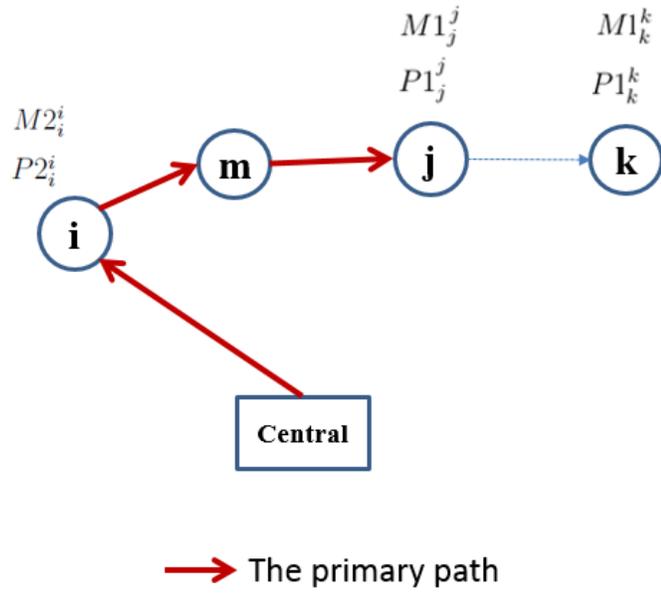
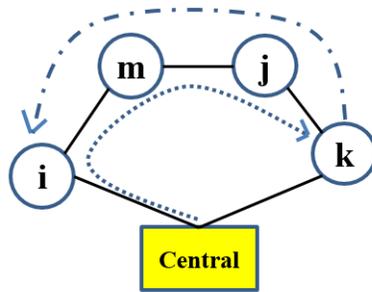


Figure 4.3: Ring Creation Algorithm- Expanding the Ring

As explained above, we also need to calculate  $P2_i^i$  (the secondary path insertion loss value of the first element of the ring) and  $M2_i^i$  (the secondary path speed level value of the first element of the ring). The only difference between  $P1_k^k$  and  $P2_i^i$  occurs due to different values of  $l_{central,i}$  and  $l_{central,k}$  as shown in Figure 4.4. Since every demand node is of type-1 PS and the same number of PS is utilized, no difference occurs due to PS usage. Hence,  $P2_i^i$  is calculated as follows:

$$P2_i^i = P1_k^k + declineway * (l_{central,k} - l_{central,i})$$

**RO = {i, m, j, k}**



Primary Path of Node k .....  
 Central -> i -> m -> j -> k

Secondary Path of Node i - . - . - .  
 Central -> k -> j -> m -> i

Figure 4.4: Ring Creation Algorithm- Paths of Node i and Node k

$M2_i^i = M1_k^k$  due to ring structure feature about the speed levels as explained before. Every ring element belongs to the same type of PS (type 1) and the same number of PS is utilized for these paths as can be seen in Figure 4.4. Hence we do not need to check  $M2_i^i$  value since  $M1_k^k$  is already considered for feasibility check.

To sum up, if  $P1_k^k$ ,  $M1_k^k$  and  $P2_i^i$  obey the corresponding bounds (ie,  $dBcapacity$  and  $mbthreshold$ ), then node  $k$  is added to the ring  $R0$  and the iterative step continues (ie, the next node in  $E$  is scanned for node insertion to  $R0$ ). Else, we complete Ring  $R0$ . We set the  $P1_k^k$ ,  $M1_k^k$  and  $P2_i^i$  values to the initial values. We assign the remaining service quality values (ie,  $P1_i^i$ ,  $M1_m^j$ ,  $P2_k^j$ ,  $M2_m^i$  etc...) of the ring element based on the ring structure feature explained before. We delete the elements of  $R0$  from set  $E$  and we create a new ring and apply the same procedure based on updated  $E$ . We keep going in this manner until  $E$  is empty (ie, every demand node is allocated to a ring structure). The created rings of this algorithm are inserted to set  $RING$ .

### **Outcast Node Service Requirement (The Issue of Rings with cardinality 1)**

Problem setting states that we have to serve the demand nodes via 2 node-disjoint paths. In order to meet 2-connectivity requirement, the cardinality of each ring should be at least 2. However, after the ring generation phase, we may face a situation where there are rings that have a cardinality 1 (insufficient ring size). Under this situation, the elements of the rings of cardinality 1 are not served via 2 paths. Hence, we need to handle this issue by performing an update operation.

Let us call the elements of the rings with cardinality 1 as “outcast nodes”. We delete the rings with cardinality 1 from set  $RING$ . Now, we make sure that every ring in set  $RING$  is a proper ring. In order to serve the outcast nodes properly (ie, via 2 node-disjoint path), we assign them as nonPS (non-passive splitter) nodes.

For every outcast node, we find a suitable PS node in order to construct its first

path. The eligible PS nodes for the outcast node are chosen as the first and last inserted elements of the ring since they have advantageous values based on the corresponding path. The first inserted element of a ring has better service quality values (ie, low insertion loss value and high speed level) based on primary path. Likewise, the last inserted element of a ring is advantageous based on secondary path. We assign the closest PS to the outcast node and we serve via the advantageous path. Hence, we make sure that nonPS nodes are served with high service quality.

We should note that a PS node can be assigned to at most one nonPS node since every PS node is of type-1 (ie, due to splitting capacity issue).

The second path of the outcast node will be constructed via direct connection from *central*.

This way, we make sure that nonPS nodes are served with high service quality and they will not cause any infeasibility for the algorithm.

The technical illustration of the Outcast Node Service is as follows:

Let us assume that, after the ring generation phase, we generated 2 proper rings (Ring R0 and R1) and we have only one outcast node  $h$  as illustrated Figure 4.5.

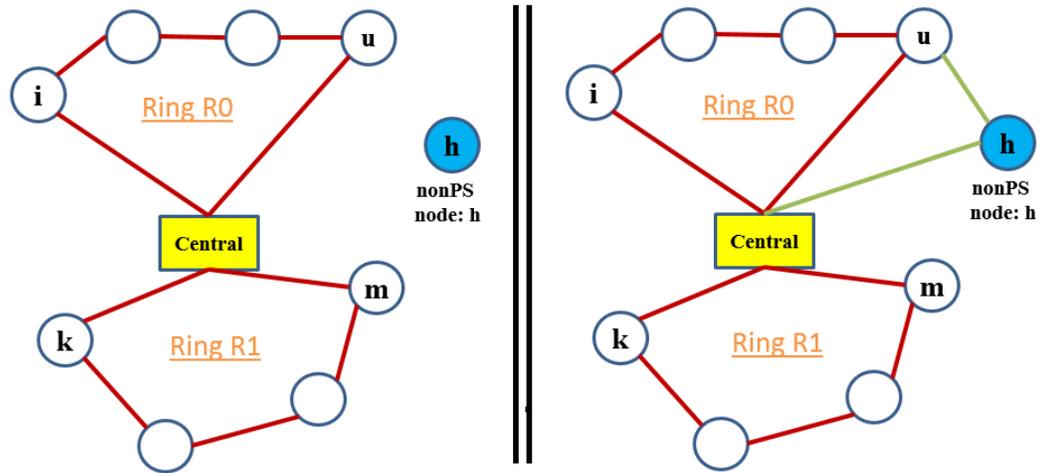


Figure 4.5: Ring Algorithm - Serving nonPS node  $h$

We assign node  $h$  as a nonPS node and insert  $h$  to set  $V$  (ie,  $V$  consists of nonPS nodes).  $V = \{h\}$ . In order to construct the first path of the nonPS node  $h$ , we aim to choose a suitable PS node. We create a set  $EligNPS$  that consists of the first inserted or last inserted element of each proper ring (ie, ring with cardinality at least 2 or equivalently that ring is in set  $RING$ ). The first element of each proper ring belongs to set  $FirstPS$  and last elements of the proper rings form  $LastPS$ . Hence, we have  $FirstPS \cup LastPS = EligNPS$ .

The first elements of  $R0$  and  $R1$  are  $i$  and  $k$ . Hence set  $FirstPS = i, k$ . Likewise,  $LastPS = u, m$  since  $u$  and  $m$  are the last elements of the rings. The eligible set for nonPS node  $h$  is  $EligNPS = FirstPS \cup LastPS = i, k, u, m$ . The closest node to  $h$  among set  $EligNPS$  is node  $u$ . We install fiber link between  $h$  and  $u$  and between  $central$  and  $h$ . Since  $u$  belongs to set  $LastPS$ , we will consider the (advantageous) secondary path values of  $u$  for accessing nonPS  $h$  from  $central$ .

Hence the first path of  $h$  will be constructed as follows:

$$\begin{aligned} z_{central,u}^h &= 1 \\ z_{u,h}^h &= 1 \end{aligned}$$

The insertion loss and speed values are calculated according to the secondary path values of  $u$ . We know that  $P2_h^h = P2_u^u$  and  $M2_h^h = M2_u^u$  since node  $u$  is utilized on path to access node  $h$ . Hence, we have:

$$\begin{aligned} P2_h^h &= P2_u^u + \text{declineway} * l_{uh} \\ M2_h^h &= \frac{M2_u^u}{2^{f_1}} \end{aligned}$$

For the remaining path, we install a fiber wire directly from  $central$  to the nonPS nodes. The second path of  $h$  and service quality value are:

$$\begin{aligned} x_{central,h}^h &= 1 \\ P1_h^h &= \text{declineway} * l_{central,h} \\ M1_h^h &= mb_{central} \end{aligned}$$

After this process, we assign  $P2_u^h$  value, we delete  $u$  from set  $EligNPS$  and  $LastPS$ .

We also delete node  $h$  from  $V$  since  $h$  is served properly in this setting. We should perform this operation for every outcast node.

After we ensure 2-node connectivity for every outcast node, we calculate the objective function value of this algorithm based on installed fiber wires.

### Ring Creation Algorithm - Example

We provide an example of this algorithm for illustration. Consider that we are given demand node set  $K$  (14 demand nodes), and *central* at node 32 (yellow node) as in Figure 4.6. Initially, eligible set  $E$  consists of demand nodes where the nodes are sorted based on the closeness criteria to the *central*.  $E = \{40, 19, 27, 41, 23, 29, 26, 38, 31, 37, 15, 13, 44, 2\}$ . The first element of  $E$  (ie, the closest node to the central) is node 40. We add this node to the ring. We delete this node 40 from  $E$ . Now, the first element of  $E$  is node 19 and we examine whether we can perform node insertion. Since the feasibility is maintained, we add node 19 and we scan the second element of  $E$  (node 27) for node insertion. We keep going in this manner iteratively. The resulting ring  $R_0$  consists of nodes 40, 19, 27, 41, 23 and 29. No more addition (ie, addition of node 26 to  $R_0$ ) to this ring became possible due to feasibility issue (ie, the service quality values of ring elements). We delete this ring's elements from Eligible Set  $E$  and we begin a new ring in the same manner. After the deletion, the first element of  $E$  is node 26. Hence it is the first inserted element of the  $R_1$ . Then we delete node 26 from  $E$ . Now the first element of  $E$  is 38 and we check the feasibility situation for node 38. We keep going in this manner. The given network and the resulting output are illustrated in Figure 4.6.

The output of this algorithm is as follows:

$R_0$  (demand nodes): 40, 19, 27, 41, 23, 29

$R_1$  (demand nodes): 26, 38, 31, 37, 15, 13

$R_2$  (demand nodes): 44, 2

Since the cardinality of each ring is larger than 1, we do not perform an update operation for the outcast nodes. We should note that each ring starts and ends from *central* as mentioned before, therefore node 32 (*central*) is an element of every ring by default.

According to the generated rings, we install the fiber links as follows: For R0, we install these links: 32-40 , 40-19, 19-27,27-41, 41-23 and 23-29. Hence  $W_{32,40} = 1$ ,  $W_{40,19} = 1$  etc... These fiber links are presented separately for Ring R0, R1 and R2 in Figure 4.6.

The pseudo-code for Ring Creation is presented in Algorithm 1.

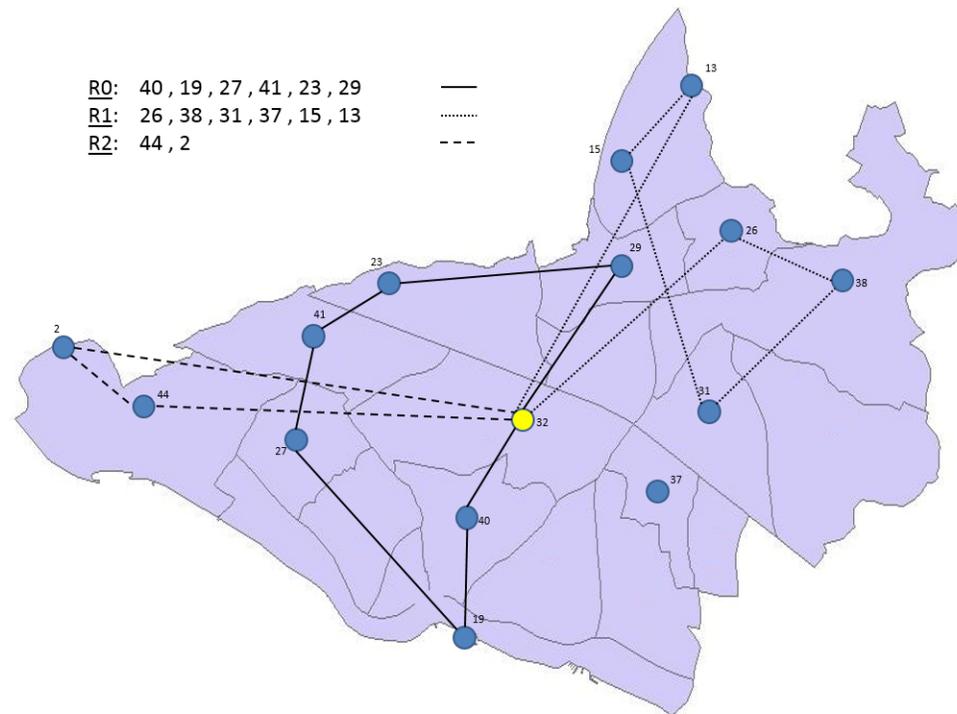
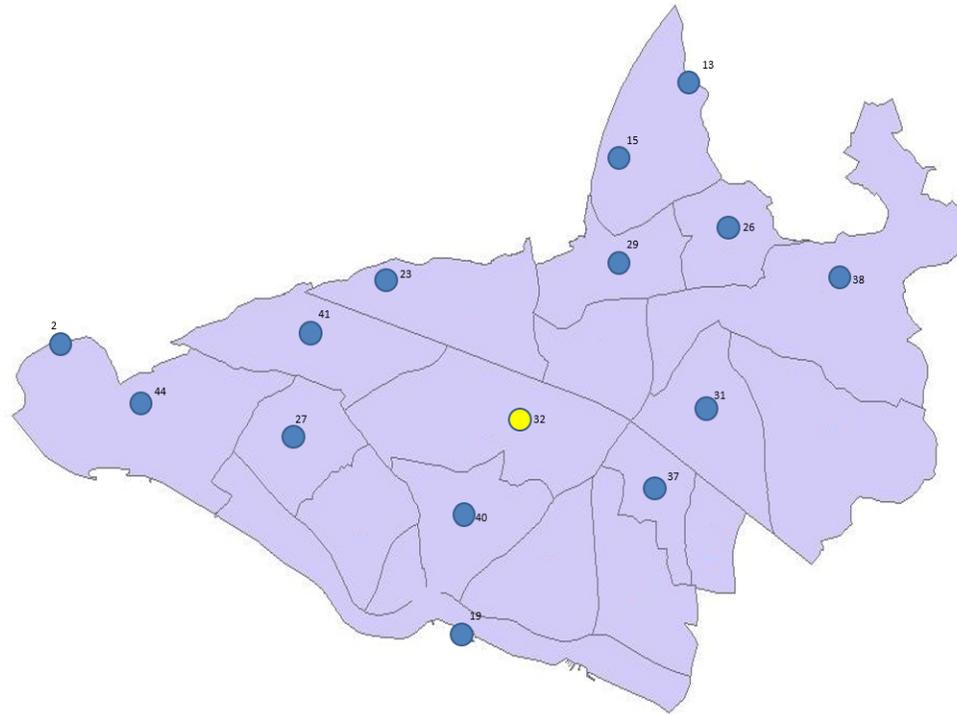


Figure 4.6: Ring Example Network

---

**Algorithm 1: Ring Creation Algorithm**

---

**Data:** Sets  $K, N$  parameters  $central, l_{ij}, c_{ij} \quad \forall i, j \in N,$   
 $mbcentral, mbthreshold, dBcapacity, declineps, declineway f_1$

- 1 Eligible set  $E$  : demand nodes sorted by Nearest Neighbor ;
- 2  $y_{j,1} = 1 \quad \forall j \in K$  ;
- 3  $P1_j^j = P2_j^j = 0 \quad \forall j \in K$ ;
- 4  $M1_j^j = M2_j^j = mbcentral \quad \forall j \in K$ ;
- 5  $V$  : empty,  $RING$  : empty and  $q = 0$ ;
- 6 **while**  $E$  : nonempty **do**
- 7     Create Ring  $R_q$  ;
- 8     Add the first element of  $E$ , say  $\hat{i}$ , to Ring  $R_q$  ;
- 9      $P1_{\hat{i}}^{\hat{i}} = declineway * l_{central, \hat{i}} + declineps * f_1$  ;
- 10     $M1_{\hat{i}}^{\hat{i}} = mbcentral$  ;
- 11    Remove  $\hat{i}$  from set  $E$    Assign  $currentNode = \hat{i}$  ;
- 12     $t = 1 \quad repeat = true$  ;
- 13    **while**  $t \leq card(E)$  and  $repeat = true$  **do**
- 14      $t^{th}$  element of  $E$ :  $\hat{j}$  ;
- 15      $P1_{\hat{j}}^{\hat{j}} = P1_{currentNode}^{currentNode} + declineway * l_{currentNode, \hat{j}} + declineps * f_1$  ;
- 16      $M1_{\hat{j}}^{\hat{j}} = \frac{M1_{currentNode}^{currentNode}}{2^{f_1}}$  ;
- 17     **if**  $(P1_{\hat{j}}^{\hat{j}} \leq dBcapacity)$  and  $(M1_{\hat{j}}^{\hat{j}} \geq mbthreshold)$  and  
       $(P1_{\hat{j}}^{\hat{j}} + declineway * (l_{central, \hat{j}} - l_{central, currentNode}) \leq dBcapacity)$   
      **then**
- 18        Add node  $\hat{j}$  to Ring  $R_q$  ;
- 19         $currentNode = \hat{j}$  ;
- 20     **end**
- 21     **else**
- 22         $P1_{\hat{j}}^{\hat{j}} = 0 \quad M1_{\hat{j}}^{\hat{j}} = mbcentral \quad repeat = false$  ;
- 23     **end**
- 24      $t = t + 1$
- 25    **end**
- 26    Add  $R_q$  to  $RING$ .   Remove the elements of  $R_q$  from  $E$
- 27     $q = q + 1$
- 28 **end**
- 29 **forall the**  $q$  such that Ring  $R_q \in RING$  **do**
- 30    **if**  $card(R_q) = 1$  **then**
- 31     Add  $R_q$ 's elements to set  $V$    Delete  $R_q$  from set  $RING$
- 32    **end**
- 33 **end**
- 34 Based on ring structure feature, calculate the values  $P1_i^j, M1_i^j, P2_i^j, M2_i^j$   
     $\forall i, j \in K, i \neq j$  ;

---

---

**Algorithm 1: Ring Creation Algorithm (Continued)**

---

```
1 forall the  $q$  such that  $R_q \in RING$  do
2   | Connect the fiber links starting from and ending at central with respect
   | to insertion order of the elements of  $R_q$ ;
3   | Construct primary and secondary paths for elements of  $R_q$ 
4 end
5 if  $V$  : nonempty then
6   | forall the node  $h \in V$  do
7     |  $y_{h,1} = 0$ 
8   | end
9     | Create set FirstPS, LastPS and EligNPS ;
10  | forall the  $q$  such that Ring  $R_q \in RING$  do
11    | Add the first element of  $R_q$  to set FirstPS
12    | Add the Last element of  $R_q$  to set LastPS
13  | end
14  |  $EligNPS = FirstPS \cup LastPS$ 
15  | forall the  $h \in V$  do
16    | From EligNPS, find the closest node to  $h$  : call it  $u$ 
17    |  $W_{u,h} = 1$     $W_{central,h} = 1$ 
18    | if  $u \in FirstPS$  then
19      | Use the primary path of  $u$  for the first path of  $h$ . Hence the first
      | path of  $h$  will be:  $x_{central,u}^h = x_{u,h}^h = 1$ 
20      |  $P1_h^h = P1_u^u + declineway * W_{u,h}$ 
21      |  $M1_h^h = \frac{M1_u^u}{2^{f1}}$ 
22      |  $P1_u^h = P1_u^u$  and  $M1_u^h = M1_u^u$ 
23      | The second path of  $h$  :  $z_{central,h}^h = 1$ 
24      |  $P2_h^h = declineway * W_{central,h}$ 
25      |  $M2_h^h = mbcentral$ 
26      | Remove node  $u$  from FirstPS and EligNPS
27    | end
28    | else
29      | Use the secondary path of  $u$  for the first path of  $h$ . Hence the
      | first path of  $h$  will be:  $z_{central,u}^h = z_{u,h}^h = 1$ 
30      |  $P2_h^h = P2_u^u + declineway * W_{u,h}$ 
31      |  $M2_h^h = \frac{M2_u^u}{2^{f1}}$ 
32      |  $P2_u^h = P2_u^u$  and  $M2_u^h = M2_u^u$ 
33      | The second path of  $h$  :  $x_{central,h}^h = 1$ 
34      |  $P1_h^h = declineway * W_{central,h}$ 
35      |  $M2_h^h = mbcentral$ 
36      | Remove node  $u$  from FirstPS and EligNPS
37    | end
38  | end
39 end
Output:  $W_{ij}, y_{j1}, x_{ij}^k, z_{ij}^k, P1_i^k, P2_i^k, M1_i^k, M2_i^k$ 
```

---

## 4.2 Clustering Algorithm

This algorithm aims to group the demand nodes exclusively based on their distance values and then solve each cluster independently. Each cluster set will be taken as the input demand node set separately for the SGFNDP mathematical model described in Chapter 3. We do not wish to create too small or too large clusters since very large clusters will cause inefficiency in terms of CPU times and very small clusters will not yield meaningful results for the problem. We specify a  $\delta$  threshold value while adding the nodes to the clusters and  $\delta$  is the average pairwise distances. Hence  $\delta = \sum_{i,j \in N, j \neq i} l_{ij} / \text{card}(N)$ . The lower bound of each cluster cardinality is  $\beta_1$ . Hence each cluster should consist of at least  $\beta_1$  demand nodes. This algorithm consists of two steps and the upper bounds for the cluster cardinality for Step 1 and Step 2 are  $\beta_2$  and  $\beta_3$ , respectively.  $\beta_2$  is utilized for initial cluster generation whereas  $\beta_3$  is the upper bound for each generated cluster and their utilization will be explained in detail below.

### Clustering Algorithm - Step 1

The Clustering Algorithm consists of 2 steps. In Step 1, we seek to create exclusive clusters such that each cluster's cardinality is between  $\beta_1$  and  $\beta_2$ . For the first trial,  $\beta_2 = \beta_3$ . Initially, Eligible Set  $E$  for cluster generation corresponds to the demand node set.  $E = K$ .

The algorithm first finds the closest two nodes from set  $E$ , say  $i$  and  $j$ . If the distance between  $i$  and  $j$  obeys the  $\delta$  limit value (ie,  $l_{i,j} \leq \delta$ ), then we generate a cluster. These two chosen nodes are removed from Eligible Set  $E$ . This procedure is referred as "find2closest".

After this procedure, the nodes in set  $E$  are scanned one by one for node insertion to the current cluster. For instance, node  $k \in E$  will be chosen and added to that cluster if node  $k$  gives the "minimum average distance" to the current cluster and this "minimum average distance" value obeys the  $\delta$  limit. Then, the chosen node  $k$  is deleted from Eligible Set  $E$ . This procedure is referred to as "find1closest".

In order to find node  $k$ 's average distance value to cluster  $Z$  (let us denote this value as  $avgDistance(k, Z)$ ), we first sum the distances from  $k$  to every element of the cluster. Then we divide this value to the cardinality of  $Z$ . For instance, consider that the cluster  $Z$  consists of node  $i, j, t, h$ . (ie,  $Z = \{i, j, t, h\}$ ). Hence,  $card(Z) = 4$ . Then, node  $k$ 's average distance to cluster  $Z$  is calculated as follows:

$$avgDistance(k, Z) = \frac{l_{k,i} + l_{k,j} + l_{k,t} + l_{k,h}}{4}$$

The addition of the nodes to the cluster  $C$  (ie, “find1closest” procedure) can occur as long as upperbound  $\beta_2$  is not exceeded (ie,  $card(C) \leq \beta_2$ ). We perform update operation on eligible set  $E$  right after the node insertion.

After the “find1closest” procedure, we may face a situation where the resulting cluster  $C$  is too small (ie,  $card(C) < \beta_1$ ). In this case, we force the addition of nodes based on minimum  $avgDistance$  values without considering  $\delta$  limit. We stop this addition we reach lowerbound for cardinality of Cluster  $C$  (ie, we stop this procedure when  $card(C) = \beta_1$ ). This procedure is referred as “force1closest”.

The algorithm generates a new cluster in these two cases: (i) no addition to the previous cluster is possible due to  $\beta_2$  upperbound. (ii) the lowerbound cardinality  $\beta_1$  is reached after forceful addition phase. The algorithm continues in this manner until every demand node is scanned for cluster generation (ie, when set  $E$  is empty).

In this step, we wish to obtain proper clusters such as each cluster's cardinality is between  $\beta_1$  and  $\beta_2$ . However, after performing this step, we may face a situation where some nodes are not properly clustered. This can happen due to two reasons:

(1) Initial cluster generation (ie, “find2closest” procedure) can only occur if there are two nodes  $i$  and  $j$  such that  $l_{i,j} \leq \delta$ . We may have a case where the remaining nodes in the eligible set  $E$  does not satisfy this distance constraint. Hence, there

are non-chosen nodes in this case.

(2) The last generated cluster's cardinality may be less than  $\beta_1$  since there might be no more node left in eligible set  $E$  for applying forceful addition. Hence, there are nodes that are elements of a small size cluster in this case.

If we incur a situation where there are nodes that are not properly clustered (ie, one of these situation occurs after Step 1), then we perform Step 2.

### **Clustering Algorithm - Step 2**

We generate a list  $NC$  that consists of the outcast nodes (the nodes that are not properly clustered by Step 1 due to non-chosen and/or small size cluster belongingness).

We aim to add these outcast nodes to the proper clusters that are generated by Step 1 (the cardinality of a proper cluster is between  $\beta_1$  and  $\beta_2$ ). We create an Eligible Cluster Set, *EligClus*, that corresponds to the clusters with cardinality strictly less than  $\beta_3$ . The difference between  $\beta_2$  and  $\beta_3$  becomes important for this step. If  $\beta_2 = \beta_3$ , then Eligible Cluster Set will be empty and we will not be able to insert the outcast nodes anywhere. Since we aim to group all nodes, the corresponding output becomes infeasible. In this situation, we lower  $\beta_2$  by 1 to get rid of infeasibility for the second trial. We apply the same algorithm with lowered  $\beta_2$  value. Note that  $\beta_2$  must always be between  $\beta_1$  and  $\beta_3$ .

When  $\beta_2 < \beta_3$ , Eligible Cluster Set becomes non-empty for outcast node insertion. For every element in  $NC$  (ie, for each outcast node), we find the most suitable Cluster from Eligible Cluster Set (*EligClus*) based on minimum average distance. Then we insert this outcast node to the chosen Cluster. After the insertion, we check again for eligibility statues of the chosen cluster. This procedure is referred to as "ClusterChosen".

We apply the same procedure as long as every outcast node is inserted. If there are still outcast nodes after the insertion, then this output becomes infeasible and we lower  $\beta_2$  by 1 again to get rid of infeasibility for the following trial. In

this manner, we lower  $\beta_2$  until we obtain feasibility. Since  $\beta_1$  corresponds to the lower bound for cluster cardinality, we can keep lowering  $\beta_2$  as long as  $\beta_2 \geq \beta_1$  holds.

If we encounter a case where  $\beta_2$  is updated such that  $\beta_2 < \beta_1$ , then this case becomes infeasible. In order to maintain feasibility, we should consider lowering  $\beta_1$  in this case. Our algorithm does not lower  $\beta_1$  since it takes  $\beta_1$  and  $\beta_3$  as inflexible parameters.

### **Clustering Algorithm - Example**

We provide an example of this algorithm for illustration. Consider that we are given demand node set  $K$  (14 demand nodes), and *central* at node 32 as in Figure 4.7. Say that we choose  $\beta_1 = 4$  and  $\beta_3 = 6$ . For the first trial,  $\beta_2 = \beta_3 = 6$ .  $\delta$  is calculated from the average pairwise distance data and  $\delta = 14.74$ .

The resulting clusters of the first and second trial are given in Figure 4.7.

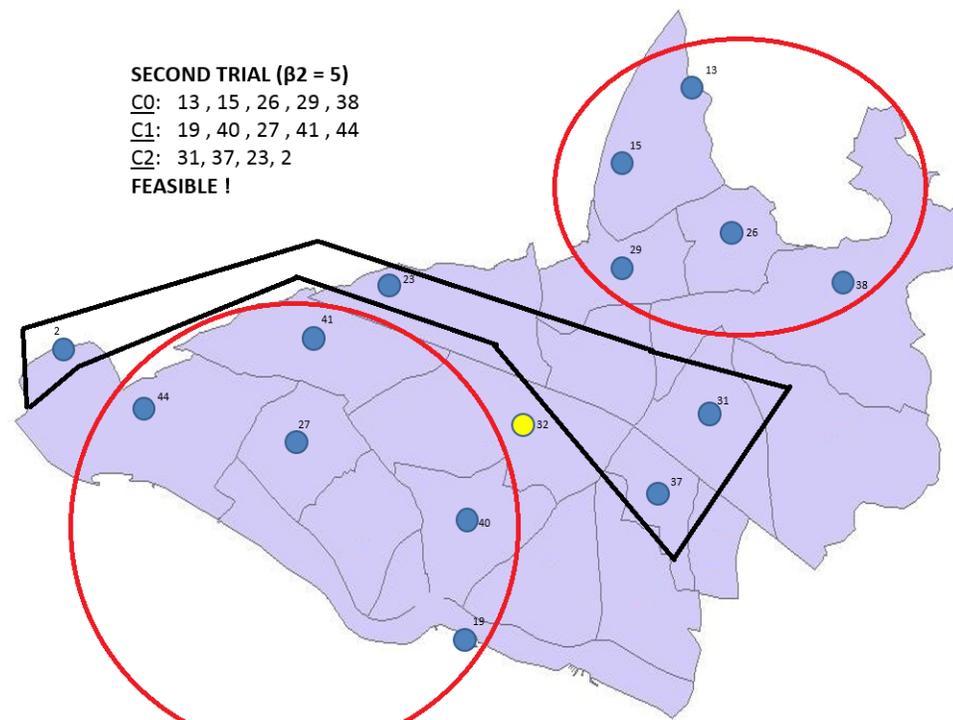
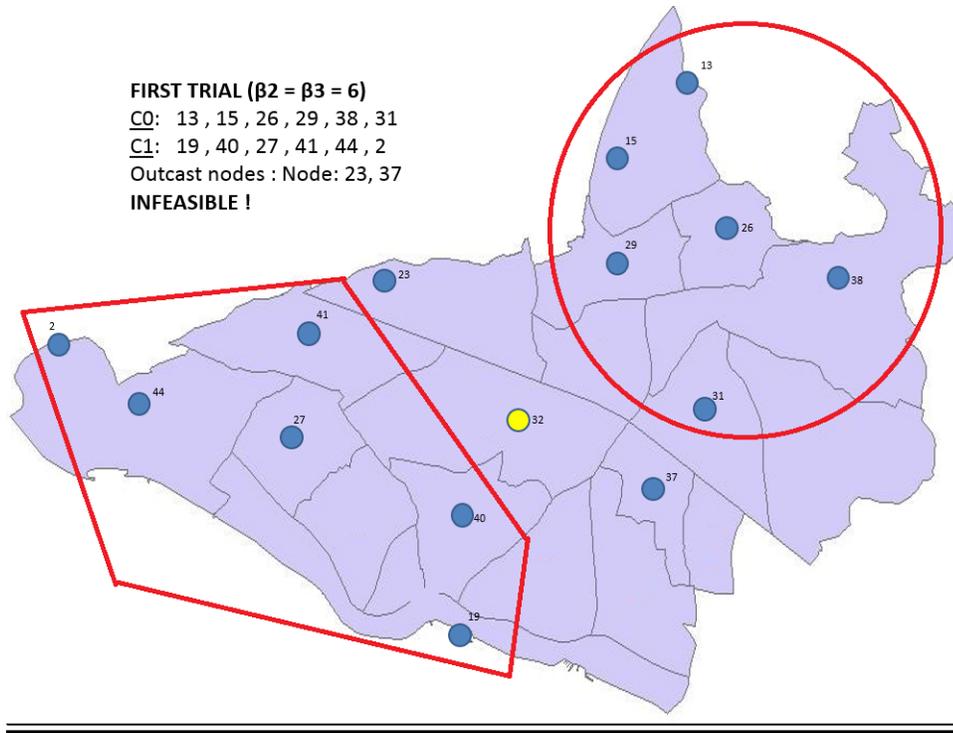


Figure 4.7: Clustering Example Network-  $\beta_1 = 4$  and  $\beta_3 = 6$

### Step 1 (First trial)

For the first trial,  $\beta_1 = 4$  and  $\beta_2 = \beta_3 = 6$ . As stated before, eligible node set E consists of all demand nodes. Initially  $\text{card}(K)/2 = 7$  clusters are generated. The first two chosen nodes for cluster C0 are 13 and 15 since they are the closest nodes in E and  $l_{13,15} \leq \delta$ . We add these nodes to Cluster C0 and removed them from E. Then, one by one addition occurs. The nodes 26, 29, 38, 31 are added to Cluster C0 and removed from E. No forceful addition became needed for this cluster. Since upperbound is reached, no more nodes can be added to C0.

The first two chosen nodes for cluster C1 are 19 and 40. We insert these nodes to Cluster C1 and delete them from E. Then, we add the nodes 27, 41, 44 and 2 to Cluster C1 and remove them from E. No forceful addition became needed for this cluster either. The upperbound is reached for C1, however E is still nonempty. E= 23, 27. Since  $l_{23,37} = 15 > \delta = 14.74$ , nodes 23 and 37 cannot form a cluster. These nodes 23 and 37 are outcast nodes due to non-chosen case. We go to Step 2.

### Step 2 (First trial)

Since  $\beta_2 = \beta_3$ , there is no eligible cluster set and the outcast nodes cannot be inserted anywhere (EligClus is empty). We lower  $\beta_2$  value by 1 in order to get rid of infeasibility.  $\beta_2 = 5$

### Step 1 (Second trial)

For this trial,  $\beta_1 = 4$  and  $\beta_2 = 5$  since we lowered  $\beta_2$ . Eligible Set E consists of all demand nodes. The first two chosen nodes for cluster C0 are 13 and 15. We add these nodes to Cluster C0 and removed them from E. Then, one by one addition occurs. The nodes 26, 29 and 38 are added to Cluster C0 and removed from E. We stop since we reached upperbound  $\beta_2$ . No forceful addition became needed for this cluster.

The first two chosen nodes for cluster C1 are 19 and 40. We add these nodes to Cluster C1 and remove them from E. Then, we add the nodes 27, 41, 44 to

Cluster C1 and remove them from E. The upperbound  $\beta_2$  is reached for C1.

The first two chosen nodes for cluster C2 are 31 and 37. We add these nodes to Cluster C3 and remove them from E. Then, we add the node 23 and remove it from E. Now  $E = \{2\}$ . We have  $avgDistance(2, C2) = 21 > \delta = 14.74$ , hence we cannot perform “find1closest” procedure. Since  $card(C3) < \beta_1$ , we apply “force1closest” procedure. Hence we forcefully add node 2 to C2. Now,  $card(C2) = \beta_1 = 4$ , hence C2 is a proper cluster.

Every demand node is properly clustered since every cluster obeys the  $\beta_1$  and  $\beta_3$  bounds. Hence this output is feasible and we do not perform Step 2.

The output:

The elements of C0 : 13, 15, 26, 29, 38

The elements of C1 : 19, 40, 27, 41, 44

The elements of C2 : 31, 37, 23, 2

After the clustering phase, we solve the sub-problems corresponding to the clusters. Hence we solve 3 sub-problems in this case independently. We give Cluster 0 and *central* to the mathematical model and obtain the optimal solution. We follow the same procedure for Cluster 1 and Cluster 2. Then we aggregate the solutions and obtain the output for the original problem.

The pseudo-code for Clustering Algorithm-Step 1 and Clustering Algorithm-Step 2 is presented in Algorithm 2 and Algorithm 3 respectively.

For our problem, we can obtain very fast solutions when the input node cardinality is 8 (ie,  $card(N) = 8$ ). This corresponds the case where there are 7 demand nodes (ie,  $card(K) = 7$ ). Since the clustering operation is only performed on demand nodes, we will consider the upper bound for cluster cardinality as 7 (ie,  $\beta_3 = 7$ ). Hence the sub-problems based on the clusters will be solved very efficiently in terms of CPU.

We have more flexibility for the choice of the lowerbound for cluster cardinality,  $\beta_1$ . We can choose  $\beta_1$  any value as long as  $\beta_1 > 1$ .  $\beta_1 = 1$  would not yield a feasible solution for SGFNDP since we have to maintain 2-node disjoint paths.

A high  $\beta_1$  value might yield infeasible solutions for the algorithm whereas a small  $\beta_1$  might cause a bad objective function value. Hence, there exists a tradeoff for the choice of  $\beta_1$ .

We will take  $\beta_1 = 4$  and  $\beta_3 = 7$ .

---

**Algorithm 2:** Cluster Algorithm - Step 1

---

**Data:** Set  $K, N$ , parameters  $l_{ij}, \delta, \beta_1, \beta_2$

1  $\delta$  is determined by  $l_{ij}$  inputs. (average of  $l_{ij}$  values). Set of eligible nodes:  
 $E = K$ , *Set of Clusters* :  $CLUSTER = \text{empty}$  and set NC: empty;

2 The index for cluster to be filled:  $q$  and  $q = 0$

3 **while**  $E : \text{nonempty}$  **do**

4 | (The procedure is find2closest in Set E) ;

5 | Create cluster  $C_q$  ;

6 | Let  $\hat{i}$  and  $\hat{j}$  be the two closest nodes in set E such that  $l_{\hat{i}\hat{j}} \leq \delta$  ;

7 | Add  $\hat{i}$  and  $\hat{j}$  to the cluster  $C_q$ . Remove  $\hat{i}$  and  $\hat{j}$  from E. ;

8 | **while**  $E : \text{nonempty}$  **do**

9 | | (The procedure is find1closest in Set E to Cluster  $C_q$ )

10 | |  $\hat{t} = \text{argmin}_{t \in E} \text{avgDistance}(t, C_q)$  ;

11 | | **if**  $\text{card}(C_q) < \beta_2$  and  $\text{avgDistance}(t, C_q) \leq \delta$  **then**

12 | | | Add  $\hat{t}$  to cluster  $C_q$ . Remove  $\hat{t}$  from E

13 | | **end**

14 | | **else**

15 | | | Break the while cycle

16 | | **end**

17 | **end**

18 | **while**  $E : \text{nonempty}$  and  $\text{card}(C_q) < \beta_1$  **do**

19 | | (The procedure is force1closest in Set E to Cluster  $C_q$ )

20 | |  $\hat{k} = \text{argmin}_{k \in E} \text{avgDistance}(k, C_q)$  ;

21 | | Add  $\hat{k}$  to cluster  $C_q$ . Remove  $\hat{k}$  from E

22 | **end**

23 | Add  $C_q$  to  $CLUSTER$   $q = q + 1$

24 **end**

25 **forall the node**  $i \in K$  **do**

26 | **if** node  $i \notin C_q \quad \forall q$  **then**

27 | | add node  $i$  to set NC

28 | **end**

29 **end**

30 **forall the**  $q$  **such that**  $C_q \in CLUSTER$  **do**

31 | **if**  $\text{card}(C_q) < \beta_1$  **then**

32 | | Add  $C_q$ 's elements to NC and delete cluster  $C_q$  from  $CLUSTER$

33 | **end**

34 **end**

35 **if** NC: empty **then**

36 | Stop. The output is feasible

37 **end**

38 **else**

39 | Go to Step 2.

40 **end**

**Output:** Set of Proper Clusters:  $C_q \in CLUSTER$

---

---

**Algorithm 3:** Cluster Algorithm - Step 2

---

**Data:** Set  $NC, C_q \in CLUSTER$ , parameters  $l_{ij}, \beta_1, \beta_2, \beta_3$

```
1 isFeasible = true ;
2 Set of eligible clusters: EligClus: empty ;
3 forall the  $C_q \in CLUSTER$  do
4   | if  $card(C_q) < \beta_3$  then
5   |   | Add  $C_q$  to EligClus
6   | end
7 end
8 if EligClus: empty then
9   | isFeasible = false. Finish Step-2 due to infeasibility.
10  | Update  $\beta_2$  for the next trial of Step 1. ;
11  |  $\beta_2 = \beta_2 - 1.$  if  $\beta_2 \geq \beta_1$  then
12  |   | Go to Step-1 with this  $\beta_2$  value
13  | end
14  | else
15  |   | Stop. This problem is infeasible since  $\beta_1$  is high
16  | end
17 end
18 forall the node  $j \in NC$  do
19   | (The procedure is ClusterChosen for node  $j$  to EligClus) ;
20   |  $chosenCluster = \operatorname{argmin}_{C_q \in EligClus} avgDistance(j, C_q)$  ;
21   | Remove node  $j$  from  $NC$  Add node  $j$  to chosenCluster
22   | if  $card(chosenCluster) = \beta_3$  then
23   |   | Remove chosenCluster from EligClus
24   | end
25 end
26 if  $NC : nonempty$  then
27   | isFeasible = false. ;
28   | Finish Step-2 due to infeasibility. ;
29   | Update  $\beta_2$ :  $\beta_2 = \beta_2 - 1$ 
30   | if  $\beta_2 \geq \beta_1$  then
31   |   | G
32   | end
33   | o to Step-1 with this  $\beta_2$  value
34   | else
35   |   | Stop. This problem is infeasible since  $\beta_1$  is high
36   | end
37 end
```

**Output:**  $isFeasible, \beta_2, C_q \in CLUSTER$

---

# Chapter 5

## Data and Computational Analysis

Although the mathematical model provides optimal solutions, when the problem dimension increases, it becomes inefficient to solve the problem in terms of CPU. In order to obtain good solutions quickly, we proposed two constructive heuristics as described in the previous chapter. In this chapter, we will give detailed information about the data sets. We will test the behavior of the mathematical model and evaluate the performances of the heuristic algorithms. Also, we provide the interpretation of the corresponding outputs.

Computational studies consist of five parts: In the first part, we examine the effect of MIP start (also referred to as advanced start or warm start) for our mathematical model. MIP start is utilized in order to lower the computation time by providing a feasible solution to the model beforehand. In the second part, the effect of the location of *central* node is taken into consideration. The outputs of the heuristics are presented in the third and fourth parts, respectively. Lastly, we evaluate and compare the performances of the heuristic algorithms.

## 5.1 Data

In the section, the information about the data sets is presented. We test our model and algorithms using a data set based on a district of İstanbul, Turkey. This data set, Kartal, has 45 nodes as illustrated in Figure 5.1. We will consider central station in various places since we wish to generate problem instances and examine the effect of the location of *central*. For Kartal distinct, node 8 is in the east, node 14 is in the north, node 18 is in the south, node 24 is in the west and node 32 is in the center geographically. These nodes will be taken as *central* for Kartal for different instances.

Since we can obtain an optimal solution quickly when the node set cardinality is 8, a small data subset is constructed accordingly. Figure 5.2(a) represents this network where the demand nodes are the blue nodes and the changing central stations are the yellow nodes. We also created a medium data subset (with node cardinality 15) in the same manner as shown in Figure 5.2(b). For this data set, we obtain a best integer value in a predetermined time. We will compare the best objective values with heuristic solutions for the medium data set.

The model fails to obtain even a best integer value for full data set (ie, when the node cardinality is 45) when the computation time is 4 hours. Hence, we will utilize full data set in order to compare the performances of the heuristic algorithms. This input set will be used in order to assess the solution quality of the Heuristic Algorithms among each other.

For a quick reference about the instance features, one can refer to Table 4.1. For example, Instance "K8S" corresponds to the Kartal data set with node cardinality 8 and the central station locates at the south. (ie, the central station is node 18). Likewise, "K15W" is Kartal Data set with node cardinality 15 and *central* location at the west.

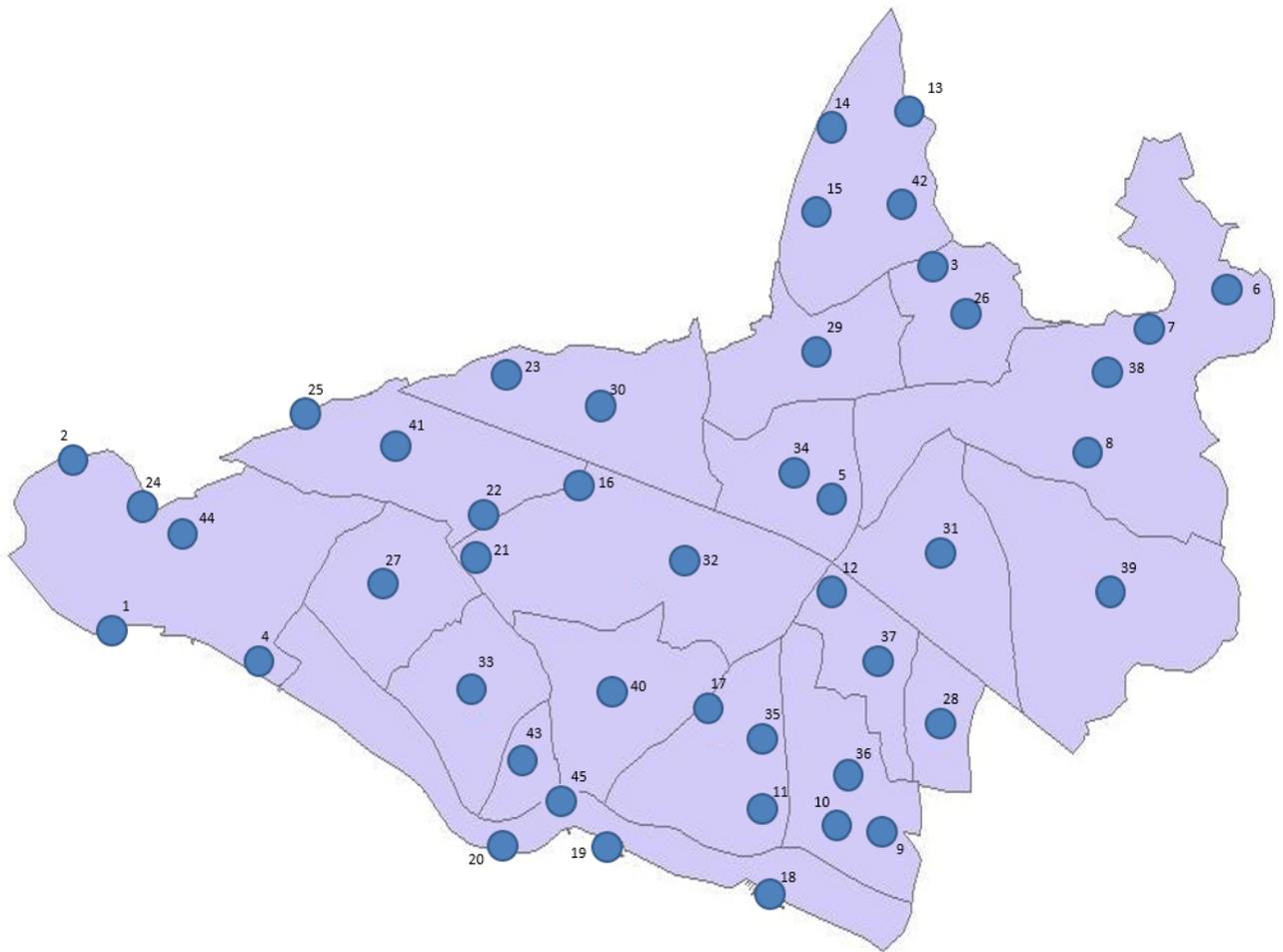
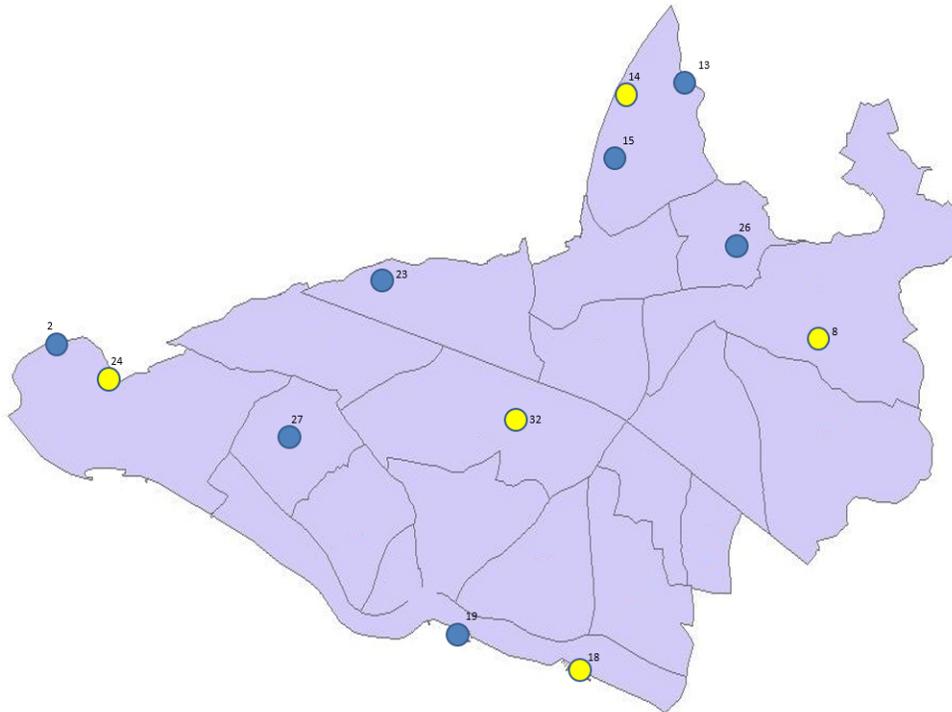
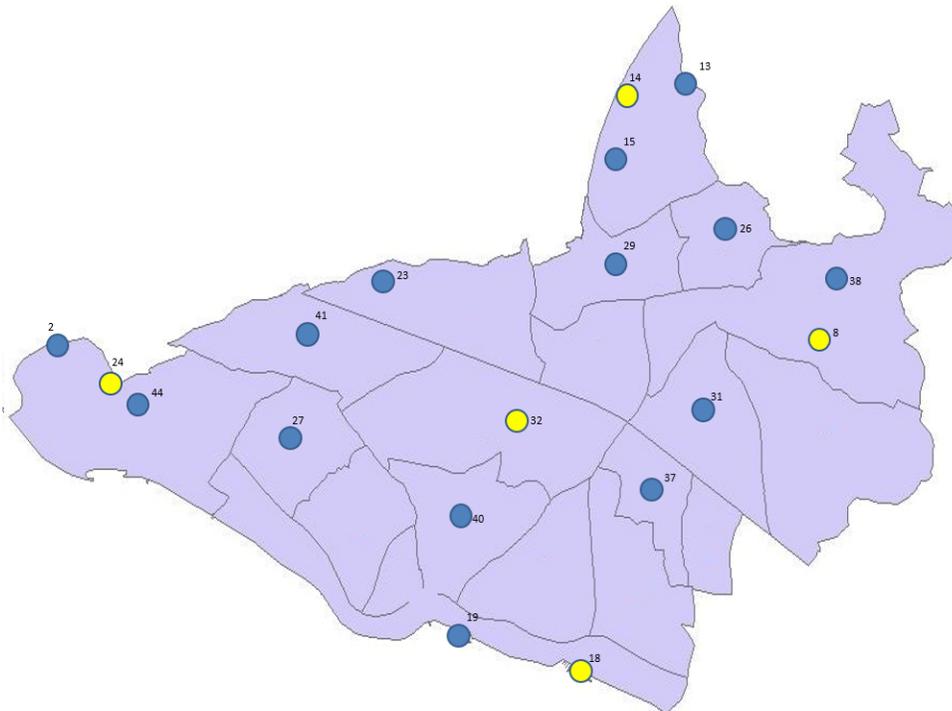


Figure 5.1: The Map of Kartal



(a) Kartal Node Set with 7 Demand Nodes and Different Central Station Locations



(b) Kartal Node Set with 14 Demand Nodes and Different Central Station Locations

Figure 5.2: The Map of Kartal with 7 and 14 demand nodes

Instance	Central Station	Demand Nodes	Cardinality
K8E	Node 8: In the east	2, 13, 15, 19, 23, 26, 27	8
K8N	Node 14: In the north	2, 13, 15, 19, 23, 26, 27	8
K8S	Node 18: In the south	2, 13, 15, 19, 23, 26, 27	8
K8W	Node 24: In the west	2, 13, 15, 19, 23, 26, 27	8
K8C	Node 32: In the center	2, 13, 15, 19, 23, 26, 27	8
K15E	Node 8: In the east	2, 13, 15, 19, 23, 26, 27, 29, 31, 37, 38, 40, 41, 44,	15
K15N	Node 14: In the north	2, 13, 15, 19, 23, 26, 27, 29, 31, 37, 38, 40, 41, 44,	15
K15S	Node 18: In the south	2, 13, 15, 19, 23, 26, 27, 29, 31, 37, 38, 40, 41, 44,	15
K15W	Node 24: In the west	2, 13, 15, 19, 23, 26, 27, 29, 31, 37, 38, 40, 41, 44,	15
K15C	Node 32: In the center	2, 13, 15, 19, 23, 26, 27, 29, 31, 37, 38, 40, 41, 44,	15
K45E	Node 8: In the east	Nodes 1..45 except node 8	45
K45N	Node 14: In the north	Nodes 1..45 except node 14	45
K45S	Node 18: In the south	Nodes 1..45 except node 18	45
K45W	Node 24: In the west	Nodes 1..45 except node 24	45
K45C	Node 32: In the center	Nodes 1..45 except node 32	45

Table 5.1: Features of the data set

## Parameter Values

The parameters are explained in detail in Chapter 3. A quick reminder is as follows:

The highway distances  $l_{ij}$  are generated by using ArcGIS roadmap and the locations of the nodes are obtained by using Google Maps. Detailed information about the data set construction is provided in [23].

Fiber wire installment cost  $c_{ij} = l_{ij} \quad \forall i, j \in N$ .

$\alpha$  indicates the proportion between the splitter and fiber wiring cost and it will taken as zero as explained before.

The port number values  $f_q$  are explained in Chapter 3.  $f_1 = 1, f_2 = 2, f_3 = 3$ , and  $f_4 = 4$ . The company specified the parameters that are associated with the service requirement.  $declineps = 3, declineway = 0.2, dBcapacity = 35, mbcentral = 2500, mbcentral = 25000$  and  $mbthreshold = 75$ .

As stated in the previous chapter, for Clustering Algorithm, we will choose the lowerbound and upperbound for cluster cardinality as 4 and 7, respectively. Hence  $\beta_1 = 4$  and  $\beta_3 = 7$ .

## 5.2 Computational Analysis

Using the parameter values and instances as explained in the previous section, we test and evaluate our model and heuristics. The experiments on mathematical models are conducted using CPLEX 12.6. The heuristic algorithms are coded in Java (Eclipse IDE) and ran on a personal computer with a processor Intel Core i7 CPU at 2.4 GHz and 16 GB of RAM. The solution time of “Ring Creation Algorithm” is not reported since this heuristic obtains a solution in milliseconds.

## 5.2.1 Performance of MIP Start Option

IBM ILOG CPLEX Optimizer provides MIP start (also referred to as advanced start or warm start) feature in order to decrease the computational time. In this case, we have an initial feasible solution beforehand by our previous knowledge about the problem. We supply this solution to CPLEX, hence the model takes this initial value as an incumbent solution instead of searching for one.

In order to obtain initial solutions for MIP start feature, we solve the “Ring Creation Algorithm” beforehand and supply its solution to the model. The effect of MIP start is given in Table 5.2. As we can see from the best integer values and the gap, MIP start feature utilization is advantageous in terms of solution quality. The system time is limited to 4 hours for both cases (ie for the cases with MIP start and without MIP start).

<b>Instance</b>	<b>Objective Function Value (without MIP start)</b>	<b>Objective Function Value (with MIP start)</b>	<b>Gap after 4 hours (without MIP start)</b>	<b>Gap after 4 hours (with MIP start)</b>
K15E	(224)	(166)	%55	%40
K15N	(230)	(154)	%56	%36
K15S	(178)	(156)	%44	%36
K15W	(194)	(180)	%50	%46
K15C	(177)	(163)	%43	%38

Table 5.2: Comparison based on MIP Start Feature

## 5.2.2 The effect of the Location of Central Station

In this section, we examine how the location of central station effects the solution output. Table 5.3 and 5.4 represents the outputs of the mathematical model. Figure 5.3 illustrates the optimal network for small data sets. The PS Usage column in the tables gives information about the passive splitter utilization for the corresponding instance. The representation “ $a : b$ ” indicates that  $a^{th}$  type passive splitter is used  $b$  times in the network. Objective function value is represented in the next column. Average DB value (AvgDB) corresponds to the average value for insertion loss at the demand nodes. The end node values are utilized for this calculation. Average DB is derived as follows:

$$AvgDB = \frac{\sum_{k \in K} P1_k^k + P2_k^k}{card(K) * 2}$$

Likewise, Average MB (AvgMB) corresponds to the average value for speed level at the demand nodes. It is derived similarly as follows:

$$AvgMB = \frac{\sum_{k \in K} M1_k^k + M2_k^k}{card(K) * 2}$$

Best bound indicates the best bound on the root node. The last column in Table 5.3 represents the CPU performance in the terms of seconds, hence it corresponds to the solution time. As we can see, the objective function values do not vary a lot. The reason for this issue is that when the central station changes its place for different instance, it always gets closer to some demand nodes while getting far away from the others. Hence the total distance between *central* and the demand nodes does not vary to a high extent. Therefore, the instances with the same node cardinality have similar objective function values as we can see from Table 5.3 and Table 5.4.

Instance	PS Usage	Objective Function Value	Average DB Value	Average MB Value	CPU (sec)
K8E	1:6 2:1	84	20.6	680.8	5.625
K8N	1:7	81	16.21	881.7	3.937
K8S	1:7	82	18.24	753.35	7.422
K8W	1:7	78	18.66	714.29	6.422
K8C	1:7	84	18.44	747.77	6.937

Table 5.3: Optimal Results of instances with 8 node cardinality

Instance	PS Usage	Objective Function Value (Best Integer)	Best Bound	Average DB Amount (Best Integer)	Average MB Amount (Best Integer)	Gap after 4 hours (Best Integer)
K15E	(1:14)	(166)	(100)	15.69	926.34	%40
K15N	(1:14)	(154)	(98)	14.92	934.71	%36
K15S	(1:14)	(156)	(99)	16.64	703.12	%36
K15W	(1:14)	(180)	(96)	16.43	926.34	%46
K15C	(1:14)	(163)	(100)	16.33	926.34	%38

Table 5.4: Best Integer Result of instances with 15 node cardinality

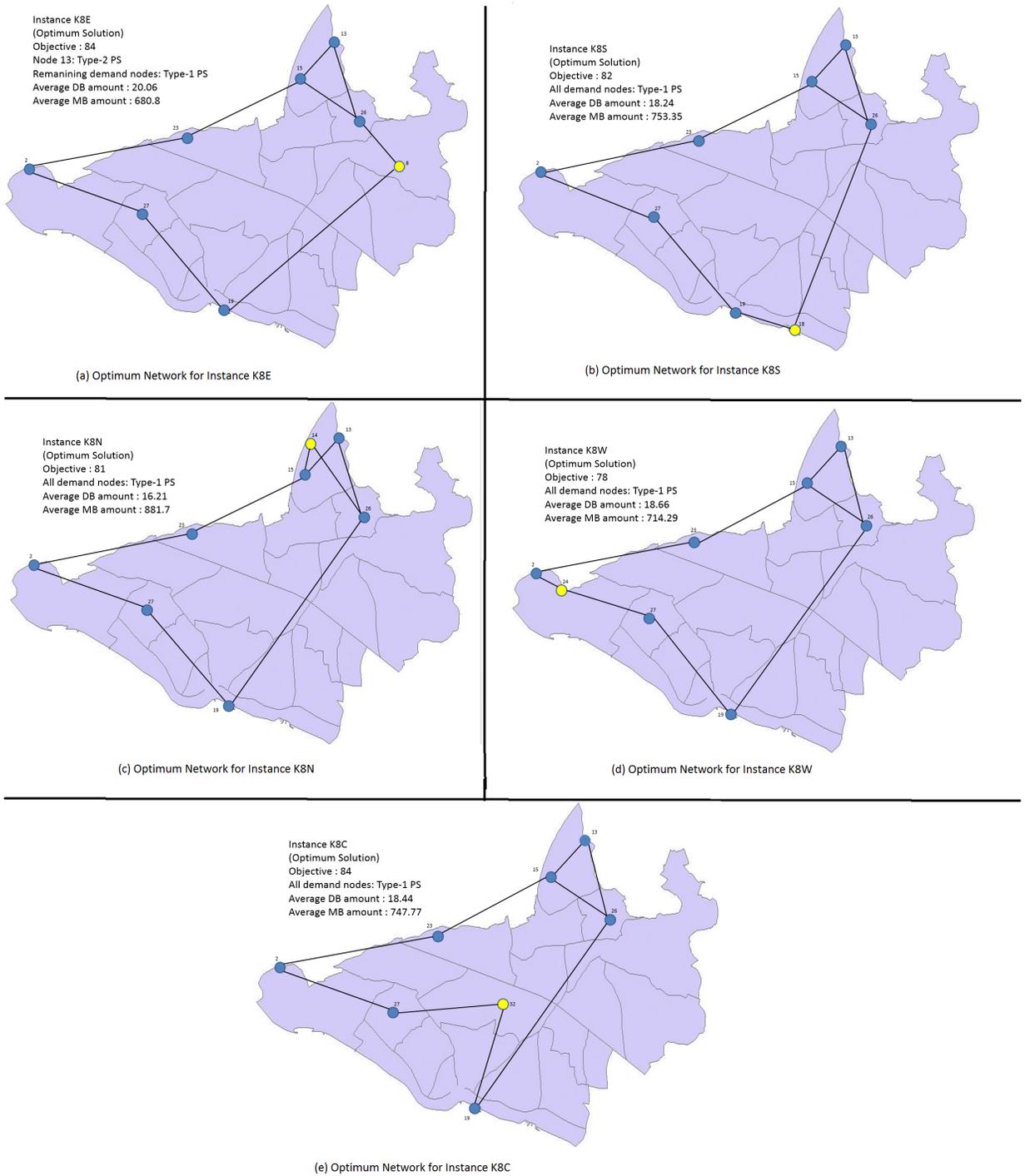


Figure 5.3: Optimum Designs for 7 demand node network and different central locations

### 5.2.3 The outputs of Ring Creation Algorithm

In this section, we present the results of the “Ring Creation Algorithm”. Table 5.5, 5.6 and 5.7 provides the output values for each instance. Figure 5.4 illustrates the resulting network of instance K15N where three fiber link ring structures are constructed for this case.

Instance	Rings	PS Usage	Objective Function Value	Average DB Value	Average MB Value
K8E	R0= { 26 , 15 , 13 , 23 , 27 , 2 }	1:6 (Node 19 : nonPS, its chosen PS: 26)	112	16.39	970.98
K8N	R0= {15 , 13 , 26 , 23 , 27 , 2 }	1:6 (Node 19 : nonPS, its chosen PS: 2)	111	16.34	970.98
K8S	R0= {19 , 27 , 2 , 23 , 15 , 13 }	1:6 (Node 26 : nonPS, its chosen PS: 13)	99	16.31	970.98
K8W	R0= {2 , 27 , 19 , 26 , 15 , 13 }	1:6 (Node 23 : nonPS, its chosen PS: 2)	102	16	970.98
K8C	R0= { 27 , 2 , 23 , 15 , 13 , 26 }	1:6 (Node 19 : nonPS, its chosen PS: 27)	87	15.56	970.98

Table 5.5: Ring Algorithm- Results of instances with 8 node cardinality

Instance	Rings	PS Usage	Objective Function Value	Average DB Value	Average MB Value
K15E	R0= {26 , 38 , 29 , 15 , 13 , 31} R1= {37 , 40 , 19 , 27 , 41 , 23} R2= {44 , 2}	1:14	166	15.04	970.98
K15N	R0= { 15 , 13 , 26 , 38 , 29 , 23} R1= {41 , 27 , 40 , 19 , 37 , 31} R2= {44 , 2}	1:14	173	15.34	970.98
K15S	R0= {19 , 40 , 27 , 41 , 23 , 29} R1= {26 , 38 , 31 , 37 , 15 , 13} R2= {44 , 2}	1:14	174	15.84	970.98
K15W	R0= { 2 , 44 , 27 , 41 , 23 , 29} R1= { 26 , 38 , 31 , 37 , 40 , 19 } R2= { 15 , 13}	1:14	180	15.79	970.98
K15C	R0= {40 , 19 , 27 , 41 , 23 , 29 } R1= {26 , 38 , 31 , 37 , 15 , 13} R2= {44 , 2 }	1:14	163	15.69	970.98

Table 5.6: Ring Algorithm- Results of instances with 15 node cardinality

Instance	Ring Structures (Demand Nodes)	PS Usage	Objective Function Value	Average DB Value	Average MB Value
K45E	R0= {26 , 3 , 7 , 38 , 6 , 29 } R1= { 5 , 34 , 30 , 23 , 25 , 24 } R2= {2 , 44 , 1 , 4 , 27 , 21 } R3= {22 , 41 , 16 , 32 , 40 , 17} R4= {35 , 11 , 10 , 36 , 9 , 28} R5= {37 , 12 , 31 , 39 , 13 , 14 } R6= {15 , 42 , 18 , 19 , 45 , 20 } R7= {43 , 33}	1:44	412	15.47	868.25
K45N	R0= {15 , 42 , 13 , 3 , 26 , 7 } R1= {38 , 6 , 8 , 39 , 31 , 5 } R2= {34 , 29 , 30 , 23 , 25 , 24} R3= {2 , 44 , 1 , 4 , 27 , 21} R4= {22 , 41 , 16 , 32 , 40 , 17 } R5= {35 , 11 , 10 , 36 , 9 , 28} R6= {37 , 12 , 18 , 19 , 45 , 20} R7= {43 , 33}	1:44	434	15.7	868.25
K45S	R0= {10 , 36 , 9 , 28 , 37 , 12 } R1= {17 , 35 , 11 , 19 , 45 , 20} R2= {43 , 33 , 27 , 21 , 22 , 41} R3= {16 , 32 , 40 , 4 , 1 , 44 } R4= {24 , 2 , 25 , 23 , 30 , 29} R5= {5 , 34 , 31 , 39 , 8 , 26 } R6= {3 , 7 , 38 , 6 , 13 , 14 } R7= {15 , 42}	1:44	364	14.82	868.25
K45W	R0= {2 , 44 , 1 , 4 , 27 , 21} R1= {22 , 41 , 16 , 32 , 40 , 17} R2= {35 , 11 , 10 , 36 , 9 , 28} R3= {37 , 12 , 5 , 34 , 29 , 3} R4= {26 , 7 , 38 , 6 , 8 , 39 } R5= {31 , 30 , 23 , 25 , 33 , 43} R6= {45 , 19 , 20 , 18 , 15 , 42} R7= {14 , 13}	1:44	466	16.07	868.25
K45C	R0= {16 , 21 , 22 , 27 , 33 , 43 } R1= {45 , 19 , 20 , 11 , 35 , 17 } R2= {40 , 10 , 36 , 9 , 28 , 37} R3= {12 , 5 , 34 , 29 , 3 , 26} R4= {7 , 38 , 6 , 8 , 39 , 31} R5= {30 , 23 , 25 , 24 , 2 , 44} R6= {1 , 4 , 41 , 18 , 15 , 42} R7= {14 , 13}	1:44	364	14.77	868.25

Table 5.7: Ring Algorithm- Results of instances with 45 node cardinality

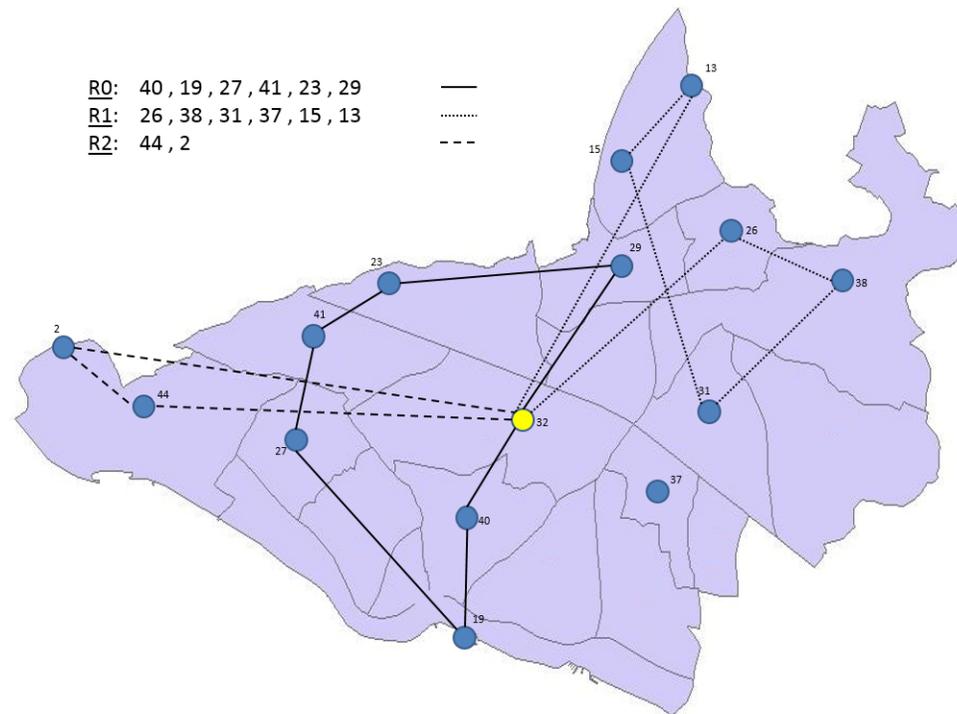
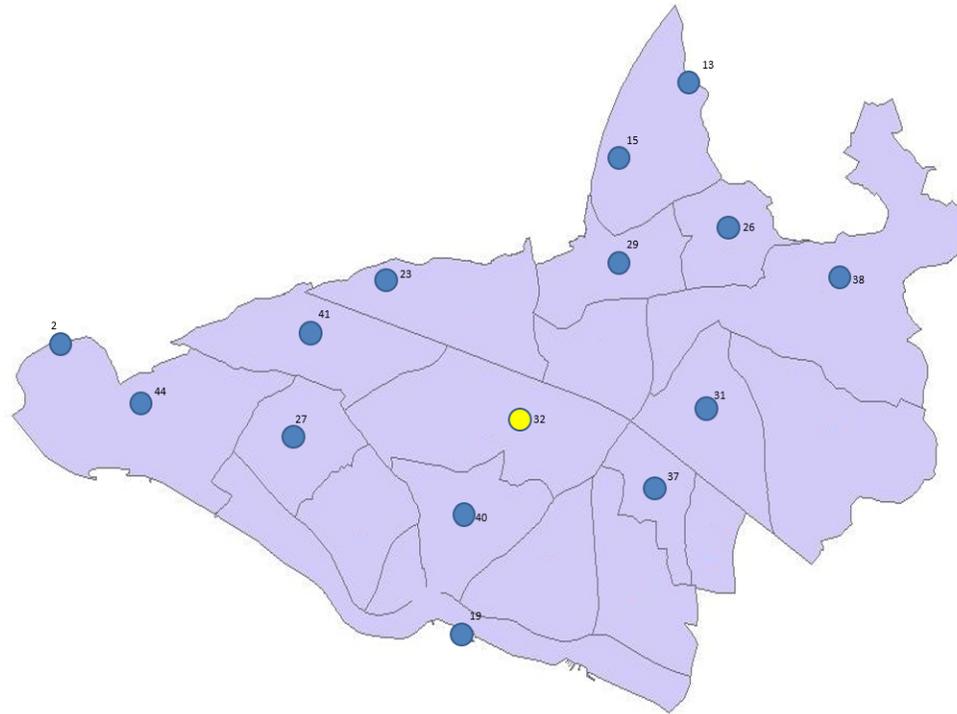


Figure 5.4: Ring Algorithm- Output of instance K15C

## 5.2.4 The outputs of Clustering Algorithm

For the instances K8E, K8N, K8S, K8W and K8C, the clustering algorithm only forms 1 cluster. Hence the results for these cases are the same as the optimal solution and are not included in this section. Table 5.8 and 5.9 presents the outputs of Clustering Algorithm.

C0 corresponds to the cluster 0 which includes *central* and clustered demand nodes. As explained before, every cluster will be solved independently and the separate solutions are aggregated for overall instance output. ObjC0 is the objective function value of the model where the input node set is C0. Likewise, AvgDB-C0 and AvgMB-C0 is the average insertion loss amount and speed level respectively with input node set C0. TimeC0 is the solution time for cluster C0. The values for other clusters, C1, C2 etc... are derived similarly. TotalObj, AvgDB, AvgMB and Total Time refers to the combined values for the original problem.

Figure 5.5 illustrates the result of this algorithm for instance K15C.

Instance	Cluster Sets (Central + Clustered Demand Nodes)	PS Usage	Objective Function Value	Average DB Value	Average MB Value	CPU (sec)
K15E	C0 = { 8 , 13 , 15 , 26 , 29 , 38 , 31 , 23 } C1 = { 8 , 19 , 40 , 27 , 41 , 44 , 2 , 37 }	1:14	ObjC0 = 59 ObjC1 = 81 Total Obj = 140	AvgDB-C0 : 15.99 AvgDB-C1 : 16.27 AvgDB: 16.13	AvgMB-C0 : 853.79 AvgMB-C1 : 842.63 AvgMB : 848.21	TimeC0 =19.468 sec TimeC1 =20.125 sec Total Time = 39 sec
K15N	C0= { 14 , 13 , 15 , 26 , 29 , 38 , 31 , 23 } C1 = { 14 , 19 , 40 , 27 , 41 , 44 , 2 , 37 }	1:13 2:1 (node 41)	ObjC0 = 55 ObjC1 = 84 Total Obj = 139	AvgDB-C0 : 13.8 AvgDB-C1 : 19.66 AvgDB: 16.73	AvgMB-C0 : 881.7 AvgMB-C1 : 641.74 AvgMB : 761.72	TimeC0 =1288.359 sec TimeC1 =30.062 sec Total Time = 1300 sec
K15S	C0= { 18 , 13 , 15 , 26 , 29 , 38 , 31 , 23 } C1 = { 18 , 19 , 40 , 27 , 41 , 44 , 2 , 37 }	1:12 4:2 (node 13 and 19)	ObjC0 = 76 ObjC1 = 63 Total Obj = 139	AvgDB-C0 : 18.66 AvgDB-C1 : 15.5 AvgDB: 17.08	AvgMB-C0 : 747.77 AvgMB-C1 : 725.45 AvgMB : 736.61	TimeC0 = 19.313 sec TimeC1 =14.343 sec Total Time = 23 sec
K15W	C0={ 24 , 13 , 15 , 26 , 29 , 38 , 31 , 23 , } C1 = { 24 , 19 , 40 , 27 , 41 , 44 , 2 , 37 }	1:14	ObjC0 = 81 ObjC1 = 59 Total Obj = 140	AvgDB-C0 : 18.96 AvgDB-C1 : 15.6 AvgDB: 17.28	AvgMB-C0 : 714.29 AvgMB-C1 : 797.99 AvgMB : 756.14	TimeC0 =23.703 sec TimeC1 =27.781 sec Total Time =51 sec
K15C	C0={ 32 , 13 , 15 , 26 , 29 , 38 , 31 , 23 } C1 = { 32 , 19 , 40 , 27 , 41 , 44 , 2 , 37 }	1:14	ObjC0 = 70 ObjC1 = 61 Total Obj = 131	AvgDB-C0 : 17.23 AvgDB-C1 : 16.33 AvgDB: 16.78	AvgMB-C0 : 725.45 AvgMB-C1 : 725.45 AvgMB : 725.45	TimeC0 =17.218 sec TimeC1 = 22.907 sec Total Time = 40 sec

Table 5.8: Clustering Algorithm- Results of instances with 15 node cardinality

Instance	Clusters	PS Usage	Obj Function Value	Average DB Value	Average MB Value	CPU (sec)
K45E	C0 = {8, 7, 38, 6, 26, 3, 29, 15} C1 = {8, 10, 36, 9, 35, 11, 17, 18} C2 = {8, 21, 22, 27, 41, 33, 32, 40} C3 = {8, 5, 34, 31, 12, 37, 28, 39} C4 = {8, 14, 42, 13, 30, 23} C5 = {8, 19, 45, 20, 43, 4, 1, 44} C6 = {8, 2, 24, 25, 16}	1:14 3:2 (node 10 and 2) 2:2 (node 41 and 30)	ObjC0 = 37 ObjC1 = 45 ObjC2 = 55 ObjC3 = 44 ObjC4 = 44 ObjC5 = 71 ObjC6 = 64 Total Obj = 359	AvgDB-C0 : 14.83 AvgDB-C1 : 17.1 AvgDB-C2 : 16.97 AvgDB-C3 : 13.1 AvgDB-C4 : 15.2 AvgDB-C5 : 18.39 AvgDB-C6 : 17.65 AvgDB: 16.17	AvgMB-C0 : 736.61 AvgMB-C1 : 680.8 AvgMB-C2 : 714.29 AvgMB-C3 : 881.7 AvgMB-C4 : 906.25 AvgMB-C5 : 714.29 AvgMB-C6 : 1054.69 AvgMB: 812.66	TimeC0 =11.938 sec TimeC1 =20.265 sec TimeC2 =13.062 sec TimeC3 =240.969 sec TimeC4 =0.938 sec TimeC5 =49.922 sec TimeC6 =0.343 sec Total Time = 345 sec
K45N	C0 = {14, 7, 38, 6, 26, 3, 8} C1 = {14, 10, 36, 9, 35, 11, 17} C2 = {14, 15, 42, 13, 29, 34, 5, 23} C3 = {14, 21, 22, 27, 41, 33, 32, 16} C4 = {14, 19, 45, 20, 43, 40, 18} C5 = {14, 2, 24, 44, 1, 4, 25} C6 = {14, 12, 37, 28, 31, 39, 30}	1:14	ObjC0 = 32 ObjC1 = 48 ObjC2 = 42 ObjC3 = 65 ObjC4 = 56 ObjC5 = 72 ObjC6 = 54 Total Obj = 369	AvgDB-C0 : 13.7 AvgDB-C1 : 15.3 AvgDB-C2 : 13.47 AvgDB-C3 : 16.17 AvgDB-C4 : 16.1 AvgDB-C5 : 17.7 AvgDB-C6 : 15.9 AvgDB: 15.47	AvgMB-C0 : 820.31 AvgMB-C1 : 820.31 AvgMB-C2 : 792.41 AvgMB-C3 : 758.93 AvgMB-C4 : 820.31 AvgMB-C5 : 820.31 AvgMB-C6 : 820.31 AvgMB: 807.55	TimeC0 =5.719 sec TimeC1 =5.688 sec TimeC2 =16.172 sec TimeC3 =27.5 sec TimeC4 =2.438 sec TimeC5 =3.094 sec TimeC6 =2.234 sec Total Time = 60 sec
K45S	C0 = {18, 7, 38, 6, 26, 3, 8} C1 = {18, 10, 36, 9, 35, 11, 17, 28} C2 = {18, 15, 42, 14, 13, 29, 34} C3 = {18, 21, 22, 27, 41, 33, 32, 16} C4 = {18, 19, 45, 20, 43, 40, 12} C5 = {18, 2, 24, 44, 1, 4, 25} C6 = {18, 23, 30, 5, 31, 39, 37}	1:13 2:1 (node 41)	ObjC0 = 51 ObjC1 = 28 ObjC2 = 53 ObjC3 = 44 ObjC4 = 30 ObjC5 = 61 ObjC6 = 54 Total Obj = 321	AvgDB-C0 : 15.6 AvgDB-C1 : 13.46 AvgDB-C2 : 15.8 AvgDB-C3 : 15.87 AvgDB-C4 : 13.5 AvgDB-C5 : 16.6 AvgDB-C6 : 15.9 AvgDB: 15.24	AvgMB-C0 : 820.31 AvgMB-C1 : 786.83 AvgMB-C2 : 820.31 AvgMB-C3 : 708.71 AvgMB-C4 : 820.31 AvgMB-C5 : 820.31 AvgMB-C6 : 820.31 AvgMB: 799.58	TimeC0 =6.578 sec TimeC1 =1142.469 sec TimeC2 =3.547 sec TimeC3 =45.875 sec TimeC4 = 3.219 sec TimeC5 =7.765 sec TimeC6 =2.344 sec Total Time = 1207 sec
K45W	C0 = {24, 7, 38, 6, 26, 3, 8} C1 = {24, 10, 36, 9, 35, 11, 17} C2 = {24, 15, 42, 14, 13, 29, 34} C3 = {24, 21, 22, 27, 41, 33, 32, 4} C4 = {24, 19, 45, 20, 43, 40, 18, 16} C5 = {24, 12, 37, 28, 31, 5, 39} C6 = {24, 23, 30, 25, 44, 2, 1}	1:14	ObjC0 = 66 ObjC1 = 44 ObjC2 = 64 ObjC3 = 40 ObjC4 = 54 ObjC5 = 61 ObjC6 = 45 Total Obj = 374	AvgDB-C0 : 17.1 AvgDB-C1 : 14.9 AvgDB-C2 : 16.9 AvgDB-C3 : 15.27 AvgDB-C4 : 16.69 AvgDB-C5 : 16.6 AvgDB-C6 : 15 AvgDB: 16.06	AvgMB-C0 : 820.31 AvgMB-C1 : 820.31 AvgMB-C2 : 820.31 AvgMB-C3 : 714.29 AvgMB-C4 : 714.29 AvgMB-C5 : 820.31 AvgMB-C6 : 820.31 AvgMB: 790.01	TimeC0 =9.547 sec TimeC1 =10.719 sec TimeC2 =6.516 sec TimeC3 =50.578 sec TimeC4 =34.844 sec TimeC5 =2.422 sec TimeC6 =1.766 sec Total Time = 115 sec
K45C	C0 = {32, 7, 38, 6, 26, 3, 8} C1 = {32, 10, 36, 9, 35, 11, 17, 28} C2 = {32, 15, 42, 14, 13, 29, 34} C3 = {32, 21, 22, 27, 41, 33, 40, 16} C4 = {32, 19, 45, 20, 43, 18, 12} C5 = {32, 2, 24, 44, 1, 4, 25} C6 = {32, 23, 30, 5, 31, 39, 37}	1:13 2:1 (node 9)	ObjC0 = 51 ObjC1 = 32 ObjC2 = 51 ObjC3 = 31 ObjC4 = 33 ObjC5 = 49 ObjC6 = 49 Total Obj = 296	AvgDB-C0 : 15.6 AvgDB-C1 : 14.5 AvgDB-C2 : 15.6 AvgDB-C3 : 13.16 AvgDB-C4 : 13.8 AvgDB-C5 : 15.4 AvgDB-C6 : 15.4 AvgDB: 14.78	AvgMB-C0 : 820.31 AvgMB-C1 : 703.12 AvgMB-C2 : 820.31 AvgMB-C3 : 770.06 AvgMB-C4 : 820.31 AvgMB-C5 : 820.31 AvgMB-C6 : 820.31 AvgMB: 796.39	TimeC0 =5.875 sec TimeC1 =31.937 sec TimeC2 =1.875 sec TimeC3 =46.922 sec TimeC4 =3.36 sec TimeC5 =3.531 sec TimeC6 =2.843 sec Total Time = 80 sec

Table 5.9: Clustering Algorithm- Results of instances with 45 node cardinality

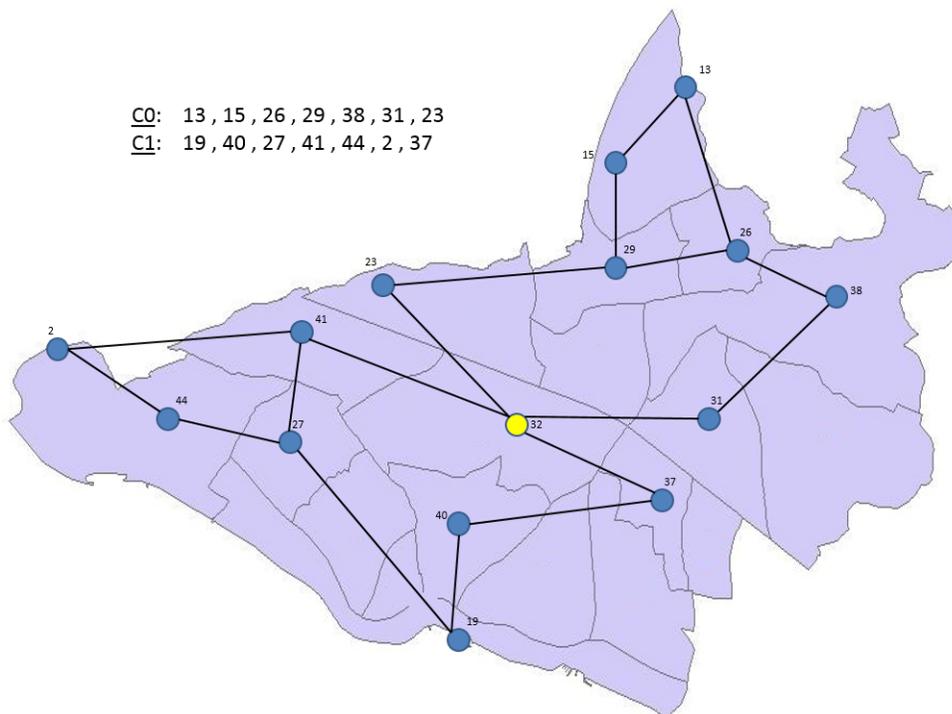


Figure 5.5: Clustering Algorithm- Output of instance K15C

### 5.2.5 Performance of Heuristic Algorithms

Based on the outputs of the mathematical model and heuristic algorithms, we compare and evaluate the performances of them. Table 5.10 illustrates the objective function and quality of service values for instances with 8 and 15 node cardinality. The percentage comparison can be seen in Table 5.11. As mentioned before, for the full Kartal set, we cannot receive any solution (even a best integer value) through the mathematical model. Hence we compare the results of the heuristic algorithms for the instances with 45 node cardinality. It is represented in Table 5.12.

As mentioned before, Clustering Algorithm only generates 1 cluster for the K8E, K8N, K8S, K8W and K8C instances. Hence, the output of the clustering algorithm corresponds to the optimal result. That is why the optimality gap of “Clustering Algorithm” is zero for these instances in Table 5.12.

As stated before, for the MIP start values, we supply the results of the “Ring Creation Algorithm” to CPLEX. However, the model could not obtain a better objective function value than the MIP start for K15E, K15W and K15C instances in 4 hours. However, the decision variables are different for these instances since the corresponding service quality values are different.

Based on the computational study, we can clearly see that Clustering Algorithm outperforms Ring Creation Algorithm in terms of objective function values for every instance. The reason for this situation is that Ring Creation Algorithm installs the fiber wires based on a sorted demand node set in a greedy approach while Clustering Algorithm finds the optimal fiber wire links for each cluster. Another reason is that, maintaining service to the nonPS nodes causes a considerable amount of cost increase in Ring Creation Algorithm.

However, Ring Creation Algorithm provides higher quality of service levels (ie, low insertion loss and high speed amount). The reason for this situation is mentioned in Chapter 4. Since we aim to ensure feasibility by maintaining high quality service for nonPS nodes, the resulting network will have high quality of service

values.

We can see that the solution time and good objective function values are clashing factors. Ring Creation Algorithm obtains a result in milisecond whereas Clustering Algorithm requires more time and effort.

Instance	OPTIMAL —Best Integer	OPTIMAL – Best Bound	RING CREATION-Objective Function Value	CLUSTER-Objective Function Value	OPTIMAL-Average DB Value(Best Integer)	RING CREATION-Average DB Value	CLUSTER-Average DB Value	OPTIMAL-Average MB Value (Best Integer)	RING CREATION-Average MB Value	CLUSTER-Average MB Value
K8E	84	84	112	84	20.6	16.39	20.6	680.8	970.98	680.8
K8N	81	81	111	81	16.21	16.34	16.21	881.7	970.98	881.7
K8S	82	82	99	82	18.24	16.31	18.24	753.35	970.98	753.35
K8W	78	78	102	78	18.66	16	18.66	714.29	970.98	714.29
K8C	84	84	87	84	18.44	15.56	18.44	747.77	970.98	747.77
K15E	(166)	(100)	166	140	15.69	15.04	16.13	926.34	970.98	848.21
K15N	(154)	(98)	173	139	14.92	15.34	16.73	934.71	970.98	761.72
K15S	(156)	(99)	174	139	16.64	15.84	17.08	703.12	970.98	736.61
K15W	(180)	(96)	180	140	16.43	15.79	17.28	926.34	970.98	756.14
K15C	(163)	(100)	163	131	16.33	15.69	16.78	926.34	970.98	725.45

Table 5.10: Comparison for Instances with 8 and 15 node cardinality

Instance	Gap of RING CREATION	Gap of Cluster Algorithm
K8E	33.33%	0.00%
K8N	37.04%	0.00%
K8S	20.73%	0.00%
K8W	30.77%	0.00%
K8C	3.57%	0.00%
K15E	66.00%	40.00%
K15N	76.53%	41.84%
K15S	75.76%	40.40%
K15W	87.50%	45.83%
K15C	63.00%	31.00%

Table 5.11: Percentage Comparison for Instances with 8 and 15 node cardinality

Instance	CLUSTER-Objective Function Value	RING-Objective Function Value	Obj Value Difference in (%)	CLUSTER-Average DB Value	RING-Average DB Value	DB Value Difference in (%)	CLUSTER-Average MB Value	RING-Average MB Value	MB Value Difference in (%)
K45E	359	412	14.76%	16.17	15.47	-4.33%	812.66	868.25	6.84%
K45N	369	434	17.62%	15.47	15.7	1.49%	807.55	868.25	7.52%
K45S	321	364	13.40%	15.24	14.82	-2.76%	799.58	868.25	8.59%
K45W	374	466	24.60%	16.06	16.07	0.06%	790.01	868.25	9.90%
K45C	296	364	22.97%	14.78	14.77	-0.07%	796.39	868.25	9.02%

Table 5.12: Comparison for Instances with 45 node cardinality

# Chapter 6

## Conclusion

In this problem, we studied a survivable fiber optical network design problem. This problem corresponds to the survivable extension of previously defined “Green Field Network Design Problem”. Our aim is to design a cost effective fiber optical network that will provide high quality Internet access from a prelocated central station to a set of demand nodes via 2-node disjoint paths. The access from central station to the demand nodes can be provided either directly by installing fibers or indirectly by utilizing special telecommunication devices called Passive Splitters. Installing passive splitters to the network is regarded from a hub location perspective. The quality of service requirements originate from real world application dynamics. Hence the problem structure is interesting.

For higher dimensions, it became difficult to obtain a solution in terms of CPU. We developed two heuristics which provide feasible solutions in a fast manner. In the computational study, we evaluated the performances of the heuristic algorithms along with some analysis. We also utilized a special feature, MIP Start, in order to attain better solution quality. Based on different problem instances, we analyzed the effect of the central station location.

A possible future direction might be developing improvement heuristic algorithms

for the problem in order to increase the solution quality of the constructive heuristics. Also, valid inequalities strengthening the mathematical model could be added.

# Bibliography

- [1] A. Faye, M. Trampont, M. Chardy, and M. C. Costa, “Optimizing splitter and fiber location in a multilevel optical ftth network,” *European Journal of Operational Research*, 2012.
- [2] E. Gourdin, M. Labbe, and H. Yaman, “Telecommunication and location,” *Drezner Z, Hamacher H, eds. Facility Location: Applications and Theory*, pp. 275–205, 2005.
- [3] D. E. Comer, *Computer Networks and Internets: With Internet Applications*. No. v. 1 in *Computer Networks and Internets: With Internet Applications*, Pearson Education, 5th ed., 2004.
- [4] K. Pahlavan and P. Krishnamurthy, *Principles of Wireless Networks: A Unified Approach*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1st ed., 2001.
- [5] B. Yazar, “Fiber Optical Network Design Problems: Case for Turkey,” Master’s thesis, Bilkent University, Ankara, 2013.
- [6] J. Campbell and M. O’Kelly, “Twenty-five years of hub location research,” *Transportation Science*, vol. 46, no. 2, pp. 153–169, 2012.
- [7] M. O’Kelly, “The location of interacting hub facilities,” *Transportation Science*, vol. 20, no. 2, pp. 92–105, 1986.
- [8] J. Campbell, “Integer programming formulations of discrete hub location problems,” *European Journal of Operational Research*, vol. 72, pp. 387–405, 1994.

- [9] J. Campbell, A. Ernst, and M. Krishnamoorthy, “Hub arc location problems: Part 1 - Introduction and Results,” *Management Science*, vol. 51, no. 10, pp. 1540–1555, 2005.
- [10] S. Alumur and B. Kara, “Hub location problems: The state of the art,” *European Journal of Operational Research*, vol. 190, no. 1, pp. 1–21, 2008.
- [11] J. Campbell, A. Ernst, and M. Krishnamoorthy, “Hub location problems,” *Facility Location: Applications and Theory (Springer, Berlin)*, pp. 373–407, 2008.
- [12] J. Klincewicz, “Hub location in backbone/tributary network design: a review,” *Location Science*, vol. 6, pp. 307–335, 1998.
- [13] S. Van Hoesel, “Optimization in telecommunication networks,” *Statistica Neerlandica*, vol. 59, no. 2, pp. 180–205, 2004.
- [14] G. Park, Y. Lee, and J. Han, “A two-level location-allocation problem in designing local access fiber optic networks,” *Computers and Operations Research*, vol. 51, pp. 52–63, Nov. 2014.
- [15] Y. Kim, Y. Lee, and J. Han, “A splitter location-allocation problem in designing fiber optic access networks,” *European Journal of Operational Research*, vol. 210, pp. 425–435, 2011.
- [16] H. Kim and M. O’Kelly, “Reliable p-hub location problems in telecommunication networks,” *Geographical Analysis*, pp. 283–306, 2009.
- [17] B. Zeng, Y. An, Y. Zhang, and H. Kim, “A reliable hub-spoke model in transportation systems,” *Proceedings of the 4th International Symposium on Transportation Network Reliability (INSTR) Conference Proceedings, Minneapolis, Minnesota, USA*, 2010.
- [18] J. Suurballe, “Disjoint paths in a network,” *Networks*, vol. 4, pp. 125–145, 1974.
- [19] J. Suurballe and R. Tarjan, “A quick method for finding shortest pairs of disjoint paths,” *Networks*, vol. 14, pp. 325–336, 1984.

- [20] P. Damcı, “Algorithms For The Survivable Telecommunications Network Design Problem Under Dedicated Protection,” Master’s thesis, Bilkent University, Ankara, 2010.
- [21] B. Yıldız and O. E. Karaşan, “Regenerator location problem and survivable extensions: A hub covering location perspective,” *Transportation Research Part B*, pp. 32–55, 2015.
- [22] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988.
- [23] F. Kılıcı and B. Kara, “A decision support system for shelter site selection with Gis integration: Case for Turkey,” *European Journal of Operational Research (To Appear)*, 2015.