# A TWO-LEVEL NETWORK DESIGN PROBLEM WITH DECENTRALIZED AND CENTRALIZED INTERMEDIATE FACILITIES

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

Emirhan Buğday

September 2017

A two-level network design problem with decentralized and centralized intermediate facilities
By Emirhan Buğday
September 2017

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.

Ayşe Selin Kocaman(Advisor)

Kağan Gökbayrak (Co-advisor)

Oya Karaşan

İmdat Kara

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

# ABSTRACT

# A TWO-LEVEL NETWORK DESIGN PROBLEM WITH DECENTRALIZED AND CENTRALIZED INTERMEDIATE FACILITIES

Emirhan Buğday
M.S. in Industrial Engineering
Advisor: Ayşe Selin Kocaman
Co-advisor: Kağan Gökbayrak
September 2017

Within the framework of the Two-Level Network Design Problems (TLND), we propose a hybrid network design problem that consists of both decentralized and centralized intermediate facilities for resource distribution systems. Single-commodity flow based and multi-commodity flow based Mixed Integer Linear Programs (MILP) are developed for the problem. In addition, a heuristic algorithm that designs the lower-level and the upper-level networks sequentially is proposed as a solution method. We perform computational experiments with real and random spatial distributions to demonstrate the performance of the heuristic method. Wide ranges of cost parameter combinations are used and it is observed that for some parameter combinations hybrid networks may be the least cost design compared to centralized or decentralized networks.

*Keywords:* Two-Level Network Design, Hierarchical Facility Location, Hybrid Network, Centralized Facilities, Decentralized Facilities, Heuristic.

# ÖZET

## MERKEZİ VE MERKEZİ OLMAYAN TESİSLER KULLANARAK İKİ SEVİYE DAĞITIM AĞI TASARLAMA PROBLEMİ

Emirhan Buğday
Endüstri Mühendisliği, Yüksek Lisans
Tez Danışmanı: Ayşe Selin Kocaman
Eş-Tez Danışmanı: Kağan Gökbayrak
Eylül 2017

Bu çalışmada iki seviye dağıtım ağı tasarım problemi çerçevesinde, merkezi ve merkezi olmayan tesislerden oluşan melez ağların, kaynak dağıtım sistemleri için daha düşük maliyetli sonuçlar vereceği önerilmiştir. Problem için tek madde akış bazlı ve çoklu madde akış bazlı iki adet karışık tamsayı programı oluşturulmuştur. Ayrıca, birinci ve ikinci seviye ağları sırayla tasarlayacak bir sezgisel yöntem geliştirilmiştir. Farklı büyüklüklerde ve farklı uzaysal dağılım desenlerine sahip çeşitli örneklerle, sezgisel yöntemin performansı analiz edilmiş ve karışık tamsayı programları ile kıyaslanmştır. Ayrıca, geniş yelpazede fiyat parametreleri kullanılarak, melez ağların, tek başına merkezi veya merkezi olmayan ağlardan iyi sonuçlar vereceği örnekler olduğu da gösterilmiştir.

*Anahtar sözcükler*: İki Seviye Dağıtım Ağı Tasarımı, Melez Ağ, Merkezi Tesis, Merkezi Olmayan Tesis, Sezgisel.

*To my parents...*

# Acknowledgement

I would like to express my most sincere gratitude to Asst. Prof. Ayşe Selin Kocaman and Asst. Prof. Kağan Gökbayrak for all their support and kindness throughout my research journey.

I would also like to thank Prof. Dr. Oya Karaan and Prof Dr. İmdat Kara for devoting their valuable time to read and review my thesis and their substantial comments.

I am deeply grateful to my mother, Neslihan, my father, Mehmet, my brother, Oğuzhan and my grandma, Ayten for their unbending love and their endless support. I can not thank them enough for their patience and caring towards me.

I would also like to express my sincere gratitude to my friends. Many thanks to my dear friend and my lifetime classmate Onur. Special thanks to Başak, Beyza, Erman and Kübra for their support and cheeriness during the most crucial times of my graduate study and many more friends that I failed to mention here.

The last but not the least, I would like to express my deepest gratitude to my fellow diver friends Güz, Ata and Can Ozan that I am deeply bonded with and my love to the Bilkent University Underwater Society members for all good the good memories.

# Contents


**1 Introduction** **1**

**2 Literature Review** **4**

2.1 Centralized Network Design Problems . . . . . . . . . . . . . . . . . 5

2.2 Decentralized Network Design Problems . . . . . . . . . . . . . . . 7

**3 Problem Definition and Formulation** **9**

3.1 Single-Commodity Flow Based Formulation . . . . . . . . . . . . . 11

3.2 Multi-Commodity Flow Based Formulation . . . . . . . . . . . . . . 14

3.3 Comparison of the Formulations . . . . . . . . . . . . . . . . . . . . 16

3.4 Optimality Cuts and Valid Inequalities . . . . . . . . . . . . . . . . 21

**4 Solution Approach** **25**

4.1 Subproblems Solved in the Heuristic Algorithm . . . . . . . . . . . 27

4.1.1 Simple Plant Location Problem . . . . . . . . . . . . . . . . 27

4.1.2 p-Median Problem . . . . . . . . . . . . . . . . . . . . . . . . 28

4.1.3 Prize Collecting Steiner Tree Problem . . . . . . . . . . . . . 29

4.2 Heuristic Algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . . 31

**5 Numerical Analysis 34**

5.1 Experiments with Random Data . . . . . . . . . . . . . . . . . . . . 35

5.2 Experiments with Data from Network Planner . . . . . . . . . . . . 46

5.3 Iteration Performance of the Heuristic Algorithm . . . . . . . . . . 58

**6 Conclusion 62**

**A RANDOM DATA RESULTS 70**

**B ITERATION AND CPU TIME BREAKDOWN OF THE HEURISTIC ALGORITHM 75**

**C HEURISTIC ALGORITHM vs SPLP COMPARISON 77**

# List of Figures


2.1 Literature Tree . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

3.1 Maps of the Randomly Generated Data . . . . . . . . . . . . . . . 17

4.1 Flow Chart of the Heuristic Algorithm . . . . . . . . . . . . . . . . 26

5.1 Random Data for the Numerical Analysis . . . . . . . . . . . . . . 35

5.2 Comparison of the Heuristic and Optimal Solution for Centralized Solutions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37

5.3 Comparison of the Heuristic and Optimal Solution for Hybrid Solutions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

5.4 Comparison of Heuristic with Optimal for Decentralized solutions 40

5.5 Results of the Type III Random Data . . . . . . . . . . . . . . . . 43

5.6 Results of the Type I Random Data . . . . . . . . . . . . . . . . . 45

5.7 Maps Obtained from Network Planner . . . . . . . . . . . . . . . . 47

5.8 Solution Instances of *LeonaG* Experiment . . . . . . . . . . . . . 51

5.9 Solution Instances of *BOL* Experiment . . . . . . . . . . . . . . . . 53

5.10 Solution Instances of *Guinea* Experiment . . . . . . . . . . . . . . 54

5.11 Solution Instances of *LGA* Experiment . . . . . . . . . . . . . . . . 56

5.12 Iteration Patterns of the *LeonaG* Experiment . . . . . . . . . . . . 59

5.13 Iteration Patterns of the *BOL* Experiment . . . . . . . . . . . . . 60

A.1 Results of the Type II Random Data . . . . . . . . . . . . . . . . . 71

A.2 Results of the Type I Random Data . . . . . . . . . . . . . . . . . . 72

A.3 Results of the Type II Random Data . . . . . . . . . . . . . . . . . 73

A.4 Results of the Type III Random Data . . . . . . . . . . . . . . . . . 74

# List of Tables

3.1 Numerical Analysis of the Comparison of the Single-Commodity Flow Formulation and the Multi-Commodity Flow Formulation . 19

3.2 LP Relaxations of 2LDN and 2LDN-MCF . . . . . . . . . . . . . . 20

3.3 Summary of the Optimality Cut/Valid Inequality Experiment . . 22

3.4 Numerical Analysis of the Optimality Cut/Valid Inequality Combinations . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

3.5 LP Relaxation of the Optimality Cut/Valid Inequality Combinations 24

5.1 Numerical Analysis of the Heuristic Algorithm with Random Data 38

5.2 Numerical Analysis of the Modified and the Extended Heuristic Algorithms . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42

5.3 Parameter Set . . . . . . . . . . . . . . . . . . . . . . . . . . 48

5.4 Results of the Experiment with *LeonaG* . . . . . . . . . . . . . . . 50

5.5 Results of the Experiment with *BOL* . . . . . . . . . . . . . . . . . 52

5.6 Results of the Experiment with *Guinea* . . . . . . . . . . . . . . . . 55

5.7 Results of the Experiment with *LGA* . . . . . . . . . . . . . . . . . 57

B.1 Iteration of the Experiments . . . . . . . . . . . . . . . . . . . . . . . 75

B.2 CPU Time Breakdown of the Experiments . . . . . . . . . . . . . . 76

C.1 Comparison of the Heuristic Algorithm with SPLP . . . . . . . . . 78

# Chapter 1

# Introduction

Many source location and allocation problems have been studied to help decision makers in strategic planning for sustainable development. There exist various forms of location problems covered by several studies. Although the context of these problems may vary, the main objective of these problems is to determine locations of *facilities* subject to different constraints. The term *facility* stands for entities, such as energy distribution centers, warehouses, schools, hospitals etc. that serve *customers/demand points*. The problems that assume a discrete set candidate locations for facilities are called the Discrete Location Problems. In general, two types of costs are associated with Discrete Location Problems: *facility opening/service cost* and *connection cost* [1].

Furthermore, location problems that have two or more levels of facilities are classified as Hierarchical Location Problems. The Hierarchical Location Problems aim to design centralized distribution systems where the demand of a customer is satisfied from a source through intermediate facilities. These kind of networks have at least two levels of connections: one is in between the demand nodes and the facilities, the other one is in between the facilities and the source [2]. Networks with two levels are called two-level or two-echelon networks where the lower-level is mentioned as secondary or local access network and the upper-level is mentioned as primary or backbone network in the literature.

As an alternative to centralized distribution systems, it is possible to satisfy the demand of the customers directly from a source facility that is connected to them. Such networks have only one level of connection and we refer to these systems as *decentralized distribution systems*.

Designing decentralized systems is widely studied in the last 150 years [3]. On the other hand, designing centralized systems have become popular in the last two decades [4]. Recently, there are studies published about preferring decentralized or centralized networks over the other, especially in designing infrastructures [5], power distribution systems [6, 7], supply chain [8, 9], telecommunications [10], and urban networks [11]. However, to the best of our knowledge there is no study that has focused on designing networks consisting of both centralized and decentralized systems in the literature.

We introduce a new problem to the literature that is designing hierarchical networks utilizing both decentralized and centralized intermediate facilities. A *decentralized* facility serves customers as a stand-alone source point. On the other hand, a centralized facility has to be connected to an additional source to serve customers. We refer these networks that contain both centralized and decentralized facilities as hybrid networks.

A decentralized facility is more expensive than a centralized facility as it can act as a stand-alone source point. The centralized, on the other hand, needs a connection to a source directly or over another centralized facility, which is another cost component.

For any given set of demand points, we search for the least cost network design utilizing both decentralized and centralized intermediate facilities. When the demand points form clusters that are away from each other a decentralized network may be the least cost solution. For demand points that are close to each other, however, a centralized network may be preferable. There may be some cases when hybrid networks provide the least cost solution. Thus, we study the least-cost design of two level networks with both decentralized and centralized intermediate facilities in this thesis.

The rest of the thesis is organized as follows: In Chapter 2, a literature review regarding centralized and decentralized systems and related problems to the defined problem is given. We also provide the position of the defined problem in the existing literature. Chapter 3 introduces the problem specific requirements in detail and presents two mixed integer linear programming (MILP) formulations of the introduced problem. Valid inequalities are proposed and tested to obtain a better formulation. In Chapter 4, a heuristic algorithm is proposed as an alternative solution method to the introduced problem. In Chapter 5, the performance of the proposed heuristic algorithm is analyzed. Computational studies are carried out with both synthetic and real data. Heuristic algorithm is compared with the solution of the MILP formulation and one other widely studied integer linear programming formulation from the literature. In addition, sensitivity analysis is conducted to understand the dynamics of the problem. Finally, the summary of the results are provided and discussed in Chapter 6.

# Chapter 2

# Literature Review

We conduct our literature review under the centralized network design problems (CNDPs) and decentralized network design problems (DNDPs) due to the fact that we are working on hybrid systems which involve both centralized and decentralized facilities .

CNDPs refer to the problems that aim to design efficient and effective networks that connect customers to a source point through intermediate facilities. On the other hand, DNDPs refer to the single-level facility location problems. In Figure 2.1 the positioning of the above mentioned categories and the introduced problem in the Discrete Location Problems literature is presented.

First, we provide our literature review regarding the CNDPs and then continue with the DNDPs.

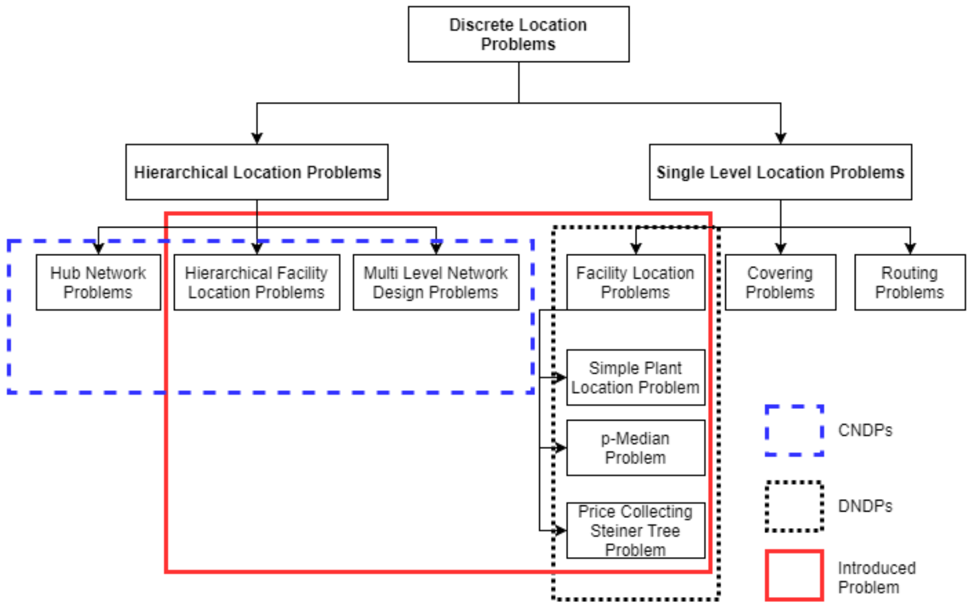



Figure 2.1: Literature Tree

## 2.1 Centralized Network Design Problems

Balakrishnan [12] claims that the motivation of having intermediate facilities is to allow flow between source and demand nodes at a lower cost compared to the single level location/network design problems.

Several topological designs are used in CDNPs such as complete, mesh, tree, ring, path and star. In a complete network, which is also referred as fully interconnected or full mesh network, each node is connected to every other node. Mesh network is a complete network with one or more missing links. In tree networks, nodes are connected like the branches of a tree. Ring network is a structure that forms a cycle and flow between nodes can occur in two directions. Path is a network type where the nodes linked one by one create a single branch. And finally, if each node is directly connected to a single node, that is called a star network. Since hierarchical systems have multiple levels of networks, each level can have a different configuration [13].

CNDPs are studied in the literature under different categories using different terminologies. Although they all try to design a layered centralized system with interacting facilities, they differ from each other due to the variations in their application areas. The three main categories of the CNDPs and their major differences are listed below:

- **Multi-Level Network Design Problems (MLNDPs)** focus on the topological structure of the network, and has been originally defined by Balakrishnan [12]. MLNDPs are mostly studied in the context of telecommunication networks [14, 15], fiber optic networks [16, 17], transportation networks [18], and electric power distribution networks [19, 20]
- **Hierarchical Facility Location Problems (HFLPs)** focus on dealing with the special requirements of wide application areas such as supply chain management [21], health care systems [22, 23, 24], and solid waste management systems [25]. HFLPs mostly have a star-star topology [26].
- **Hub Network Problems (HNPs)**, are another network design problem classification that also focuses on different topological designs. the primary characteristic of HNP is not having a source. Rather than connecting to a source, the aim is to connect all demand nodes to each other through intermediate facilities with different hierarchies [27]. Since HNPs do not have any source stations, they are out of the context of this thesis.

A special case of the MLNDPs including only two levels of networks are called Two Level Network Design Problems (TLNDPs). In the TLNDPs, demand nodes are connected to intermediate facilities in the first level, which is called as the local access network and intermediate facilities are connected to a source node in the second level which is called as the backbone network. Without the centralized facilities, our problem reduces to a TLDNP. Various studies focus on TLDNP with different topological designs.

Lee et al. [28] focused on a two-level network design problem with a ring-star configuration. In the study, connections established between the facilities form a ring while the connections between the demand nodes and the facilities have a star configuration. In [22], Chung et al. studied a mesh-star configuration. Since we use a tree-star topology in our problem, Kim et al. [29] has the most similar structure to our work from the centralized system perspective. They developed a dual based integer programing model to design a two level hierarchical network that has a tree configuration in the upper-level network and star configuration in the lower-level network while optimizing the total facility opening and total connection costs. Their novelty to the literature is allowing junction points that enables upper-level connections through the facilities which are not chosen to be opened. They proposed a branch and bound solution method to solve.

Another related problem is the Multi-level Uncapacitated Facility Location Problem (MLUFLP), which is a version of HFLP that the centralized system is designed as a star-star topology. In MLUFLP, the aim is to assign each demand node to an intermediate facility and each intermediate facility to a source [30, 31].

## 2.2 Decentralized Network Design Problems

In this section three related DCNP is provided: p-Median Problem (PMP), Simple Plant Location Problem (SPLP), and Price Collecting Steiner Tree Problem (PCSTP).

The first decentralized network design problem that is related our problem is the p-Median Problem (PMP), which was originally defined by Hakimi [32] and the classical formulation of PMP is developed by ReVelle and Swain [33]. The objective is to find the minimum total connection cost while serving all demand nodes with a predetermined number of facilities, which is denoted by $p$. In the solution, each facility has a cluster of demand nodes. In PMP, two types of decisions arise: deciding facility locations and the connections between the customers and facilities. PMP is an NP-hard problem [34]. The PMP is further studied by Rosing et al. [35], Church [36], Cornuejols et al. [37] and Elloumi [38].

The second related problem is the Simple Plant Location Problem (SPLP) which is introduced as the Warehouse House Location Problem by Kuehn [39]. SPLP that is also referred as Uncapacitated Facility Location (UFLP) in the literature, is a vastly studied problem and dates back to the works of Balinski [40], Manne [41] and Stollsteimer [42]. The problem aims to reduce the transportation and warehouse opening costs of a network via deciding warehouse locations. Note that that number of facilities is not fixed in this problem. The problem defined in this thesis can be reduced to SPLP when centralized facilities are not allowed and all facilities are selected as decentralized. If, in addition, the total number of decentralized facilities is given as an input, then our problem reduces to the PMP.

The third related problem is an extension of the Steiner Tree Problem (STP) that is called Price Collecting Steiner Tree (PCST). In STP, the aim is constructing a tree in order to span a specified subset of the vertices of a given undirected graph [43] and it is known to be NP-hard [44]. Two sets of demand nodes are given: Terminal and non-Terminal. Terminal nodes which are also called as Steiner nodes has to be included in the subtree, while non-Terminal nodes can be omitted. On the other hand, PCST aims to find a subtree while maximizing the total prize collected from the selected vertices after paying the edge costs of the established connections [45]. The problem is also studied by Bienstock et al. [46] with an objective of minimizing the sum of the weights of the subtree edges plus the prices of the nodes not spanned in a given undirected graph. This version of the problem is referred as Goemans and Williamson Minimization Problem (GWMP) in the literature. Bienstock et al. developed the first approximation algorithm with an approximation guarantee of 3. Then, Goemans and Williamson improved the approximation guarantee to below 2 [47, 48].

The upper-level network design of the introduced problem can be interpreted as GWMP when the total number of facilities and their locations are fixed and prices are defined as the difference of the opening costs of the decentralized and centralized facilities. This transition is covered in detail in Chapter 4.1.3.

# Chapter 3

# Problem Definition and Formulation

Consider an with no distribution network at all but a source station available. Our aim is to design the least cost hybrid distribution system for the given greenfield area. In this problem, we assume that the set of the candidate locations for opening facilities is the same as the given locations of the consumers. Locations of the consumers are called demand nodes in the rest of the paper. All demand nodes have to be directly connected to a facility. We assume that each demand node is served by a single facility.

There are two types of facilities: centralized facilities and decentralized facilities. Both types of the facilities are assumed to have an unlimited capacity in terms of the total number of demand nodes they can serve. Decentralized facilities are more expensive than the centralized ones because they are stand alone sources and can serve the demand nodes that are connected to them without being supplied from a source. On the other hand, centralized facilities do not have such capabilities; thus, they have to be connected to a source station. A demand node which is connected to a centralized facility is served from the source station thorough that centralized facility. However, the source station cannot serve consumers directly.

Centralized networks, which are constituted by centralized facilities, have two levels. The first level is between the demand nodes and the centralized facilities while the second level is between the centralized facilities and the source node. On the other hand, decentralized networks, which do not contain any centralized facilities, have only one level. Nevertheless, in both cases, connection types between the demand nodes and the facilities are the same. They have the same connection cost and star topology. Thus, we refer the network that consists of all connections between the demand nodes and the facilities as the lower-level network. Besides, we call the connections made between the source station and the centralized facilities as the upper-level network.

Upper network connections and lower-level network connections can be differentiated from each other in various aspects. While lower-level network has a star configuration, which means each node has to be connected directly to a facility, the upper-level network has a tree configuration which means the connections between the centralized facilities and the source node has to form a tree. Moreover, the upper-level network connection cost per meter is higher than the lower-level network connection cost per meter.

We propose two mixed integer programming formulations for the specified problem. While the first one is a single-commodity flow based formulation, the other is a multicommodity flow based formulation. First, we provide the formulations and then compare their performances.

## 3.1 Single-Commodity Flow Based Formulation

Consider a given area with $n$ demand nodes and a source station. The locations of the demand nodes and the source station can be provided either in latitude and longitude format, which is the exact coordinates of the nodes on the Earth or as $(x, y)$ coordinates on a rectangular Euclidean space. Let $N$ be the set of demand nodes from 1 to $n$. Let the source station be the node 0, which will be mentioned as the source node in the rest of the chapter, and $N^0$ be the set of nodes, including the source node, from 0 to $n$. The parameters are defined as the following:

- the cost of opening a centralized facility, $C_{CF}$
- the cost of opening a decentralized facility, $C_{DF}$
- the unit cost of lower-level network connection, $C_{LN}$
- the unit cost of upper-level network connection, $C_{UN}$
- the distance between node $i \in \{0, \ldots, n\}$ and node $j \in \{0, \ldots, n\}$, $D_{ij}$

Note that, if the locations of demand nodes and the source node are given in latitude and longitude format, we calculate the distances, $D_{ij}$, according to the great-circle distance formula [49], otherwise we use the Euclidean distances.

To minimize the total system cost, our objective is to determine:

- The number, the location and the type of the facilities.
- The lower-level network, the assignment of demand nodes to facilities.
- The upper-level network, the tree constructed between centralized facilities and source node, if at least one centralized facility is opened.

In order to accomplish that we define following decision variables:

$$x_j = \begin{cases} 1 & \text{if a centralized facility is opened at node } j\text{, where } j \in N, \\ 0 & \text{otherwise.} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{if a decentralized facility is opened at node } j\text{, where } j \in N, \\ 0 & \text{otherwise.} \end{cases}$$

$$z_{ij} = \begin{cases} 1 & \text{if demand node } i \text{ is served by facility } j\text{, where } i \in N \text{ and } j \in N, \\ 0 & \text{otherwise.} \end{cases}$$

$$u_{jk} = \begin{cases} 1 & \text{if there exists an upper-level network connection between node } j \\ & \text{and node } k\text{, where } j \in N^0 \text{ and } k \in N, \\ 0 & \text{otherwise.} \end{cases}$$

$f_{jk} =$ flow from $j^{th}$ facility/source to $k^{th}$ facility, where $j \in N^0$ and $k \in N$.

We propose the following mixed integer optimization problem as a single-commodity flow based formulation, denoted by 2LND, for the defined problem:

$$min \quad \sum_{j=1}^{n} x_j C_{CF} + \sum_{j=1}^{n} y_j C_{DF} + \sum_{i=1}^{n}\sum_{j=1}^{n} z_{ij} D_{ij} C_{LN} + \sum_{j=0}^{n}\sum_{k=1}^{n} u_{jk} D_{jk} C_{UN} \tag{3.1}$$

$s.t.$

$$\sum_{j=1}^{n} z_{ij} = 1 \qquad \forall i \in N \tag{3.2}$$

$$z_{ij} \leq x_j + y_j \qquad \forall i \in N,\ \forall j \in N \tag{3.3}$$

$$x_j + \sum_{k=1,k\neq j}^{n} f_{jk} = \sum_{l=0,l\neq j}^{n} f_{lj} \qquad \forall j \in N \tag{3.4}$$

$$f_{jk} \leq n u_{jk} \qquad \forall j \in N^0,\ \forall k \in N \tag{3.5}$$

$$2u_{jk} \leq x_j + x_k \qquad \forall j \in N,\ \forall k \in N \tag{3.6}$$

$$x_j \in \{0,1\} \qquad \forall j \in N^0 \tag{3.7}$$

$$y_j \in \{0,1\} \qquad \forall j \in N \tag{3.8}$$

$$z_{ij} \in \{0,1\} \qquad \forall i \in N,\ \forall j \in N \tag{3.9}$$

$$u_{jk} \in \{0,1\} \qquad \forall j \in N^0,\ \forall k \in N \tag{3.10}$$

$$f_{jk} \in \mathbb{Z}^+ \qquad \forall j \in N^0,\ \forall k \in N \tag{3.11}$$

We minimize the total system cost in (3.1) which is composed of the cost of opening centralized and decentralized facilities and the cost of establishing lower-level and upper-level network connections. Problem has $2n^2+3n$ binary variables, $n^2+n$ integer variables, and $7n+6n^2+1$ constraints. Constraint (3.2) assigns each demand node to an exactly one facility, while (3.3) guarantees that each facility that a demand node is assigned has to be opened as either a centralized facility or a decentralized facility. Equation (3.4) is the flow balance constraint of the formulation. It ensures that if node $j$ is opened as a centralized facility, the total flow coming from the other opened centralized facilities and the source station has to be equal to the total flow going out from this node j to other opened centralized facilities. The flow is allowed to occur in one direction, from the source node to other opened facilities. While minimizing the objective function, all possible cycles in the upper-level network are eliminated by this constraint and constraint (3.5), which guarantees that in order to send a flow from any opened centralized facility or the source node, to another opened centralized facility, there has to be an upper-level network connection established between them. Thus, these two constraints under minimization setting, ensures that the total flow going from the source node to opened centralized facilities is equal to the total number of opened centralized facilities, which yields an upper-level network in tree configuration. Constraint (3.6) simply prevents to establish an upper-level network connection between nodes if they are not opened as centralized facilities. Finally, constraints (3.7)-(3.11) are the domain constraints of the decision variables.

The problem is NP-hard because special cases of the problem are result in NP-hard problems as discussed in Chapter 2.

## 3.2 Multi-Commodity Flow Based Formulation

Since the most related work with our problem in the literature formulate the problem [29] with multi-commodity flow constraints, we wanted to compare the performance of the multi-commodity flow based formulation of our problem with the single-commodity flow based formulation. In this formulation, instead of the flow variable defined in the previous formulation, $f_{jk}$, we defined the following flow variable;

$f^t_{jk}$ = flow towards node t on the upper-level network connection between opened centralized facilities at nodes $j$ and $k$, where $j \in N^0$, $k \in N$ and $t \in N$.

The rest of the parameters and decision variables remain unchanged and the following formulation is obtained;

$$min \quad \sum_{j=1}^{n} x_j C_{CF} + \sum_{j=1}^{n} y_j C_{DF} + \sum_{i=1}^{n}\sum_{j=1}^{n} z_{ij} D_{ij} C_{LN} + \sum_{j=0}^{n}\sum_{k=0}^{n} u_{jk} D_{jk} C_{UN}$$

$s.t.$

$$(3.2), (3.3), (3.6 - 3.10), and$$

$$\sum_{k=1}^{n} f_{0k}^{t} = x_t \qquad \forall t \in N \tag{3.12}$$

$$\sum_{k=0}^{n} f_{kt}^{t} = x_t \qquad \forall t \in N \tag{3.13}$$

$$\sum_{k=0}^{n} f_{kj}^{t} = \sum_{k=1}^{n} f_{jk}^{t} \qquad \forall j \in N^0,\ \forall t \in N : j \neq k \tag{3.14}$$

$$f_{jk}^{t} \leq u_{jk} \qquad \forall (j,k) \mid j \in N^0, k \in N, j \neq k,\ \forall t \in N \tag{3.15}$$

$$f_{jk}^{t} \in \{0,1\} \qquad \forall j \in N^0,\ \forall k \in N^0,\ \forall t \in N \tag{3.16}$$

Problem has $n^3 + 4n^2 + 4n$ binary variables and $5n + 5n^2 + n^3 + 1$ constraints. Constraints (3.12 - 3.14) are the flow balance constraints of this formulation. These constraints together with (3.15) are for the upper-level network. Constraints (3.12 - 3.14) forces one unit of commodity to be routed from source node to node t only when a centralized facility is opened at node t, thereby establishing connectivity of source station to all opened centralized facilities. Constraint (3.15) ensures that the flow between nodes is allowed only if there exist an upper-level network connection between them. Again, these flow constraints under the minimization setting yield a tree-type upper-level network, as it is in the single-commodity flow based formulation. Lastly, constraint (3.16) is the domain constraints for the decision variable $f_{jk}^{t}$. We denote this "multi-commodity flow based formulation" by 2LDN-MCF.

## 3.3 Comparison of the Formulations

In this section, we compare the computational performances of 2LND and 2LDN-MCF.

We created test instances at two different sizes (50 nodes and 200 nodes) and three different spatial distribution patterns, which we call Type I, II and III. The first one consists of uniformly distributed demand nodes. The second one is clustered and the third one is a mix of I and II, which has one uniformly distributed dense part that constitutes the majority of its demand nodes and has some separated clusters containing the remaining nodes. We also decided to generate these three types of maps for two different numbers of nodes in order to have a better comparison. One is small enough to find optimal solutions, at least for some of the instances we experiment, in a reasonable time and another is big enough to challenge the models. Thereby, we are able to compare these two formulations according to their solution times, when they are able to solve the problem optimally, as well as their objective values when they are not able to prove optimality in a given time limit of 18000 CPU seconds. The figures of the generated maps with different number of nodes are provided in Figure 3.1.

Each formulation is coded in the OPL script and solved by using IBM CPLEX Optimizer version 12.6.3 on a computer with the following specifications:

- CPU: Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz
- RAM: 64 GB

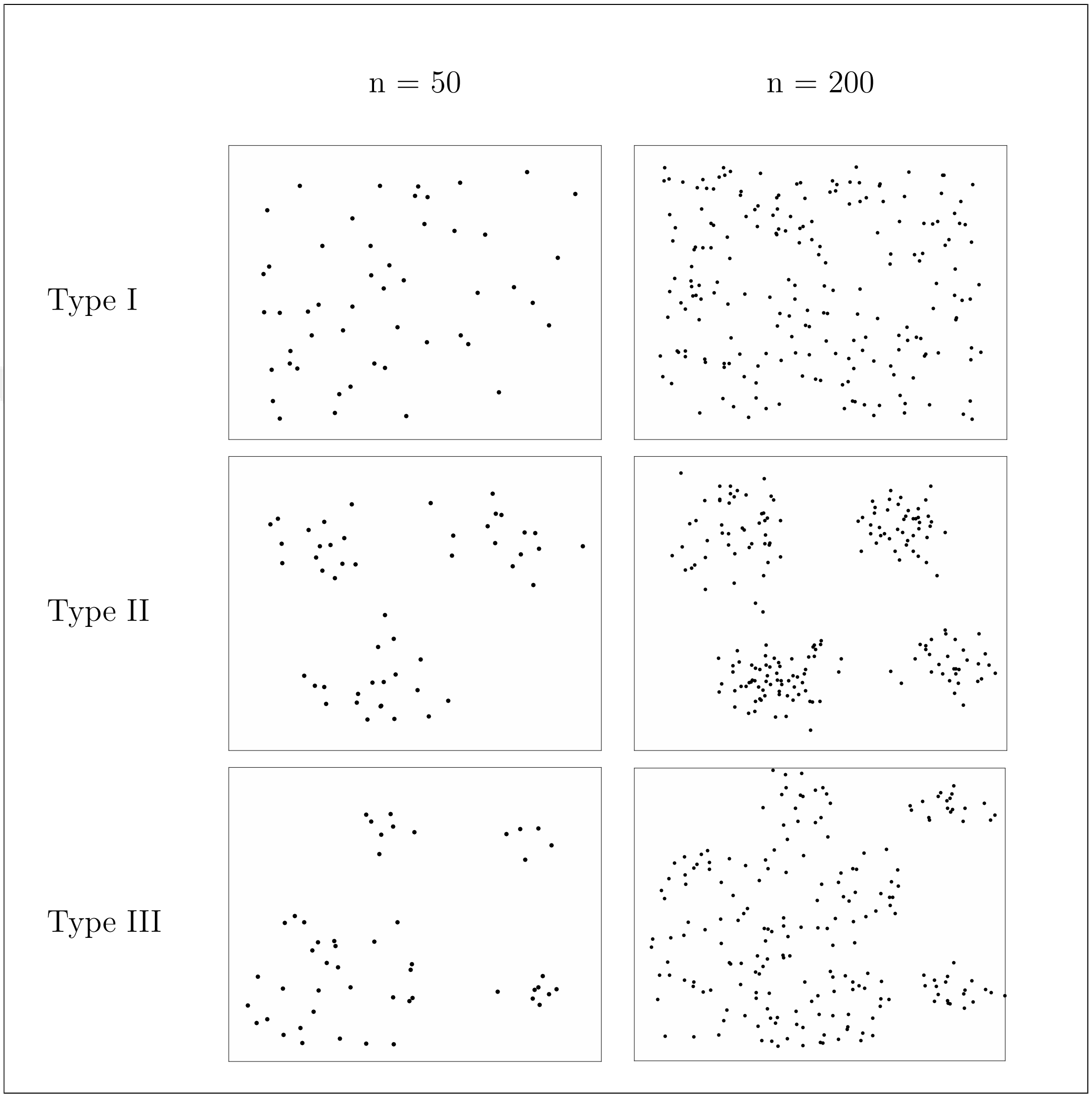



Figure 3.1: Maps of the Randomly Generated Data

The computational results with various parameter combinations are provided in Table 3.1. The first 6 columns represent the settings of the instances: type of the spatial distribution pattern, number of the demand nodes, cost of opening a centralized facility, cost of opening a decentralized facility, cost per meter of lower-level network connection, cost per meter of upper-level network connection. The *Objective* column indicates the objective value of the obtained solution, the column titled *# of CF* shows the number of opened centralized facilities, and the column titled *# of DF* shows the number of opened decentralized facilities. The column titled *Gap*, in percentage format shows the difference between the objective and the best bound found. *Time*, in seconds, is the CPU time of the solution recorded from IBM CPLEX Optimizer. The time limit set as 18000 seconds for this experiment. Additionally, the objective and the best bound values are rounded, the entry *N/A* indicates that no solutions can be obtained due to either time or memory limitations, and bold entries indicate preferable ones.

According to the Table 3.1, single-commodity flow based formulation outperforms the multi-commodity flow based formulation in both the solution time, for the instances they are able to solve optimally, and the ability of obtaining better objective values for the instances they are not able to solve optimally within the given time limit.

In addition, LP relaxations of 2LDN and 2LDN-MCF are compared in Table 3.2. For the instances with size $n = 50$, 2LDN-MCF provided better LP relaxations, however, for the instances with size $n = 200$ 2LDN-MCF is not able to solve even the LP relaxation due to time and memory limitations. In conclusion, we decided to use single-commodity flow based formulation as a benchmark to test the performance of the heuristic approach proposed in Chapter 4.

Table 3.1: Numerical Analysis of the Comparison of the Single-Commodity Flow Formulation and the Multi-Commodity Flow Formulation

| Scenario Setting | | | | | | 2LND | | | | | | 2LND-MCF | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Pattern** | **n** | $C_{CF}$ | $C_{DF}$ | $C_{LN}$ | $C_{UN}$ | **Objective** | **# of CF** | **# of DF** | **Gap (%)** | **Best Bound** | **Time (sec.)** | **Objective** | **# of CF** | **# of DF** | **Gap (%)** | **Best Bound** | **Time (sec.)** |
| Type I | 50 | 5000 | 15000 | 1 | 2.5 | **169359** | 4 | 1 | 2.45 | 165202 | 18000 | 174473 | 5 | 2 | 18.42 | 142329 | 18000 |
| Type I | 50 | 5000 | 20000 | 1 | 2.5 | **172218** | 5 | 0 | 5.84 | 162165 | 18000 | 174452 | 7 | 0 | 18.16 | 142769 | 18000 |
| Type I | 50 | 5000 | 20000 | 1 | 5 | **189097** | 1 | 3 | 0 | 189097 | 792 | 215775 | 5 | 0 | 24.24 | 163467 | 18000 |
| Type I | 50 | 5000 | 30000 | 1 | 2.5 | **172170** | 6 | 0 | 5.62 | 162489 | 18000 | 172604 | 6 | 0 | 16.13 | 144758 | 18000 |
| Type I | 50 | 5000 | 30000 | 1 | 5 | **210182** | 4 | 0 | 9.96 | 189238 | 18000 | 215775 | 5 | 0 | 23.85 | 164317 | 18000 |
| Type II | 50 | 5000 | 15000 | 1 | 2.5 | **168554** | 0 | 3 | 0.01 | 168545 | 2695 | **168554** | 0 | 3 | 0 | 168554 | 7102 |
| Type II | 50 | 5000 | 20000 | 1 | 2.5 | **180307** | 3 | 2 | 6.82 | 168007 | 18000 | 192767 | 4 | 1 | 20.76 | 152758 | 18000 |
| Type II | 50 | 5000 | 20000 | 1 | 5 | 183554 | 0 | 3 | 0 | 183551 | **167** | 183554 | 0 | 3 | 0 | 183554 | 4388 |
| Type II | 50 | 5000 | 30000 | 1 | 2.5 | 196750 | 8 | 0 | 15.9 | 165464 | 18000 | **195567** | 7 | 0 | 23.69 | 149239 | 18000 |
| Type II | 50 | 5000 | 30000 | 1 | 5 | **213554** | 0 | 3 | 0.01 | 213533 | 3165 | 243038 | 4 | 1 | 20.49 | 193244 | 18000 |
| Type III | 50 | 5000 | 15000 | 1 | 2.5 | 171823 | 1 | 5 | 0 | 171823 | **607** | 171823 | 1 | 5 | 0 | 171823 | 16506 |
| Type III | 50 | 5000 | 20000 | 1 | 2.5 | **194597** | 4 | 3 | 0.01 | 194578 | 10317 | 217084 | 8 | 0 | 29.08 | 153964 | 18000 |
| Type III | 50 | 5000 | 20000 | 1 | 5 | 202226 | 1 | 5 | 0.01 | 202208 | **295** | 202226 | 1 | 5 | 0 | 202226 | 8745 |
| Type III | 50 | 5000 | 30000 | 1 | 2.5 | **212120** | 7 | 0 | 9.44 | 192088 | 18000 | 217745 | 7 | 0 | 27.56 | 157741 | 18000 |
| Type III | 50 | 5000 | 30000 | 1 | 5 | **247872** | 1 | 4 | 0.01 | 247847 | 5238 | 309914 | 7 | 0 | 39.4 | 187812 | 18000 |
| Type I | 200 | 5000 | 15000 | 1 | 2.5 | **409733** | 19 | 0 | 25.24 | 306311 | 18000 | N/A | N/A | N/A | N/A | N/A | N/A |
| Type I | 200 | 5000 | 20000 | 1 | 2.5 | **404177** | 20 | 0 | 24.52 | 305075 | 18000 | N/A | N/A | N/A | N/A | N/A | N/A |
| Type I | 200 | 5000 | 20000 | 1 | 5 | **517425** | 21 | 0 | 36.99 | 326036 | 18000 | N/A | N/A | N/A | N/A | N/A | N/A |
| Type I | 200 | 5000 | 30000 | 1 | 2.5 | **399474** | 19 | 0 | 23.69 | 304831 | 18000 | N/A | N/A | N/A | N/A | N/A | N/A |
| Type I | 200 | 5000 | 30000 | 1 | 5 | **532813** | 20 | 1 | 38.43 | 328066 | 18000 | N/A | N/A | N/A | N/A | N/A | N/A |
| Type II | 200 | 5000 | 15000 | 1 | 2.5 | **441769** | 19 | 0 | 26.22 | 325923 | 18000 | N/A | N/A | N/A | N/A | N/A | N/A |
| Type II | 200 | 5000 | 20000 | 1 | 2.5 | **444451** | 21 | 0 | 26.55 | 326468 | 18000 | N/A | N/A | N/A | N/A | N/A | N/A |
| Type II | 200 | 5000 | 20000 | 1 | 5 | **545074** | 5 | 13 | 33.7 | 361394 | 18000 | N/A | N/A | N/A | N/A | N/A | N/A |
| Type II | 200 | 5000 | 30000 | 1 | 2.5 | **445719** | 22 | 0 | 26 | 329816 | 18000 | N/A | N/A | N/A | N/A | N/A | N/A |
| Type II | 200 | 5000 | 30000 | 1 | 5 | **581487** | 20 | 0 | 38.92 | 355172 | 18000 | N/A | N/A | N/A | N/A | N/A | N/A |
| Type III | 200 | 5000 | 15000 | 1 | 2.5 | **520149** | 20 | 1 | 23.2 | 399457 | 18000 | N/A | N/A | N/A | N/A | N/A | N/A |
| Type III | 200 | 5000 | 20000 | 1 | 2.5 | **550356** | 20 | 0 | 30.44 | 382804 | 18000 | N/A | N/A | N/A | N/A | N/A | N/A |
| Type III | 200 | 5000 | 20000 | 1 | 5 | **696550** | 22 | 1 | 37.79 | 433343 | 18000 | N/A | N/A | N/A | N/A | N/A | N/A |
| Type III | 200 | 5000 | 30000 | 1 | 2.5 | **547544** | 27 | 0 | 30.34 | 381409 | 18000 | N/A | N/A | N/A | N/A | N/A | N/A |
| Type III | 200 | 5000 | 30000 | 1 | 5 | **732455** | 26 | 1 | 43.82 | 411514 | 18000 | N/A | N/A | N/A | N/A | N/A | N/A |

Table 3.2: LP Relaxations of 2LDN and 2LDN-MCF

| Scenario Setting | | | | | | 2LDN | 2LDN-MCF |
|---|---|---|---|---|---|---|---|
| **Pattern** | **n** | $C_{CF}$ | $C_{DF}$ | $C_{LN}$ | $C_{UN}$ | **LP Relaxation** | **LP Relaxation** |
| Type I | 50 | 5000 | 15000 | 1 | 2.5 | 114553.3 | **140380.565** |
| Type I | 50 | 5000 | 20000 | 1 | 2.5 | 114553.3 | **140380.565** |
| Type I | 50 | 5000 | 20000 | 1 | 5 | 118457.4 | **158125.517** |
| Type I | 50 | 5000 | 30000 | 1 | 2.5 | 114553.3 | **140380.565** |
| Type I | 50 | 5000 | 30000 | 1 | 5 | 118457.4 | **158125.517** |
| Type II | 50 | 5000 | 15000 | 1 | 2.5 | 116823.85 | **145824.394** |
| Type II | 50 | 5000 | 20000 | 1 | 2.5 | 116823.85 | **145824.394** |
| Type II | 50 | 5000 | 20000 | 1 | 5 | 121571.7 | **165919.738** |
| Type II | 50 | 5000 | 30000 | 1 | 2.5 | 116823.85 | **145824.394** |
| Type II | 50 | 5000 | 30000 | 1 | 5 | 121571.7 | **165919.738** |
| Type III | 50 | 5000 | 15000 | 1 | 2.5 | 113179.7 | **146674.978** |
| Type III | 50 | 5000 | 20000 | 1 | 2.5 | 113179.7 | **146674.978** |
| Type III | 50 | 5000 | 20000 | 1 | 5 | 118384.9 | **173189.024** |
| Type III | 50 | 5000 | 30000 | 1 | 2.5 | 113179.7 | **146674.978** |
| Type III | 50 | 5000 | 30000 | 1 | 5 | 118384.9 | **173453.667** |
| Type I | 200 | 5000 | 15000 | 1 | 2.5 | **277721.8** | N/A |
| Type I | 200 | 5000 | 20000 | 1 | 2.5 | **277721.8** | N/A |
| Type I | 200 | 5000 | 20000 | 1 | 5 | **280544.825** | N/A |
| Type I | 200 | 5000 | 30000 | 1 | 2.5 | **277721.8** | N/A |
| Type I | 200 | 5000 | 30000 | 1 | 5 | **280544.825** | N/A |
| Type II | 200 | 5000 | 15000 | 1 | 2.5 | **287293.038** | N/A |
| Type II | 200 | 5000 | 20000 | 1 | 2.5 | **287293.037** | N/A |
| Type II | 200 | 5000 | 20000 | 1 | 5 | **291004.35** | N/A |
| Type II | 200 | 5000 | 30000 | 1 | 2.5 | **287293.037** | N/A |
| Type II | 200 | 5000 | 30000 | 1 | 5 | **291004.35** | N/A |
| Type III | 200 | 5000 | 15000 | 1 | 2.5 | **339855.613** | N/A |
| Type III | 200 | 5000 | 20000 | 1 | 2.5 | **339855.613** | N/A |
| Type III | 200 | 5000 | 20000 | 1 | 5 | **345007.075** | N/A |
| Type III | 200 | 5000 | 30000 | 1 | 2.5 | **339855.612** | N/A |
| Type III | 200 | 5000 | 30000 | 1 | 5 | **345007.075** | N/A |

## 3.4 Optimality Cuts and Valid Inequalities

We propose the equations to improve the exact formulation we decided to use as a benchmark. In this section, we analyze their computational performances.

**Proposition 1.**

$$x_j + y_j \leq z_{jj} \qquad \forall j \in N \tag{3.17}$$

*is an optimality cut for 2LDN.*

Equation (3.17) forces that if there exist a facility on a node, the demand point located on that node should be connected to that facility, which is an optimality cut for the formulation.

**Proposition 2.**

$$\sum_{j=0}^{n} u_{jk} = x_k \qquad \forall k \in N \tag{3.18}$$

*is a valid inequality for 2LDN.*

Since formulation prevents cycles in upper-level network, there has to be only one upper-level connection coming to each centralized facility. Thereby, Equation (3.18) does not cut any feasible regions of the problem and is not implied from the constraints of the 2LDN.

We conduct another experiment with the same instances we used in Section 3.3 to test the computational performance of the proposed valid inequalities. Table 3.4 shows the results of the comparison of the single-commodity flow formulation with and without the combination of the constraints (3.17) and (3.18).

In order to compare the models, we examine the minimum *Objective* obtained for each instance and if more than one model is able to find the minimum cost then we compare their *Time* or *BestBound* depending on whether the *Objective* is optimal or not. Experiment results provided in Table 3.4 are summarized in Table 3.3. The first column indicates the formulation alternative that we tried. The second column shows the total number of the instances for which the formulation provides the best solution. The third column shows the total number of the instances for which the formulation provides the worst solution. The fourth column shows the worst gap of the formulations according to the best solutions obtained for the instances. Gap is calculated as ("Objective" - "Best Objective obtained for the Instance")/"Objective". The last column shows the average gap of the formulations according to the best solutions of the instances. Bold entries indicate the most preferable.

Table 3.3: Summary of the Optimality Cut/Valid Inequality Experiment

| Formulations | Number of Instances with Best Solution (Objective Value and Solution Time) | Number of Instances with Worst Solution (Objective Value and Solution Time) | Worst Gap from the Best Solution (%) | Average Gap from the Best Solution (%) |
|---|---|---|---|---|
| 2LDN | **12** | **5** | 12.80 | **1.21** |
| 2LDN w/ (3.17) | 6 | 11 | **7.91** | 1.36 |
| 2LDN w/ (3.18) | 7 | 6 | 11.15 | 2.56 |
| 2LDN w/ (3.17) & (3.18) | 10 | 9 | 16.33 | 2.85 |

In addition, LP relaxations of the formulations are compared in Table 3.5. Results of the LP relaxations are the same for the instances with size $n = 50$. For the instances with size $n = 200$, the third and the forth formulation obtained better bounds, however, the improvement is negligible (1.2e-007% on the average).

In conclusion, 2LDN itself provided the better solutions on more instances than the other combinations. Moreover, The LP relaxations of the formulations are indifferent. Thus, we decided to continue using 2LDN without any of the proposed valid inequality as a benchmark for our heuristic algorithm in Chapter 5.

Table 3.4: Numerical Analysis of the Optimality Cut/Valid Inequality Combinations

| Scenario Setting | | | | | | 2LDN | | | 2LND with Eq. (3.17) | | | 2LND with Eq. (3.18) | | | 2LND with Eq. (3.17) and Eq. (3.18) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Pattern** | **n** | $C_{CF}$ | $C_{DF}$ | $C_{LN}$ | $C_{UN}$ | **Objective** | **Best Bound** | **Time (sec.)** | **Objective** | **Best Bound** | **Time (sec.)** | **Objective** | **Best Bound** | **Time (sec.)** | **Objective** | **Best Bound** | **Time (sec.)** |
| Type I | 50 | 5000 | 15000 | 1 | 2.5 | **169359** | 165202 | 18000 | 169671 | 163749 | 18000 | **169359** | 167601 | 18000 | **169359** | 166434 | 18000 |
| Type I | 50 | 5000 | 20000 | 1 | 2.5 | 172218 | 162165 | 18000 | 172261 | 166521 | 18000 | 172170 | 164147 | 18000 | **172103** | **167553** | 18000 |
| Type I | 50 | 5000 | 20000 | 1 | 5 | 189097 | 189097 | 792 | 189097 | 189088 | 1937 | 189097 | 189079 | 546 | 189097 | 189089 | **190** |
| Type I | 50 | 5000 | 30000 | 1 | 2.5 | **172170** | 162489 | 18000 | 172233 | 161012 | 18000 | **172170** | 166045 | 18000 | 172233 | 165685 | 18000 |
| Type I | 50 | 5000 | 30000 | 1 | 5 | **210182** | 189238 | 18000 | 210943 | 190395 | 18000 | 210896 | 200381 | 18000 | 211402 | 197339 | 18000 |
| Type II | 50 | 5000 | 15000 | 1 | 2.5 | 168554 | 168545 | 2695 | 168554 | 168540 | 5008 | 168554 | 168540 | 1839 | 168554 | 168554 | **1320** |
| Type II | 50 | 5000 | 20000 | 1 | 2.5 | 180307 | 168007 | 18000 | **179873** | 174329 | 18000 | **179873** | 175798 | 18000 | **179873** | 176730 | 18000 |
| Type II | 50 | 5000 | 20000 | 1 | 5 | 183554 | 183551 | 167 | 183554 | 183540 | 207 | 183554 | 183554 | 65 | 183554 | 183554 | **61** |
| Type II | 50 | 5000 | 30000 | 1 | 2.5 | 196750 | 165464 | 18000 | 196611 | 167419 | 18000 | **196431** | 171140 | 18000 | 196611 | 176533 | 18000 |
| Type II | 50 | 5000 | 30000 | 1 | 5 | 213554 | 213533 | 3165 | 213554 | 213535 | 4503 | 213554 | 213546 | **1811** | 213554 | 213542 | 2556 |
| Type III | 50 | 5000 | 15000 | 1 | 2.5 | 171823 | 171823 | 607 | 171823 | 171823 | 684 | 171823 | 171806 | 362 | 171823 | 171817 | **329** |
| Type III | 50 | 5000 | 20000 | 1 | 2.5 | 194597 | 194578 | 10317 | 194597 | 194578 | 5989 | 194597 | 194578 | 2748 | 194597 | 194578 | **1899** |
| Type III | 50 | 5000 | 20000 | 1 | 5 | 202226 | 202208 | 295 | 202226 | 202226 | 598 | 202226 | 202226 | 90 | 202226 | 202226 | **77** |
| Type III | 50 | 5000 | 30000 | 1 | 2.5 | 212120 | 192088 | 18000 | 212581 | 197521 | 18000 | **211873** | 206863 | 18000 | 212114 | 209572 | 18000 |
| Type III | 50 | 5000 | 30000 | 1 | 5 | 247872 | 247847 | 5238 | 247872 | 247872 | 1532 | 247872 | 247872 | **1084** | 247872 | 247872 | 1398 |
| Type I | 200 | 5000 | 15000 | 1 | 2.5 | **409733** | 306311 | 18000 | 419596 | 306084 | 18000 | 439306 | 307044 | 18000 | 433700 | 307999 | 18000 |
| Type I | 200 | 5000 | 20000 | 1 | 2.5 | **404177** | 305075 | 18000 | 433964 | 306156 | 18000 | 434049 | 308384 | 18000 | 430280 | 306430 | 18000 |
| Type I | 200 | 5000 | 20000 | 1 | 5 | 517425 | 326036 | 18000 | **472476** | 328449 | 18000 | 531780 | 329199 | 18000 | 564708 | 328447 | 18000 |
| Type I | 200 | 5000 | 30000 | 1 | 2.5 | **399474** | 304831 | 18000 | 403729 | 305109 | 18000 | 428965 | 306961 | 18000 | 446623 | 306115 | 18000 |
| Type I | 200 | 5000 | 30000 | 1 | 5 | **532813** | 328066 | 18000 | 570117 | 326240 | 18000 | 562374 | 329399 | 18000 | 584198 | 327193 | 18000 |
| Type II | 200 | 5000 | 15000 | 1 | 2.5 | **441769** | 325923 | 18000 | 446181 | 328452 | 18000 | 449485 | 317824 | 18000 | 450205 | 318141 | 18000 |
| Type II | 200 | 5000 | 20000 | 1 | 2.5 | **444451** | 326468 | 18000 | 450365 | 329633 | 18000 | 467563 | 319347 | 18000 | 458999 | 319272 | 18000 |
| Type II | 200 | 5000 | 20000 | 1 | 5 | **545074** | 361394 | 18000 | 585289 | 355462 | 18000 | 557120 | 336797 | 18000 | 560487 | 338007 | 18000 |
| Type II | 200 | 5000 | 30000 | 1 | 2.5 | **445719** | 329816 | 18000 | 460894 | 328322 | 18000 | 465228 | 317676 | 18000 | 461533 | 321962 | 18000 |
| Type II | 200 | 5000 | 30000 | 1 | 5 | 581487 | 355172 | 18000 | **541347** | 354769 | 18000 | 605723 | 353959 | 18000 | 609533 | 337393 | 18000 |
| Type III | 200 | 5000 | 15000 | 1 | 2.5 | **520149** | 399457 | 18000 | 533385 | 390383 | 18000 | 556599 | 396636 | 18000 | 544187 | 406482 | 18000 |
| Type III | 200 | 5000 | 20000 | 1 | 2.5 | 550356 | 382804 | 18000 | **539963** | 397133 | 18000 | 542091 | 398547 | 18000 | 552627 | 396180 | 18000 |
| Type III | 200 | 5000 | 20000 | 1 | 5 | 696550 | 433343 | 18000 | 659618 | 439951 | 18000 | 615151 | 443533 | 18000 | **607413** | 438418 | 18000 |
| Type III | 200 | 5000 | 30000 | 1 | 2.5 | 547544 | 381409 | 18000 | **533477** | 402000 | 18000 | 540497 | 397944 | 18000 | 553048 | 401553 | 18000 |
| Type III | 200 | 5000 | 30000 | 1 | 5 | 732455 | 411514 | 18000 | **712091** | 412750 | 18000 | 760126 | 437451 | 18000 | 747565 | 434279 | 18000 |

Table 3.5: LP Relaxation of the Optimality Cut/Valid Inequality Combinations

| Scenario Setting | | | | | | 2LDN | 2LDN w/ (3.17) | 2LDN w/ (3.18) | 2LDN w/ (3.17) & (3.18) |
|---|---|---|---|---|---|---|---|---|---|
| **Pattern** | **n** | $C_{CF}$ | $C_{DF}$ | $C_{LN}$ | $C_{UN}$ | **LP Relaxation** | **LP Relaxation** | **LP Relaxation** | **LP Relaxation** |
| Type I | 50 | 5000 | 15000 | 1 | 2.5 | **114553.3** | **114553.3** | **114553.3** | **114553.3** |
| Type I | 50 | 5000 | 20000 | 1 | 2.5 | **114553.3** | **114553.3** | **114553.3** | **114553.3** |
| Type I | 50 | 5000 | 20000 | 1 | 5 | **118457.4** | **118457.4** | **118457.4** | **118457.4** |
| Type I | 50 | 5000 | 30000 | 1 | 2.5 | **114553.3** | **114553.3** | **114553.3** | **114553.3** |
| Type I | 50 | 5000 | 30000 | 1 | 5 | **118457.4** | **118457.4** | **118457.4** | **118457.4** |
| Type II | 50 | 5000 | 15000 | 1 | 2.5 | **116823.85** | **116823.85** | **116823.85** | **116823.85** |
| Type II | 50 | 5000 | 20000 | 1 | 2.5 | **116823.85** | **116823.85** | **116823.85** | **116823.85** |
| Type II | 50 | 5000 | 20000 | 1 | 5 | **121571.7** | **121571.7** | **121571.7** | **121571.7** |
| Type II | 50 | 5000 | 30000 | 1 | 2.5 | **116823.85** | **116823.85** | **116823.85** | **116823.85** |
| Type II | 50 | 5000 | 30000 | 1 | 5 | **121571.7** | **121571.7** | **121571.7** | **121571.7** |
| Type III | 50 | 5000 | 15000 | 1 | 2.5 | **113179.7** | **113179.7** | **113179.7** | **113179.7** |
| Type III | 50 | 5000 | 20000 | 1 | 2.5 | **113179.7** | **113179.7** | **113179.7** | **113179.7** |
| Type III | 50 | 5000 | 20000 | 1 | 5 | **118384.9** | **118384.9** | **118384.9** | **118384.9** |
| Type III | 50 | 5000 | 30000 | 1 | 2.5 | **113179.7** | **113179.7** | **113179.7** | **113179.7** |
| Type III | 50 | 5000 | 30000 | 1 | 5 | **118384.9** | **118384.9** | **118384.9** | **118384.9** |
| Type I | 200 | 5000 | 15000 | 1 | 2.5 | 277721.8 | 277721.8 | **277721.825** | **277721.825** |
| Type I | 200 | 5000 | 20000 | 1 | 2.5 | 277721.8 | 277721.8 | **277721.825** | **277721.825** |
| Type I | 200 | 5000 | 20000 | 1 | 5 | 280544.825 | 280544.825 | **280544.9** | **280544.9** |
| Type I | 200 | 5000 | 30000 | 1 | 2.5 | 277721.8 | 277721.8 | **277721.825** | **277721.825** |
| Type I | 200 | 5000 | 30000 | 1 | 5 | 280544.825 | 280544.825 | **280544.9** | **280544.9** |
| Type II | 200 | 5000 | 15000 | 1 | 2.5 | 287293.038 | 287293.037 | **287293.05** | **287293.05** |
| Type II | 200 | 5000 | 20000 | 1 | 2.5 | 287293.037 | 287293.037 | **287293.05** | **287293.05** |
| Type II | 200 | 5000 | 20000 | 1 | 5 | 291004.35 | 291004.35 | **291004.375** | **291004.375** |
| Type II | 200 | 5000 | 30000 | 1 | 2.5 | 287293.037 | 287293.037 | **287293.05** | **287293.05** |
| Type II | 200 | 5000 | 30000 | 1 | 5 | 291004.35 | 291004.35 | **291004.375** | **291004.375** |
| Type III | 200 | 5000 | 15000 | 1 | 2.5 | 339855.613 | 339855.612 | **339855.65** | **339855.65** |
| Type III | 200 | 5000 | 20000 | 1 | 2.5 | 339855.613 | 339855.613 | **339855.65** | **339855.65** |
| Type III | 200 | 5000 | 20000 | 1 | 5 | 345007.075 | 345007.075 | **345007.15** | **345007.15** |
| Type III | 200 | 5000 | 30000 | 1 | 2.5 | 339855.612 | 339855.613 | **339855.65** | **339855.65** |
| Type III | 200 | 5000 | 30000 | 1 | 5 | 345007.075 | 345007.075 | **345007.15** | **345007.15** |

# Chapter 4

# Solution Approach

In order to solve the defined problem we also propose a heuristic algorithm. This algorithm iteratively solves several mathematical models that are relatively easier to solve in terms of complexity and size than the exact formulations provided in Chapter 3.

In our heuristic approach, we decompose the original problem into smaller subproblems and propose designing the lower-level and the upper-level networks sequentially. Furthermore, we decided to fix the number of facilities in order to decrease the size and the complexity of the subproblems and iteratively solve them for different number of facilities.

Our heuristic method has 3 steps as shown in Figure 4.1. In Step 1, the algorithm uses the Simple Plant Location (SPLP) formulation to determine the upper and the lower bounds for the total number of facilities.

Afterwards, in Step 2, algorithm uses the p-Median formulation to design the lower-level network. For a specified number of facilities, it determines the locations of the facilities and establishes network connections which are, in our case, the lower-level network connections.

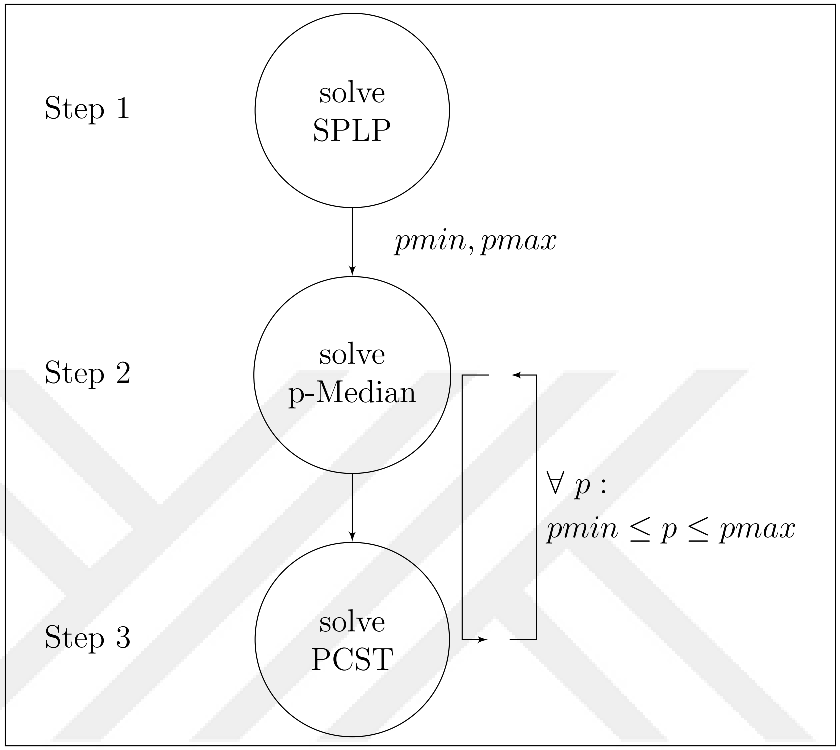



Figure 4.1: Flow Chart of the Heuristic Algorithm

Finally, in Step 3, the Price Collecting Steiner Tree (PCST) formulation is used to design the upper-level network. Since the locations of the facilities are decided, the remaining part of the problem reduces to the PCST problem. The PCST decides on the types of the facilities selected in Step 2 and establishes the upper-level level connections. Our heuristic repeats these steps for every number of facilities within the lower and the upper bounds obtained in Step 1 and selects the best result.

In the rest of this section, we first introduce the mathematical models that are used in our heuristic algorithm and then provide the pseudo code together with the detailed explanation of the heuristic.

## 4.1 Subproblems Solved in the Heuristic Algorithm

Our heuristic algorithm iteratively solves three subproblems that are widely studied in the literature as we discussed in Chapter 2. The first problem is the Simple Plant Location Problem, the second is the p-Median Problem and the third one is a variation of the Prize Collecting Steiner Tree Problem.

Hereby, we declare the notation used in the following sections to explicate the SPLP, the p-Median and the PCTS. Consider a complete network with $n$ demand nodes and a source node. Let $N$ be the set of demand nodes from 1 to $n$, $N^0$ be the set of all nodes from 0 to $n$ where 0 denotes the source node and let $D_{ij}$ be the distance between node $i \in \{0, \dots, n\}$ and node $j \in \{0, \dots, n\}$.

In addition, we declare the common decision variables used in the SPLP and the p-Median below.

$$x_j = \begin{cases} 1 & \text{if a facility is opened at node } j \text{, where } j \in N, \\ 0 & \text{otherwise.} \end{cases}$$

$$z_{ij} = \begin{cases} 1 & \text{if demand node } i \text{ is served by facility } j \text{, where } i \in N \text{ and } j \in N, \\ 0 & \text{otherwise.} \end{cases}$$

### 4.1.1 Simple Plant Location Problem

The Simple Plant Location Problem is a well-known problem that has many variations in the literature. In our heuristic algorithm, we employ a version which can be described as the uncapacitated SPLP with fixed costs [50].

The cost parameters are $f$ and $c$ which are, respectively, the facility opening cost and the unit cost of connection between a demand node and a facility. We want each demand node to be assigned exactly to one facility and our aim is to find the number and the locations of facilities that minimizes the total facility opening and connection costs. The explicit formulation is stated below.

$$min \quad \sum_{j=1}^{n} x_j f + \sum_{i=1}^{n} \sum_{j=1}^{n} z_{ij} D_{ij} c \tag{4.1}$$

$s.t$

$$\sum_{j=1}^{n} z_{ij} = 1 \qquad \forall i \in N \tag{4.2}$$

$$z_{ij} \leq x_j \qquad \forall i \in N, \ \forall j \in N \tag{4.3}$$

$$x_j \in \{0, 1\} \qquad \forall j \in N \tag{4.4}$$

$$z_{ij} \in \{0, 1\} \qquad \forall i \in N, \ \forall j \in N \tag{4.5}$$

Constraint (4.2) connects each demand node to exactly one facility and constraints (4.3) ensures that a facility assigned to a demand node has to be opened. Constraints (4.4) and (4.5) are the binary decision variable definitions.

### 4.1.2 p-Median Problem

In this problem our aim is to find the minimum connection cost while serving all demand nodes with a predetermined number of facilities, which is denoted by $p$.

$$min \quad \sum_{i=1}^{n}\sum_{j=1}^{n} z_{ij} D_{ij} c \tag{4.6}$$

$s.t$

$$(4.2) - (4.5), and$$

$$\sum_{j=1}^{n} x_j = p \tag{4.7}$$

The objective function minimizes the connection cost and constraint (4.7) ensures to open exactly $p$ number of facilities.

### 4.1.3 Prize Collecting Steiner Tree Problem

Generalized Price Collecting Steiner Tree problem is defined for a given undirected graph $G = (V, E)$, as finding a subtree, $T' = (V', E')$ where $V' \subseteq V$ and $E' \subseteq E$, while maximizing the total prize collected from the selected vertices after paying the costs of the edges selected to construct $T'$

$$max \quad \sum_{v \in V'} p(v) - \sum_{e \in E'} c(e) \tag{4.8}$$

where $p(v)$ is non-negative prize $\forall v \in V$ and $c(e)$ is the non-negative edge cost $\forall e \in E$. The PCST problem that aims to maximize Equation (4.8) is also called as the Network Maximization Problem by Johnson et al. Moreover, the rooted variant of the PCST is simply the model which forces vertex $v_0$ to be included in the subtree $T'$ where $v_0 \in V$ is a specified root vertex [51].

The last model we employ in our heuristic is a variant of the Prize Collecting Steiner Tree problem known as Goemans and Williamson Minimization Problem (GWMP). For a given undirected graph $G = (V, E)$, Goemans and Williamson Minimization Problem is the following;

$$min \quad \sum_{e \in E'} c(e) + \sum_{v \notin V'} p(v) \tag{4.9}$$

where $p(v)$ are now defined as non-negative penalty cost $\forall v \in V$ which has to be paid for the vertices which are not selected to subtree $T' = (V', E')$ where $V' \subseteq V$ and $E' \subseteq E$ [48].

In Step 3 of our heuristic method, we use rooted variant of the GWMP with the following cost parameters.

- $\rho$ : penalty cost for not selecting a node to the subtree which is, in our case, the difference between the decentralized and centralized facility opening cost; and
- $c$ : cost per meter of a connection established between nodes.

Note that general PCST may contain non-terminal nodes, also called Steiner nodes, which can be excluded without paying any penalties. However, in our problem, all of the nodes are defined as terminal nodes. Decision variables and constraints are the following;

$$x_j = \begin{cases} 1 & \text{if node j is selected in the subtree, where } j \in N, \\ 0 & \text{otherwise.} \end{cases}$$

$$u_{ij} = \begin{cases} 1 & \text{if the connection between node } i \text{ and node } j \text{ is established,} \\ & \text{where } i \in N^0 \text{ and } j \in N, \\ 0 & \text{otherwise.} \end{cases}$$

$$f_{ij} = \text{ flow from } i^{th} \text{ facility to } j^{th} \text{ facility, where } i \in N^0 \text{ and } j \in N.$$

$$min \quad \sum_{i=0}^{n} \sum_{j=1}^{n} u_{ij} D_{i,j} c + \sum_{j=1}^{n} (1 - x_j)\rho \tag{4.10}$$

$s.t$

$$(3.4) - (3.7), (3.10), and\ (3.11) \tag{4.11}$$

The objective function (4.10) is minimizing the total connection cost of the nodes included in the subtree and the total penalty paid for the nodes which are excluded from the subtree.

## 4.2 Heuristic Algorithm

The pseudo code of our heuristic is provided at the end of this section. The algorithm takes the locations of the demand nodes and the cost parameters as inputs and reports the objective value, types and locations of the selected facilities as its output.

First, the SPLP model, provided in section 4.1.1, is solved with $f = C_{CF}$ to obtain the lower bound on the number of facilities denoted by $pmin$. Then the SPLP model is solved with $f = C_{DF}$ to obtain the upper bound on the number of facilities denoted by $pmax$. Then, Steps 2 and 3 are consecutively executed for each $p$ value between $pmin$ and $pmax$.

For a fixed number of facilities, the p-Median problem designs the lower-level network by deciding on the locations of the facilities and the connections between facilities and demand nodes. Subsequently, we feed the PCST problem with the output of the p-Median problem. Then, the PCST problem decides about the type of the facilities selected by p-Median and constructs a tree between them. If a facility is decided to be included in the tree,then it is a centralized facility and there arise a connection cost, which is the upper-level connection cost. However, if a facility decided to be excluded from the tree by paying a penalty cost, which is the difference between the opening costs of centralized and decentralized facilities, then that facility becomes a decentralized facility. Thereby, the facility opening and the upper-level network connection costs for each iteration, 4.12, is obtained by adding $obj + p * C_{CF}$ to the objective value of the PCST.

Recall the objective function of the PCST, (4.10), where $c$ equals to $C_{UN}$ and $\rho$ equals $C_{DF} - C_{CF}$;

$$\sum_{i=0}^{n}\sum_{j=1}^{n} u_{ij} D_{i,j} C_{UN} + \sum_{j=1}^{n}(1 - x_j)(C_{DF} - C_{CF})$$

$$= \sum_{i=0}^{n}\sum_{j=1}^{n} u_{ij} D_{i,j} C_{UN} + \sum_{j=1}^{n} y_j C_{DF} - \sum_{j=1}^{n} y_j C_{CF}$$

add $p * C_{CF}$ which equals to $\sum_{j=1}^{n} C_{CF}$ since $n$ is equal to $p$

$$\implies \sum_{i=0}^{n}\sum_{j=1}^{n} u_{ij} D_{i,j} C_{UN} + \sum_{j=1}^{n} y_j C_{DF} + \sum_{j=1}^{n}(1 - y_j) C_{CF}$$

$$= \sum_{i=0}^{n}\sum_{j=1}^{n} u_{ij} D_{i,j} C_{UN} + \sum_{j=1}^{n} y_j C_{DF} + \sum_{j=1}^{n} x_j C_{CF} \tag{4.12}$$

Then, the objective value of the original problem for each $p$ value, (3.1), is obtained and recorded by adding (4.12) to the objective value of the p-Median problem, 4.6. Moreover, the connections obtained from the p-Median problem are recorded as the lower-level network connections and the connections obtained from the PCST problem are recorded as the upper-level network connections. The facilities selected by PCST among the facilities selected by p-Median recorded as centralized facilities while the remaining ones are recorded as decentralized facilities.

Finally, after the iteration block is completed, the minimum objective value is selected and the related solution is reported.

**Algorithm 1:** Heuristic Algorithm

**input :** $L(lat, lon)$, $C_{CF}$, $C_{DF}$, $C_{LN}$ and $C_{UN}$
**output:** $Objective$, $centF$, $decentF$, $LN$ and $UN$,

$[\sim, \mathsf{x}] \leftarrow$ `Solve_SPLP(`$C_{CF}$, $C_{LN}$, $L(lat, lon)$`)`;
pmin $\leftarrow$ `sum(x)`;
$[\sim, \mathsf{x}] \leftarrow$ `Solve_SPLP(`$C_{DF}$, $C_{LN}$, $L(lat, lon)$`)`;
pmax $\leftarrow$ `sum(x)`;

*initialization*;
results $\leftarrow$ empty array of size (pmax- pmin + 1) by 1;
solutions $\leftarrow$ empty cell of size (pmax- pmin + 1) by 4;

**for** $p \leftarrow$ pmin **to** pmax **do**
  [obj, X, z] $\leftarrow$ `Solve_p-Median(`$p$, $C_{LN}$, $L(lat, lon)$`)`;
  results $(p -$ pmin $+ 1) \leftarrow$ results $(p -$ pmin $+ 1) +$ obj;

  [obj, x, u] $\leftarrow$ `Solve_PCST(`$C_{UN}$, $(C_{DF} - C_{CF})$, $L(X > 0, :)$`)`;
  results $(p -$ pmin $+ 1) \leftarrow$ results $(p -$ pmin $+ 1) +$ obj $+ p \times C_{CF}$;

  solutions $\{p -$ pmin $+ 1, 1\} \leftarrow$ X (x $> 0$);
  solutions $\{p -$ pmin $+ 1, 2\} \leftarrow$ X (x $= 0$);
  solutions $\}p -$ pmin $+ 1, 3\} \leftarrow$ z;
  solutions $\{p -$ pmin $+ 1, 4\} \leftarrow$ u;
**end**

[Objective, index] = `min(results)`;
$centF \leftarrow$ solutions {index,1} ;
$decentF \leftarrow$ solutions {index,2} ;
$LN \leftarrow$ solutions {index,3} ;
$UN \leftarrow$ solutions {index,4} ;

**return** $Objective$, $centF$, $decentF$, $LN$ and $UN$

# Chapter 5

# Numerical Analysis

In this chapter, we analyze the performance of the heuristic algorithm proposed in Chapter 4. With this aim, we compare it with 2LND which is coded in the OPL script and solved by using IBM CPLEX Optimizer version 12.6. while the heuristic algorithm is coded in MATLAB on the same computer which has the following technical specifications; CPU: Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz, RAM: 64 GB, OS: x86_64 GNU.

We conduct several experiments with both random and real data. While generating random data and procuring real data, we paid utmost attention to obtain different scenarios in terms of spatial distribution pattern and size. Besides, we also create different instances for each dataset by using wide ranges of parameter sets.

We examine our results in terms of three cost ratios:

- $C_{DF}/C_{CF}$: ratio between the cost of opening decentralized and centralized facilities.
- $C_{CF}/C_{LN}$: ratio between the cost of establishing one unit of upper-level network and lower-level network.

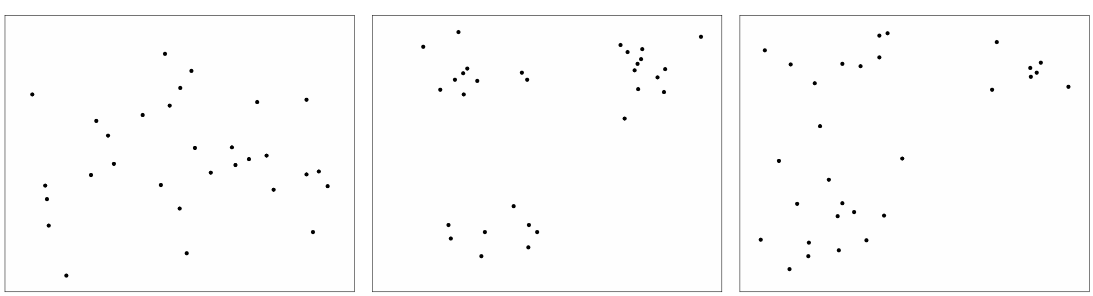
- $C_{UN}/C_{LN}$, ratio between the cost of opening one centralized facility and cost of establishing one unit of upper-level network.

According to these ratios and the characteristics of the data such as total area, distances between nodes, spatial distribution pattern etc. the problem yields to a centralized system, decentralized system or a hybrid system.

In this section, we first provide the results with the random data. Then, we continue with presenting the experiments performed with real data. While evaluating the performance of our heuristic algorithm, we analyze iterative performance of the algorithm and how we addressed the parameter selection process. In conclusion, we discuss all of the findings and provide our suggestions.

## 5.1 Experiments with Random Data

The aim of this section is to compare the performance of our heuristic with exact solutions obtained by 2LND. We conduct our experiment with three different spatial distribution patterns with node sizes of 30 and 50. In addition to the random data with size $n = 50$ used in the Chapter 3.3, we generate another random data with size $n = 30$, provided in Figure 5.1, in order to obtain optimal solution for all of the instances. Besides, time limit is set to 90000 seconds for these experiments.



(a) Type I, Uniform (b) Type II, Clustered (c) Type III, Mixed

Figure 5.1: Random Data with $n = 30$

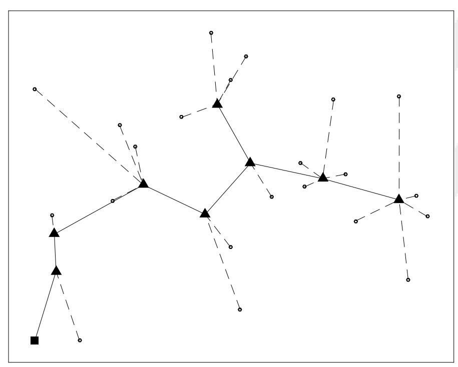
For each dataset, we used different parameter settings depending on the size of the dataset so that we can observe different types of the facilities in the solution. In addition, we assumed that the source station is located south west corner of the instances. This assumption is valid for the rest of the chapter.

The results of the experiments are presented in Table 5.1. The first 6 columns represent the parameter settings of the instances. The seventh column named *Objective* indicates the obtained solution where *# of CF* represents the number of opened centralized facilities and *# of DF* represents the total number of opened decentralized facilities. *Gap*, in percentage format, indicates the optimality gap of the objective value. *Time*, in seconds, is the CPU time of the solution recorded from IBM CPLEX Optimizer. The last column titled *Comparison (%)* is the percentage of ("2LDN objective" - "Heuristic objective") / "2LDN objective".

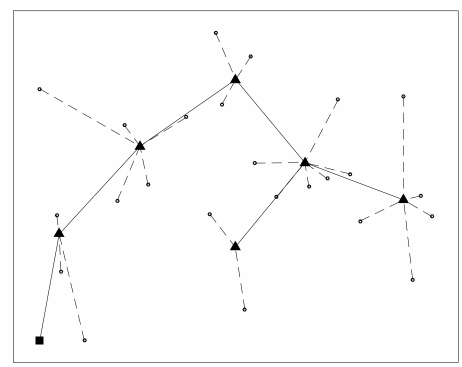
First of all, we observed a huge computational time advantage of our heuristic against 2LDN. The maximum solution time observed for the instances in Table 5.1 is 24.65 seconds, while the average time that our heuristic required is 12.06 seconds. The worst objective value that our heuristic algorithm finds for an instance is 5.57% worse than the optimal value and on the average objective values obtained by the heuristic algorithm is 1.45% worse.

In Table 5.1, we observe that our heuristic can acquire all possible solution alternatives: centralized, decentralized and hybrid solutions. As expected, heuristic algorithm finds the optimal solution for the instances with a decentralized solution. Relatively poor, performances of the heuristic algorithm, such as 5.57%, 5.61%, 3.08% away from the optimal, belong to the instances that have a centralized solution. That happens because the heuristic algorithm does not consider the upper-level connections when it designs the lower-level network. Thereby, heuristic algorithm can find more or less number of decentralized facilities than the optimal solution. The two of such instances are graphed in Figure 5.2 to demonstrate the difference. The maps of the instances provided in the rest of the thesis have the following legend:

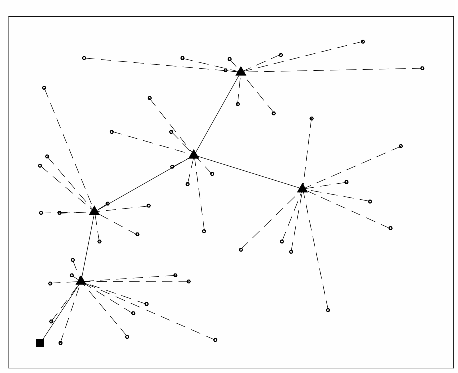
- Small circles represent demand nodes
- Triangles represent centralized facilities
- Big circles represent decentralized facilities
- Dashed lines represent lower-level network connections
- Regular lines represent upper-level network connections
- Square represents the source node



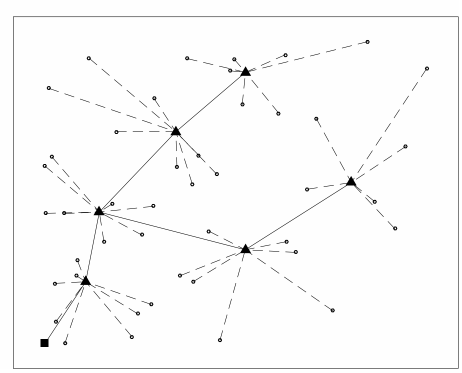
(a) 2LDN solution of the $3^{rd}$ instance
8 centralized facilities



(b) Heuristic solution of the $3^{rd}$ instance:
6 centralized facilities (5.57% worse)



(c) 2LDN solution of the $23^{th}$ instance:
5 centralized facilities



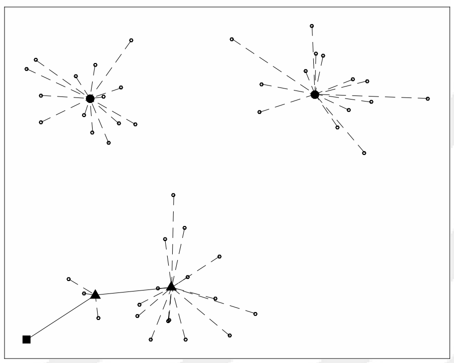
(d) Heuristic solution of the $23^{th}$ inst.:
6 centralized facilities (1.14% worse)

Figure 5.2: Comparison of the Heuristic and Optimal Solution for Centralized Solutions

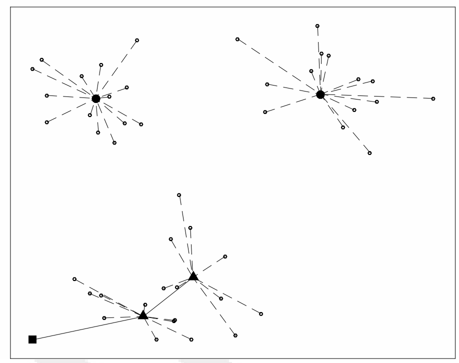
Table 5.1: Numerical Analysis of the Heuristic Algorithm with Random Data

| Scenario Setting | | | | | | 2LDN | | | | | Heuristic Algorithm | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Pattern** | **n** | $C_{CF}$ | $C_{DF}$ | $C_{LN}$ | $C_{UN}$ | **Objective** | **# of CF** | **# of DF** | **Gap (%)** | **Time (sec.)** | **Objective** | **# of CF** | **# of DF** | **Time (sec.)** | **Comparison (%)** |
| Type I | 30 | 2000 | 10000 | 1 | 2 | 109785 | 1 | 4 | 0 | 412.9 | 111203 | 0 | 5 | 19.93 | -1.29 |
| Type I | 30 | 2000 | 10000 | 1 | 4 | 111203 | 0 | 5 | 0 | 17.34 | 111203 | 0 | 5 | 2.52 | 0 |
| Type I | 30 | 2000 | 15000 | 1 | 2 | 114739 | 8 | 0 | 0 | 1121.76 | 121131 | 6 | 0 | 19.04 | -5.57 |
| Type I | 30 | 2000 | 15000 | 1 | 4 | 131431 | 0 | 3 | 0 | 94.91 | 131431 | 0 | 3 | 23.57 | 0 |
| Type I | 30 | 2000 | 20000 | 1 | 2 | 114739 | 8 | 0 | 0 | 1247.03 | 121131 | 6 | 0 | 12.46 | -5.57 |
| Type I | 30 | 2000 | 20000 | 1 | 4 | 143297 | 1 | 2 | 0 | 230.02 | 146431 | 0 | 3 | 18.7 | -2.19 |
| Type II | 30 | 2000 | 10000 | 1 | 2 | 73699 | 2 | 2 | 0 | 479.83 | 74180 | 1 | 2 | 12.19 | -0.65 |
| Type II | 30 | 2000 | 10000 | 1 | 4 | 75661 | 0 | 3 | 0 | 57.23 | 75661 | 0 | 3 | 4.43 | 0 |
| Type II | 30 | 2000 | 15000 | 1 | 2 | 83699 | 2 | 2 | 0 | 2123.86 | 84180 | 1 | 2 | 19.1 | -0.57 |
| Type II | 30 | 2000 | 15000 | 1 | 4 | 90026 | 1 | 2 | 0 | 104 | 90661 | 0 | 3 | 2.58 | -0.71 |
| Type II | 30 | 2000 | 20000 | 1 | 2 | 88706 | 5 | 0 | 0 | 5651.03 | 89531 | 5 | 0 | 19.76 | -0.93 |
| Type II | 30 | 2000 | 20000 | 1 | 4 | 100026 | 1 | 2 | 0 | 461.49 | 100699 | 1 | 2 | 19.09 | -0.67 |
| Type III | 30 | 2000 | 10000 | 1 | 2 | 85552 | 2 | 2 | 0 | 214.36 | 85909 | 2 | 2 | 17.32 | -0.42 |
| Type III | 30 | 2000 | 10000 | 1 | 4 | 89662 | 1 | 3 | 0 | 63.38 | 89947 | 0 | 3 | 2.45 | -0.32 |
| Type III | 30 | 2000 | 15000 | 1 | 2 | 95127 | 6 | 1 | 0 | 527.42 | 95909 | 2 | 2 | 23.53 | -0.82 |
| Type III | 30 | 2000 | 15000 | 1 | 4 | 104662 | 1 | 3 | 0 | 78.52 | 104947 | 0 | 3 | 13.4 | -0.27 |
| Type III | 30 | 2000 | 20000 | 1 | 2 | 97946 | 7 | 0 | 0 | 612.84 | 103436 | 4 | 0 | 22.16 | -5.61 |
| Type III | 30 | 2000 | 20000 | 1 | 4 | 115018 | 2 | 2 | 0 | 230.01 | 117421 | 2 | 2 | 20.11 | -2.09 |
| Type I | 50 | 5000 | 15000 | 1 | 2.5 | 169359 | 4 | 1 | 0 | 43289.65 | 171068 | 4 | 2 | 11.16 | -1.01 |
| Type I | 50 | 5000 | 15000 | 1 | 5 | 174097 | 1 | 3 | 0 | 117.42 | 174643 | 0 | 4 | 12.91 | -0.31 |
| Type I | 50 | 5000 | 20000 | 1 | 2.5 | 172103 | 5 | 0 | 1.86 | 90000 | 174059 | 6 | 0 | 8.53 | -1.14 |
| Type I | 50 | 5000 | 20000 | 1 | 5 | 189097 | 1 | 3 | 0 | 789.24 | 194280 | 0 | 3 | 12.8 | -2.74 |
| Type I | 50 | 5000 | 30000 | 1 | 2.5 | 172103 | 5 | 0 | 0 | 86710.63 | 174059 | 6 | 0 | 24.65 | -1.14 |
| Type I | 50 | 5000 | 30000 | 1 | 5 | 210182 | 4 | 0 | 0 | 47741.25 | 221822 | 1 | 2 | 23.74 | -5.54 |
| Type II | 50 | 5000 | 15000 | 1 | 2.5 | 168554 | 0 | 3 | 0 | 2597.07 | 168554 | 0 | 3 | 7.14 | 0 |
| Type II | 50 | 5000 | 15000 | 1 | 5 | 168554 | 0 | 3 | 0 | 50.54 | 168554 | 0 | 3 | 2.14 | 0 |
| Type II | 50 | 5000 | 20000 | 1 | 2.5 | 179873 | 2 | 2 | 0 | 35377.74 | 181044 | 2 | 2 | 6.57 | -0.65 |
| Type II | 50 | 5000 | 20000 | 1 | 5 | 183554 | 0 | 3 | 0 | 173.18 | 183554 | 0 | 3 | 2.32 | 0 |
| Type II | 50 | 5000 | 30000 | 1 | 2.5 | 197012 | 7 | 0 | 8.65 | 90000 | 201044 | 2 | 2 | 7.7 | -2.05 |
| Type II | 50 | 5000 | 30000 | 1 | 5 | 213554 | 0 | 3 | 0 | 3145.61 | 213554 | 0 | 3 | 7.46 | 0 |
| Type III | 50 | 5000 | 15000 | 1 | 2.5 | 171823 | 1 | 5 | 0 | 599.67 | 174445 | 1 | 5 | 5.68 | -1.53 |
| Type III | 50 | 5000 | 15000 | 1 | 5 | 176097 | 0 | 6 | 0 | 47.39 | 176098 | 0 | 6 | 3.61 | 0 |
| Type III | 50 | 5000 | 20000 | 1 | 2.5 | 194597 | 4 | 3 | 0 | 10295.56 | 199281 | 2 | 4 | 6.73 | -2.41 |
| Type III | 50 | 5000 | 20000 | 1 | 5 | 202226 | 1 | 5 | 0 | 301.22 | 206098 | 0 | 6 | 5.18 | -1.91 |
| Type III | 50 | 5000 | 30000 | 1 | 2.5 | 211943 | 7 | 0 | 4.94 | 90000 | 218469 | 6 | 0 | 7.29 | -3.08 |
| Type III | 50 | 5000 | 30000 | 1 | 5 | 247872 | 1 | 4 | 0 | 5236.66 | 250589 | 1 | 4 | 6.22 | -1.1 |

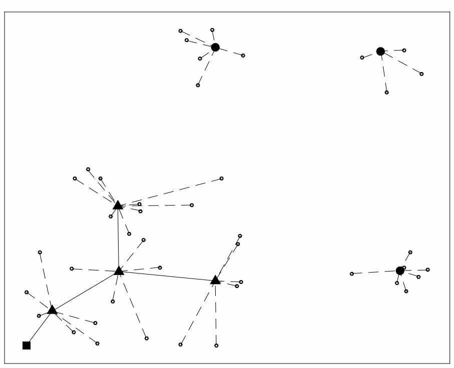
Furthermore, our heuristic algorithm also provides a hybrid solution for the instances that have a hybrid network as an optimal solution. However, it could not find the optimal solution when i) it finds a different number of facilities ii) it finds different locations for the facilities. One example for each case is graphed in Figure 5.3 to demonstrate the difference.



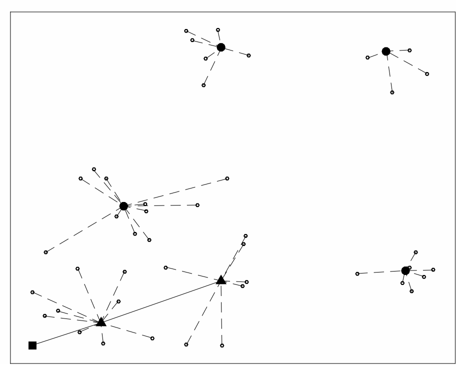
(a) 2LDN solution of the $27^{th}$ instance: 2 centralized and 2 decentralized facilities



(b) Heuristic solution of the $27^{th}$ instance: 2 centralized and 2 decentralized facilities (0.65% worse)



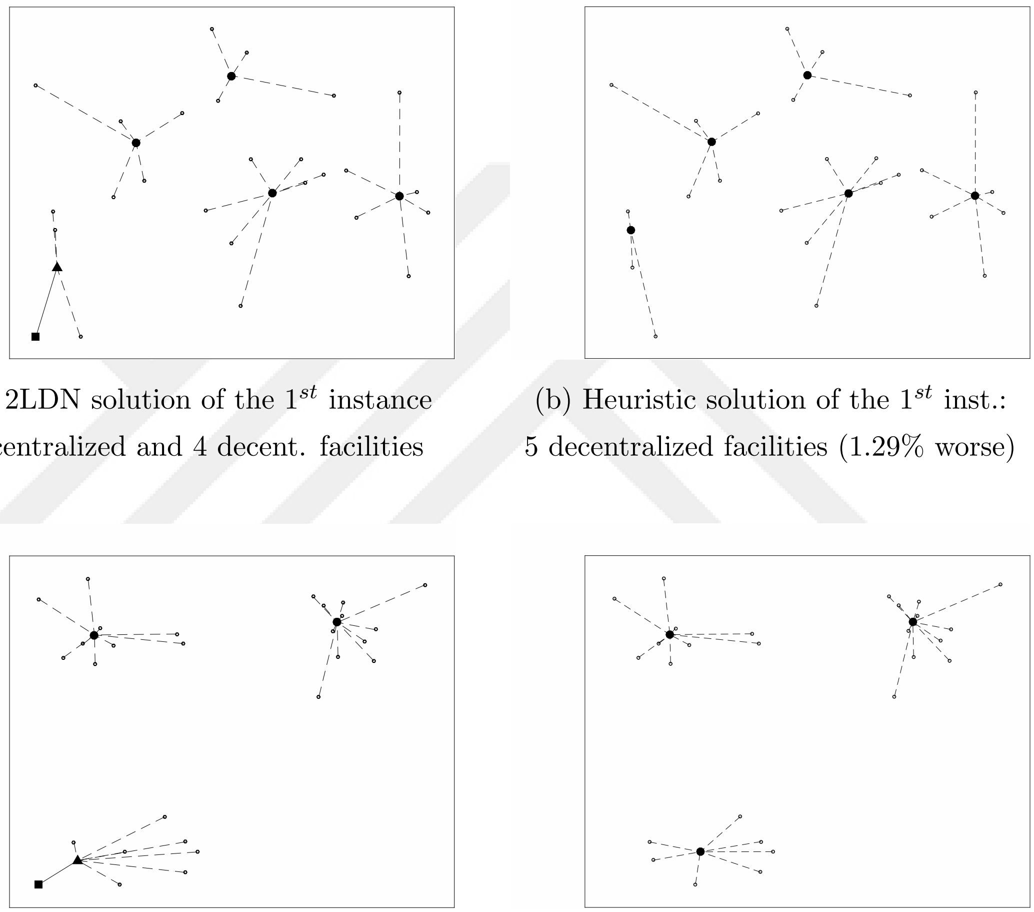
(c) 2LDN solution of the $33^{th}$ instance: 4 centralized and 3 decentralized facilities



(d) Heuristic solution of the $33^{th}$ instance: 2 centralized and 4 decentralized facilities (2.41% worse)

Figure 5.3: Comparison of the Heuristic and Optimal Solution for Hybrid Solutions

Heuristic algorithm could not detect a hybrid solution for the instances that have demand points very close to the source node, although the rest of the demand points is dispersed, such as in Figure 5.4.



(a) 2LDN solution of the $1^{st}$ instance
1 centralized and 4 decent. facilities

(b) Heuristic solution of the $1^{st}$ inst.:
5 decentralized facilities (1.29% worse)

(c) 2LDN solution of the $10^{th}$ inst.:
1 centralized and 2 decent. facilities

(d) Heuristic solution of the $10^{th}$ inst.:
3 decentralized facilities (0.71% worse)

Figure 5.4: Comparison of Heuristic with Optimal for Decentralized solutions

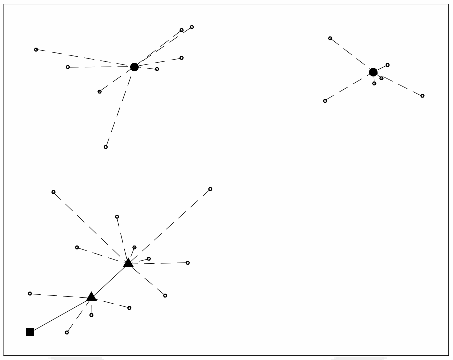
The reason of heuristic algorithm missing such cases is that it does not consider the location of the source node while deciding the locations of the facilities. In order to detect such cases, we proposed a modification to the Step 2 of our heuristic, which is the step that lower-level network is designed. The modified heuristic algorithm generates dummy demand nodes on the location of the source before solving the lower-level network. The number of the dummy nodes generated is equal to $C_{UN}/C_{LN}$ in order to create enough bias towards the source node. After designing the lover-level, dummy nodes and their connection costs are deleted. If a facility is opened on one of these dummy nodes, it is deleted as well and the original demand nodes connected to that facility are marked. Then, 1-Median is solved for those demand nodes. The rest of the steps are the same with the one presented in Section 4.

In the modified heuristic, since we create bias towards the source node, it is expected that the modification will negatively affect the cost of the decentralized networks configurations that the original heuristic finds. Therefore, we also propose a third approach that we called extended heuristic, which solves the original and the modified heuristic and selects the best solution. Results of the modified and the extended versions of the heuristic algorithm are reported in Table 5.2. The results of the experiments conducted with the same instances are depicted in Table 5.1. Modified heuristic could detect the cases discussed in Figure 5.4. However, the maximum difference between the optimal solution and the result of the heuristic approach is increased to 7.57% from 5.57% and the average objective deviation is increased from 1.45% to 1.73 %. On the other hand, the extended version provides objective values, on the average, 0.97% worse than the optimal. Yet, the worst deviation from the optimal is not improved and remained at 5.57% and, as expected, solution times almost doubled. Therefore, we decided to continue with the original heuristic.

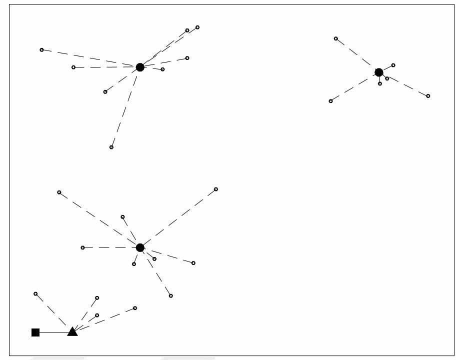
So as to validate the model and understand the dynamics between the parameter ratios, a sensitivity analysis is constructed. In Table 5.1 we observe that as the $C_{UN}/C_{LN}$ ratio increases, solutions include higher number of decentralized facilities. On the other hand, as the $C_{DF}/C_{CF}$ ratio increases, solutions include higher number of centralized facilities. The effect of the change in the $C_{DF}/C_{LN}$ ratio can not be observed in the Table 5.1 and will be discussed in the next section. To clarify the findings listed above, some of the instances are presented in more detail with Figures 5.5 and 5.6.

Table 5.2: Numerical Analysis of the Modified and the Extended Heuristic Algorithms

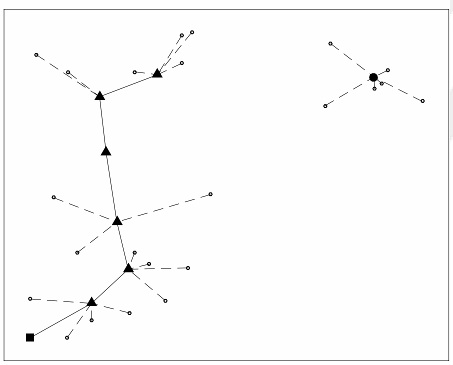
| Scenario Setting | | | | | | Modified Heuristic Algorithm | | | | | Extended Heuristic Algorithm | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Pattern** | **n** | $C_{CF}$ | $C_{DF}$ | $C_{LN}$ | $C_{UN}$ | **Objective** | **# of CF** | **# of DF** | **Time (sec.)** | **Comparison (%)** | **Objective** | **# of CF** | **# of DF** | **Time (sec.)** | **Comparison (%)** |
| Type I | 30 | 2000 | 10000 | 1 | 2 | 111611 | 1 | 5 | 20.77 | -1.66 | 111203 | 2 | 3 | 44.96 | **-1.29** |
| Type I | 30 | 2000 | 10000 | 1 | 4 | 114993 | 1 | 5 | 4.48 | -3.41 | 111203 | 0 | 5 | 8.08 | **0.00** |
| Type I | 30 | 2000 | 15000 | 1 | 2 | 121539 | 7 | 0 | 33.25 | -5.93 | 121131 | 6 | 0 | 48.33 | **-5.57** |
| Type I | 30 | 2000 | 15000 | 1 | 4 | 133898 | 0 | 3 | 29.81 | -1.88 | 131431 | 0 | 3 | 48.98 | **0.00** |
| Type I | 30 | 2000 | 20000 | 1 | 2 | 121539 | 7 | 0 | 23.35 | -5.93 | 121131 | 6 | 0 | 41.57 | **-5.57** |
| Type I | 30 | 2000 | 20000 | 1 | 4 | 148898 | 0 | 3 | 30.31 | -3.91 | 146431 | 0 | 3 | 44.63 | **-2.19** |
| Type II | 30 | 2000 | 10000 | 1 | 2 | 74180 | 1 | 2 | 7 | **-0.65** | 74180 | 1 | 2 | 17.35 | **-0.65** |
| Type II | 30 | 2000 | 10000 | 1 | 4 | 78566 | 1 | 3 | 6.2 | -3.84 | 75661 | 1 | 2 | 9.93 | **0.00** |
| Type II | 30 | 2000 | 15000 | 1 | 2 | 84180 | 1 | 2 | 17.43 | **-0.57** | 84180 | 1 | 2 | 35.4 | **-0.57** |
| Type II | 30 | 2000 | 15000 | 1 | 4 | 90026 | 1 | 2 | 5.32 | **0.00** | 90026 | 1 | 2 | 6.96 | **0.00** |
| Type II | 30 | 2000 | 20000 | 1 | 2 | 88853 | 5 | 0 | 16.31 | **-0.17** | 88853 | 5 | 0 | 30.96 | **-0.17** |
| Type II | 30 | 2000 | 20000 | 1 | 4 | 100026 | 1 | 2 | 15.48 | **0.00** | 100026 | 1 | 2 | 28.41 | **0.00** |
| Type III | 30 | 2000 | 10000 | 1 | 2 | 86914 | 1 | 3 | 21.84 | -1.59 | 85909 | 1 | 3 | 45.34 | **-0.42** |
| Type III | 30 | 2000 | 10000 | 1 | 4 | 90530 | 1 | 3 | 6.98 | -0.97 | 89947 | 0 | 3 | 9.65 | **-0.32** |
| Type III | 30 | 2000 | 15000 | 1 | 2 | 97833 | 2 | 2 | 25.58 | -2.84 | 95909 | 2 | 2 | 42.66 | **-0.82** |
| Type III | 30 | 2000 | 15000 | 1 | 4 | 105530 | 1 | 3 | 23.5 | -0.83 | 104947 | 1 | 2 | 42.48 | **-0.27** |
| Type III | 30 | 2000 | 20000 | 1 | 2 | 105360 | 4 | 0 | 25.29 | -7.57 | 103436 | 4 | 0 | 47.28 | **-5.61** |
| Type III | 30 | 2000 | 20000 | 1 | 4 | 117294 | 1 | 2 | 21.49 | **-1.98** | 117294 | 1 | 2 | 41.76 | **-1.98** |
| Type I | 50 | 5000 | 15000 | 1 | 2.5 | 171739 | 1 | 3 | 9.04 | -1.41 | 171068 | 4 | 2 | 14.5 | **-1.01** |
| Type I | 50 | 5000 | 15000 | 1 | 5 | 174094 | 1 | 3 | 9.52 | **0.00** | 174094 | 1 | 3 | 16.26 | **0.00** |
| Type I | 50 | 5000 | 20000 | 1 | 2.5 | 176756 | 6 | 0 | 10.84 | -2.70 | 174059 | 6 | 0 | 15.03 | **-1.14** |
| Type I | 50 | 5000 | 20000 | 1 | 5 | 189094 | 1 | 3 | 6.23 | **0.00** | 189094 | 1 | 3 | 14.14 | **0.00** |
| Type I | 50 | 5000 | 30000 | 1 | 2.5 | 176756 | 6 | 0 | 16.94 | -2.70 | 174059 | 6 | 0 | 29.27 | **-1.14** |
| Type I | 50 | 5000 | 30000 | 1 | 5 | 214553 | 1 | 2 | 15.7 | **-2.08** | 214553 | 1 | 2 | 30.69 | **-2.08** |
| Type II | 50 | 5000 | 15000 | 1 | 2.5 | 169943 | 1 | 3 | 8.96 | -0.82 | 168553 | 0 | 3 | 12.44 | **0.00** |
| Type II | 50 | 5000 | 15000 | 1 | 5 | 171125 | 0 | 3 | 2.79 | -1.53 | 168553 | 0 | 3 | 4.42 | **0.00** |
| Type II | 50 | 5000 | 20000 | 1 | 2.5 | 180083 | 2 | 2 | 3.78 | **-0.12** | 180083 | 2 | 2 | 10.41 | **-0.12** |
| Type II | 50 | 5000 | 20000 | 1 | 5 | 186125 | 0 | 3 | 2.96 | -1.40 | 183553 | 0 | 3 | 4.76 | **0.00** |
| Type II | 50 | 5000 | 30000 | 1 | 2.5 | 200083 | 2 | 2 | 9.05 | **-1.56** | 200083 | 2 | 2 | 13.1 | **-1.56** |
| Type II | 50 | 5000 | 30000 | 1 | 5 | 216125 | 0 | 3 | 3.2 | -1.20 | 213553 | 0 | 3 | 10.63 | **0.00** |
| Type III | 50 | 5000 | 15000 | 1 | 2.5 | 171825 | 1 | 5 | 10.41 | **0.00** | 171825 | 1 | 5 | 16.35 | **0.00** |
| Type III | 50 | 5000 | 15000 | 1 | 5 | 177226 | 1 | 5 | 12.37 | -0.64 | 176098 | 1 | 5 | 16.9 | **0.00** |
| Type III | 50 | 5000 | 20000 | 1 | 2.5 | 196825 | 1 | 5 | 11.51 | **-1.15** | 196825 | 1 | 5 | 13.44 | **-1.15** |
| Type III | 50 | 5000 | 20000 | 1 | 5 | 202226 | 1 | 5 | 17.43 | **0.00** | 202226 | 1 | 5 | 20.81 | **0.00** |
| Type III | 50 | 5000 | 30000 | 1 | 2.5 | 214758 | 7 | 0 | 9.18 | **-1.33** | 214758 | 7 | 0 | 20.69 | **-1.33** |
| Type III | 50 | 5000 | 30000 | 1 | 5 | 247873 | 1 | 4 | 12.62 | **0.00** | 247873 | 1 | 4 | 19.63 | **0.00** |



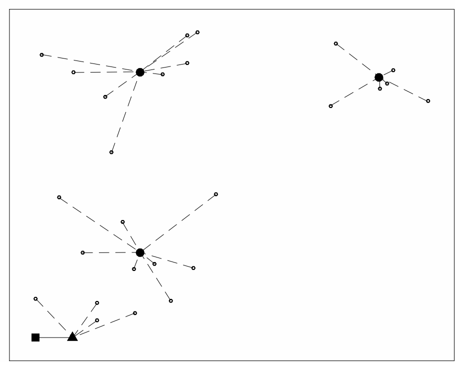
(a) $C_{CF} = 2k$, $C_{DF} = 10k$
$C_{LN} = 1$, $C_{UN} = 2$



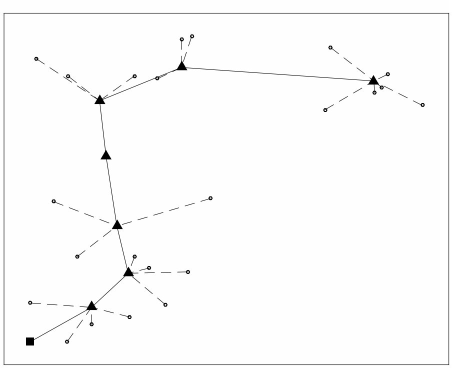
(b) $C_{CF} = 2k$, $C_{DF} = 10k$
$C_{LN} = 1$, $C_{UN} = 4$



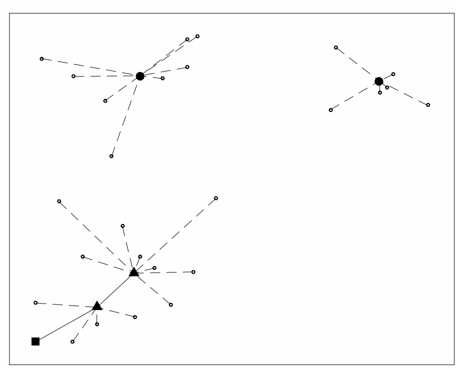
(c) $C_{CF} = 2k$, $C_{DF} = 15k$
$C_{LN} = 1$, $C_{UN} = 2$



(d) $C_{CF} = 2k$, $C_{DF} = 15k$
$C_{LN} = 1$, $C_{UN} = 4$



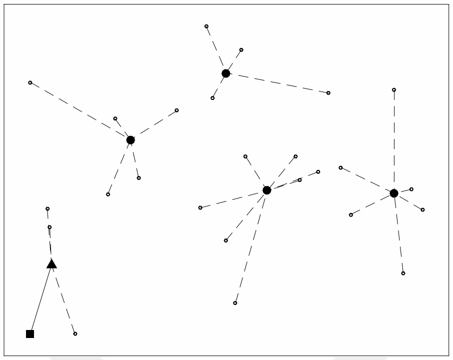
(e) $C_{CF} = 2k$, $C_{DF} = 20k$
$C_{LN} = 1$, $C_{UN} = 2$



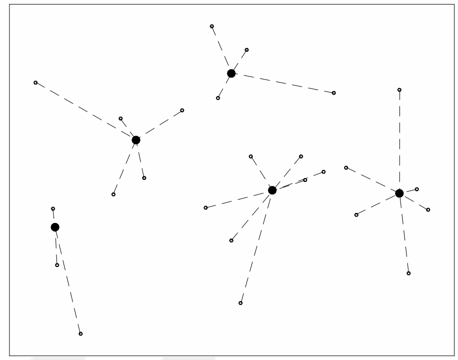
(f) $C_{CF} = 2k$, $C_{DF} = 20k$
$C_{LN} = 1$, $C_{UN} = 4$

Figure 5.5: Results of the Type III (Mixed) Random Data with $n = 30$

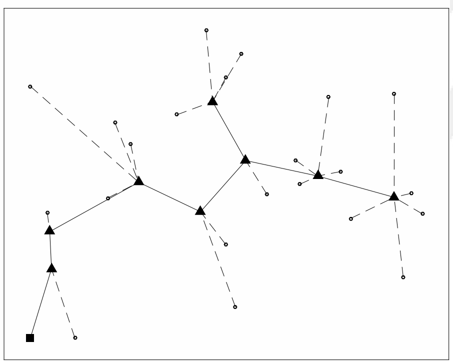
In Figures 5.5 and 5.6, the first, second and third row of the figures present the solutions of the instances where the $C_{DF}/C_{CF}$ ratio equals to 5, 7.5 and 10 respectively. In addition, the first and second rows of the figures present the solutions of the instances where the $C_{UN}/C_{LN}$ ratio equals to 2 and 4 accordingly. In the figures, the total number of decentralized facilities increase at each column and the total number of centralized facilities generally increase at rows. In the cases where the total number of centralized facilities is not increased, the $C_{UN}/C_{LN}$ ratio dominates the effect of the $C_{DF}/C_{CF}$. Therefore, we conclude that the ratios $C_{DF}/C_{CF}$ and $C_{UN}/C_{LN}$ are competing with each other in terms of the total number of centralized and decentralized facilities.



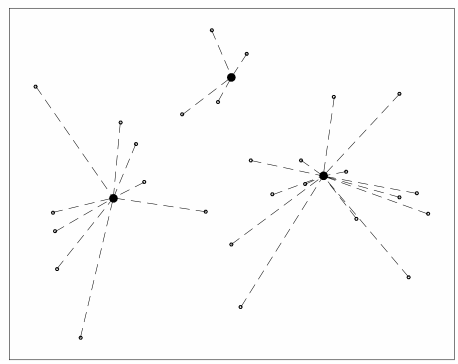
(a) $C_{CF} = 2k$, $C_{DF} = 10k$
$C_{LN} = 1$, $C_{UN} = 2$



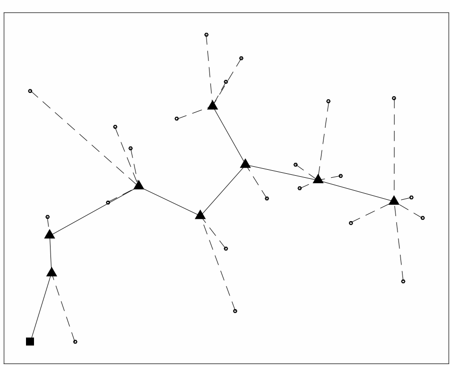
(b) $C_{CF} = 2k$, $C_{DF} = 10k$
$C_{LN} = 1$, $C_{UN} = 4$



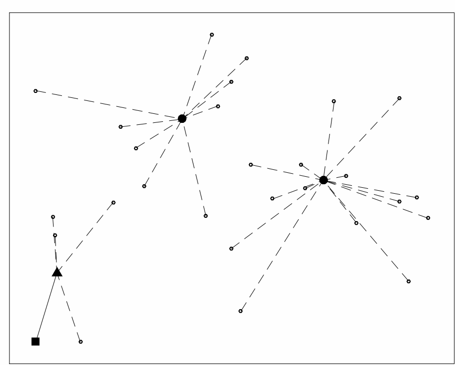
(c) $C_{CF} = 2k$, $C_{DF} = 15k$
$C_{LN} = 1$, $C_{UN} = 2$



(d) $C_{CF} = 2k$, $C_{DF} = 15k$
$C_{LN} = 1$, $C_{UN} = 4$



(e) $C_{CF} = 2k$, $C_{DF} = 20k$
$C_{LN} = 1$, $C_{UN} = 2$



(f) $C_{CF} = 2k$, $C_{DF} = 20k$
$C_{LN} = 1$, $C_{UN} = 4$

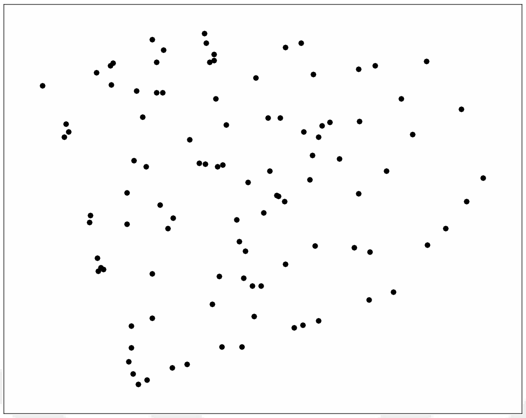
Figure 5.6: Results of the Type I (Uniform) Random Data with $n = 30$

## 5.2 Experiments with Data from Network Planner

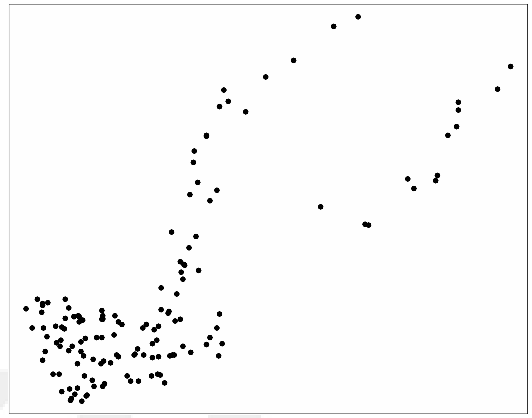
In the sequel of comparing our heuristic with 2LND using random data, in this section, we use real data which we obtained from an open source tool named Network Planner which is used in national electrification studies [52]. We obtained latitude and longitude information of 4 different villages with different sizes and spatial distribution patterns named *LeonaG*, *BOL*, *Guinea* and *LGA*.

The aim of this section is to test the performance of our heuristic algorithm with bigger instances for which we can not obtain optimal solutions to 2LNDP model and to see how our heuristic algorithm behaves under different circumstances.

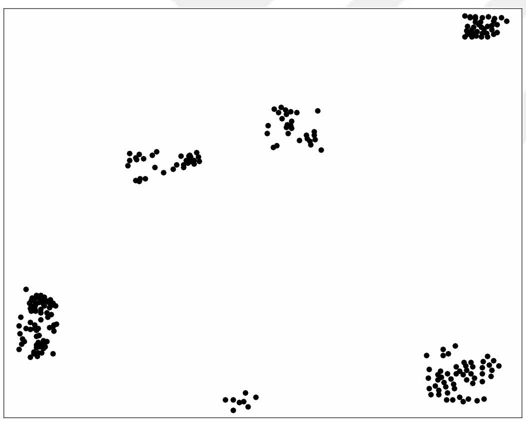
We projected their latitude and longitude coordinates on a plane using Universal Transverse Mercator(UTM) transformation [53], as in Figure 5.7. Selected instances have different spatial distribution patterns: *LeonaG* has 102 nodes which are uniformly distributed on a $22.4km \times 24.1km$ area. *BOL* has 141 nodes which are reclined through the bottom-left corner of a $55.6km \times 61.4km$ area, having dense and sparse parts. *Guinea* has clustered 230 nodes distributed on a $63.4km \times 101.8km$ area. It has six highly dense and separated clusters. Lastly, *LGA* has 274 nodes distributed on a $59.4km \times 76.9km$ area. Although, the size of the instances obtained from the Network Planner are too big to obtain exact solutions, their diverse characteristics is an opportunity to analyze how our heuristic behaves under different circumstances.



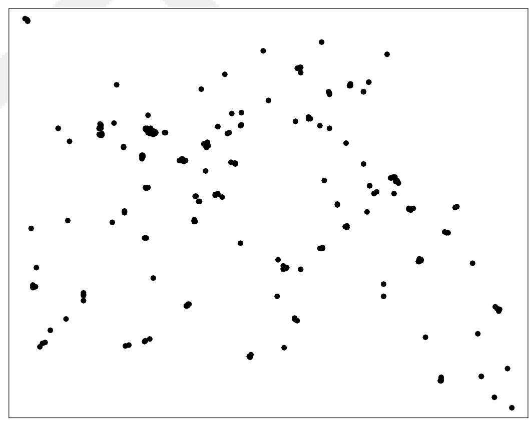
(a) LeonaG, n= 102,
22.4$km$ × 24.1$km$



(b) BOL, n= 141,
55.6$km$ × 61.4$km$



(c) Guinea, n= 230,
63.4$km$ × 101.8$km$



(d) LGA, n= 274,
59.4$km$ × 76.9$km$

Figure 5.7: Maps Obtained from Network Planner

In this section, more parameter combinations are used than the previous sections. While experimenting with real data, we employ the combination of the ratios which are generated based on the parameters used in [20] as listed below;

- $C_{DF}/C_{CF}$: 2, 3, 4, 5 and 10, which means in practice opening a decentralized facility costs 2, 3 . . . times of opening a centralized facility.

- $C_{CF}/C_{LN}$: 50000 and 5000, which means opening a centralized facility costs establishing 50000 and 5000 unit of lower-level network connection.

- $C_{UN}/C_{LN}$: 2.5 and 5, which means establishing one unit of upper-level network connection costs 2.5 and 5 times of establishing one unit of lower-level network connection.

Combination of the above mentioned ratios yields 20 instances as listed in Table 5.3. We employ this parameter set with each data obtained from the Network Planner to conduct our experiments. We refer to the instances numbered from 1 to 10 as the *first batch* and the instances numbered from 11 to 20 as the *second batch* in the rest of the thesis.

Table 5.3: Parameter Set

| Parameters | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| # | $C_{CF}$ | $C_{DF}$ | $C_{LN}$ | $C_{UN}$ | # | $C_{CF}$ | $C_{DF}$ | $C_{LN}$ | $C_{UN}$ |
| 1 | 5000 | 10000 | 1 | 2.5 | 11 | 50000 | 100000 | 1 | 2.5 |
| 2 | 5000 | 10000 | 1 | 5 | 12 | 50000 | 100000 | 1 | 5 |
| 3 | 5000 | 15000 | 1 | 2.5 | 13 | 50000 | 150000 | 1 | 2.5 |
| 4 | 5000 | 15000 | 1 | 5 | 14 | 50000 | 150000 | 1 | 5 |
| 5 | 5000 | 20000 | 1 | 2.5 | 15 | 50000 | 200000 | 1 | 2.5 |
| 6 | 5000 | 20000 | 1 | 5 | 16 | 50000 | 200000 | 1 | 5 |
| 7 | 5000 | 30000 | 1 | 2.5 | 17 | 50000 | 300000 | 1 | 2.5 |
| 8 | 5000 | 30000 | 1 | 5 | 18 | 50000 | 300000 | 1 | 5 |
| 9 | 5000 | 50000 | 1 | 2.5 | 19 | 50000 | 500000 | 1 | 2.5 |
| 10 | 5000 | 50000 | 1 | 5 | 20 | 50000 | 500000 | 1 | 5 |

The results of the experiments conducted with *LeonaG* are reported in Table 5.4. The first five columns represent the parameters for the instance. Then, it continues with the results of 2LDN in the order of objective value, total number of centralized facilities and decentralized facilities, optimality gap and computation time. Notice that, the time limit for this and every experiment reported in this chapter is 90000 seconds and we measure CPU time instead of wall time. Likewise, the following four columns are the results of the heuristic algorithm. Finally, the last column, *Comparison(%)*, is the comparison of the objective values obtained by 2LDN and the heuristic, calculated as ("2LDN objective" - "Heuristic objective") / "2LDN objective".

We examine the performance of the heuristic algorithm in terms of both the objective value and the computational time. Then, we perform a sensitivity analysis similar to Section 5.1.

As seen in Table 5.4, the solution that the heuristic algorithm finds for an instance is at most 2.8 percent worse than the objective value found by 2LDN. On the other hand, the heuristic algorithm provides a better objective than 2LDN by up to 6.9%.

As expected, our heuristic requires less computational time than 2LDN. On the average, the heuristic requires about 200 seconds for the first batch of the instances. On the other hand, the second batch of the instances, which have a higher $C_{CF}/C_{LN}$ ratio than the first batch, takes about 30 seconds, on the average, to be solved by the heuristic algorithm. Further, *Time* is increasing within the instances numbered from 1 to 10 and 11 to 20 The reason is, as the difference between $C_{CF}$ and $C_{DF}$ increases, the difference between *pmax* and *pmin* also increases. Therefore, the algorithm performs more iterations that results in a higher computation time, see Table B.1.

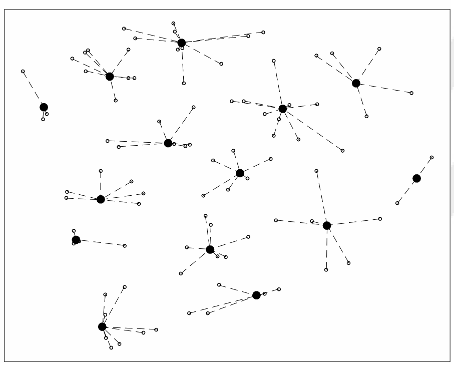
Besides, we realize that, on the average, about 70 percent of the CPU time measured for the heuristic algorithm consists of auxiliary operations such as preparing matrices to send to the IBM CPLEX Optimizer within MATLAB. The sum of the CPU times for only solving linear programs with IBM CPLEX Optimizer is reported in Table B.2.

To continue with the sensitivity analysis of the parameters, we observed that while $C_{DF}/C_{CF}$ and $C_{UN}/C_{LN}$ compete with each other in terms of the number of centralized and decentralized facilities, the $C_{CF}/C_{LN}$ ratio determines the effect of which ratio($C_{DF}/C_{CF}$ or $C_{UN}/C_{LN}$) will be more dominant.

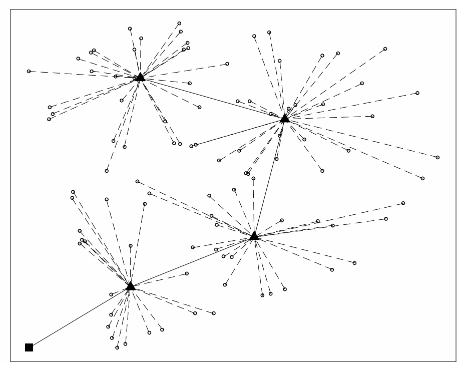
Table 5.4: Results of the Experiment with *LeonaG*

| Scenario Setting | | | | | 2LDN | | | | | Heuristic Algorithm | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **n** | $C_{CF}$ | $C_{DF}$ | $C_{LN}$ | $C_{UN}$ | **Objective** | **# of CF** | **# of DF** | **Gap (%)** | **Time (sec.)** | **Objective** | **# of CF** | **# of DF** | **Time (sec.)** | **Comparison (%)** |
| 102 | 5000 | 10000 | 1 | 2.5 | 318231 | 0 | 14 | 0 | 833.34 | 318231 | 0 | 14 | 73.41 | 0.0 |
| 102 | 5000 | 10000 | 1 | 5 | 318231 | 0 | 14 | 0 | 55.79 | 318231 | 0 | 14 | 72.07 | 0.0 |
| 102 | 5000 | 15000 | 1 | 2.5 | 395623 | 1 | 12 | 13.75 | 90000 | 378347 | 0 | 10 | 260.15 | 4.4 |
| 102 | 5000 | 15000 | 1 | 5 | 378347 | 0 | 10 | 0.01 | 6395.28 | 378347 | 0 | 10 | 92.11 | 0.0 |
| 102 | 5000 | 20000 | 1 | 2.5 | 429686 | 13 | 2 | 20.02 | 90000 | 419803 | 9 | 3 | 263.49 | 2.3 |
| 102 | 5000 | 20000 | 1 | 5 | 431967 | 0 | 9 | 8.09 | 90000 | 424207 | 0 | 8 | 103.76 | 1.8 |
| 102 | 5000 | 30000 | 1 | 2.5 | 433802 | 16 | 0 | 22.98 | 90000 | 427161 | 12 | 0 | 259.76 | 1.5 |
| 102 | 5000 | 30000 | 1 | 5 | 539035 | 4 | 4 | 27.27 | 90000 | 501782 | 1 | 6 | 343.65 | 6.9 |
| 102 | 5000 | 50000 | 1 | 2.5 | 419894 | 14 | 0 | 19.1 | 90000 | 427161 | 12 | 0 | 274.43 | -1.7 |
| 102 | 5000 | 50000 | 1 | 5 | 555973 | 11 | 0 | 28.97 | 90000 | 571367 | 9 | 0 | 272.08 | -2.8 |
| 102 | 50000 | 100000 | 1 | 2.5 | 697726 | 4 | 0 | 4.96 | 90000 | 703614 | 3 | 0 | 21.97 | -0.8 |
| 102 | 50000 | 100000 | 1 | 5 | 773094 | 0 | 3 | 13.28 | 90000 | 772841 | 0 | 3 | 22.44 | 0.0 |
| 102 | 50000 | 150000 | 1 | 2.5 | 697726 | 4 | 0 | 3.77 | 90000 | 703614 | 3 | 0 | 26.31 | -0.8 |
| 102 | 50000 | 150000 | 1 | 5 | 778522 | 4 | 0 | 13.7 | 90000 | 784386 | 3 | 0 | 26.89 | -0.8 |
| 102 | 50000 | 200000 | 1 | 2.5 | 697726 | 4 | 0 | 3.66 | 90000 | 703614 | 3 | 0 | 27.22 | -0.8 |
| 102 | 50000 | 200000 | 1 | 5 | 776999 | 4 | 0 | 10.53 | 90000 | 784386 | 3 | 0 | 26.74 | -1.0 |
| 102 | 50000 | 300000 | 1 | 2.5 | 697726 | 4 | 0 | 4.21 | 90000 | 703614 | 3 | 0 | 44.45 | -0.8 |
| 102 | 50000 | 300000 | 1 | 5 | 777027 | 4 | 0 | 12.45 | 90000 | 784386 | 3 | 0 | 45.41 | -0.9 |
| 102 | 50000 | 500000 | 1 | 2.5 | 697726 | 4 | 0 | 3.38 | 90000 | 703614 | 3 | 0 | 45.12 | -0.8 |
| 102 | 50000 | 500000 | 1 | 5 | 776140 | 4 | 0 | 10.93 | 90000 | 784386 | 3 | 0 | 46 | -1.1 |

For example, solutions of the $1^{st}$ and the $11^{th}$ instances of the *Leona* experiment are graphed in Figure 5.8. Both the $1^{st}$ and the $11^{th}$ instances have the same $C_{DF}/C_{CF}$ and $C_{UN}/C_{LN}$ ratios, however the $C_{CF}/C_{LN}$ ratio is equal to 5000 in the $1^{st}$ and 50000 in the $11^{th}$ instance. As a result, we observe a decentralized system in Figure 5.8a which has the lower $C_{CF}/C_{LN}$ ratio while we observe a centralized system in Figure 5.8b. The reason behind that is when $C_{CF}/C_{LN}$ is increased from 5000 to 50000, connection costs become less effective and the solution becomes a decentralized system. Our heuristic algorithm also detects the transition.



(a) 2LDN result of the $1^{st}$ instance where $C_{CF} = 5k$, $C_{DF} = 10k$, $C_{LN} = 1$ and $C_{UN} = 2.5$



(b) 2LDN result of the $11^{th}$ instance where $C_{CF} = 50k$, $C_{DF} = 100k$, $C_{LN} = 1$ and $C_{UN} = 2.5$

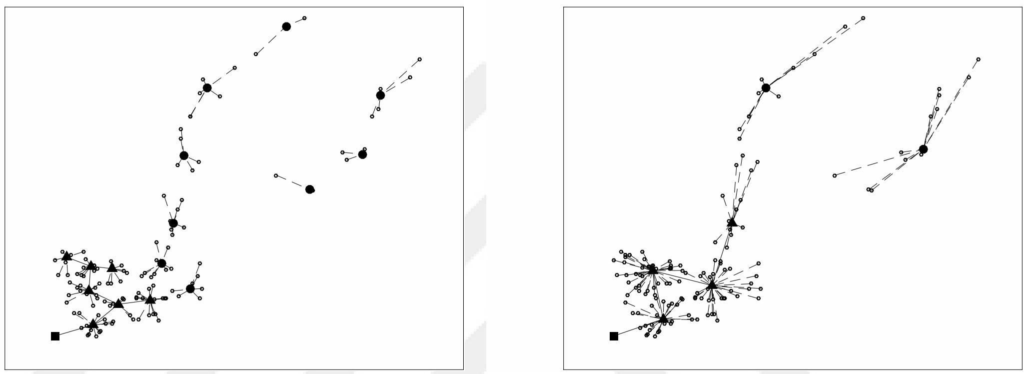
Figure 5.8: Solution Instances of *LeonaG* Experiment

The results of the experiment conducted with *BOL* are reported in Table 5.5. In this experiment, our heuristic finds a solution that is at most 1.5 percent worse than 2LDN and it finds a solution that is better than 2LDN by up to 9.5%. The heuristic algorithm requires about 300 seconds on the average. We observe an increase in the solution time when it is compared to *Leona*. The difference is due to the size of the subproblems solved in *BOL*, because the total number iterations performed for each dataset are almost the same, as seen in Table B.1.

Table 5.5: Results of the Experiment with *BOL*

| Scenario Setting | | | | | 2LDN | | | | | Heuristic Algorithm | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **n** | $C_{CF}$ | $C_{DF}$ | $C_{LN}$ | $C_{UN}$ | **Objective** | **# of CF** | **# of DF** | **Gap (%)** | **Time (sec.)** | **Objective** | **# of CF** | **# of DF** | **Time (sec.)** | **Comparison (%)** |
| 141 | 5000 | 10000 | 1 | 2.5 | 448754 | 0 | 21 | 0 | 10640 | 448754 | 0 | 21 | 165.86 | 0 |
| 141 | 5000 | 10000 | 1 | 5 | 448754 | 0 | 21 | 0 | 158.46 | 448754 | 0 | 21 | 158.97 | 0 |
| 141 | 5000 | 15000 | 1 | 2.5 | 556263 | 0 | 18 | 12.77 | 90000 | 540087 | 0 | 16 | 491.23 | 2.9 |
| 141 | 5000 | 15000 | 1 | 5 | 540087 | 0 | 16 | 0.01 | 15693 | 540087 | 0 | 16 | 223.9 | 0 |
| 141 | 5000 | 20000 | 1 | 2.5 | 594958 | 16 | 7 | 15.42 | 90000 | 598542 | 7 | 9 | 438.56 | -0.6 |
| 141 | 5000 | 20000 | 1 | 5 | 647207 | 1 | 16 | 12.83 | 90000 | 618037 | 0 | 15 | 259.58 | 4.5 |
| 141 | 5000 | 30000 | 1 | 2.5 | 663149 | 20 | 5 | 22.02 | 90000 | 666202 | 10 | 6 | 511.83 | -0.5 |
| 141 | 5000 | 30000 | 1 | 5 | 821651 | 17 | 6 | 26.37 | 90000 | 747708 | 0 | 12 | 539.02 | 9 |
| 141 | 5000 | 50000 | 1 | 2.5 | 715570 | 31 | 0 | 27.42 | 90000 | 717567 | 19 | 0 | 696.16 | -0.3 |
| 141 | 5000 | 50000 | 1 | 5 | 980057 | 29 | 3 | 35.31 | 90000 | 886831 | 5 | 5 | 618.73 | 9.5 |
| 141 | 50000 | 100000 | 1 | 2.5 | 1163195 | 6 | 1 | 13.18 | 90000 | 1146171 | 4 | 2 | 97.16 | 1.5 |
| 141 | 50000 | 100000 | 1 | 5 | 1230584 | 2 | 4 | 13.71 | 90000 | 1230656 | 2 | 3 | 92.25 | 0 |
| 141 | 50000 | 150000 | 1 | 2.5 | 1166204 | 6 | 0 | 12.27 | 90000 | 1179194 | 6 | 0 | 157.48 | -1.1 |
| 141 | 50000 | 150000 | 1 | 5 | 1350343 | 4 | 2 | 21.39 | 90000 | 1335621 | 3 | 2 | 163.37 | 1.1 |
| 141 | 50000 | 200000 | 1 | 2.5 | 1175207 | 6 | 0 | 13.89 | 90000 | 1179194 | 6 | 0 | 176.35 | -0.3 |
| 141 | 50000 | 200000 | 1 | 5 | 1388878 | 6 | 0 | 24.87 | 90000 | 1410136 | 6 | 0 | 180.11 | -1.5 |
| 141 | 50000 | 300000 | 1 | 2.5 | 1176582 | 6 | 0 | 10.77 | 90000 | 1179194 | 6 | 0 | 295.02 | -0.2 |
| 141 | 50000 | 300000 | 1 | 5 | 1409642 | 7 | 0 | 24.48 | 90000 | 1410136 | 6 | 0 | 300.99 | 0 |
| 141 | 50000 | 500000 | 1 | 2.5 | 1180454 | 7 | 0 | 13.98 | 90000 | 1179194 | 6 | 0 | 313.18 | 0.1 |
| 141 | 50000 | 500000 | 1 | 5 | 1400033 | 6 | 0 | 21.54 | 90000 | 1410136 | 6 | 0 | 316.57 | -0.7 |

With *BOL* data higher number of hybrid systems are observed as a solution to different parameter combinations than the *LeonaG* experiment. Two of these solutions are shown in Figure 5.9. The reason is that the *BOL* data has a non-uniform spatial distribution pattern contrary to the *Leona* data, as can be seen in Figure 5.7.

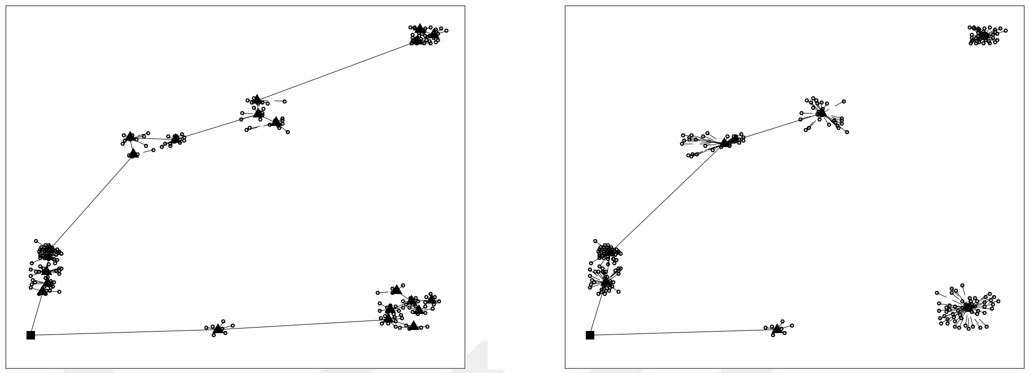


(a) Heuristic result of instance with $C_{CF} = 5k$, $C_{DF} = 20k$, $C_{LN} = 1$ and $C_{UN} = 2.5$

(b) Heuristic result of instance with $C_{CF} = 50k$, $C_{DF} = 100k$, $C_{LN} = 1$ and $C_{UN} = 2.5$

Figure 5.9: Solution Instances of *BOL* Experiment

On the other hand we mostly obtain decentralized solutions in the experiments conducted with the *Guinea* data due to its highly clustered spatial distribution pattern, as seen in Figure 5.7. Results of the *Guinea* experiment are presented in Table 5.6. Yet, we still observe centralized and hybrid systems as $C_{DF}/C_{CF}$ is increased, especially for high $C_{CF}/C_{LN}$ ratios. Two example solutions are provided in Figure 5.10.



(a) Heuristic result of instance with $C_{CF} = 5k$, $C_{DF} = 50k$, $C_{LN} = 1$ and $C_{UN} = 2.5$

(b) Heuristic result of instance with $C_{CF} = 50k$, $C_{DF} = 200k$, $C_{LN} = 1$ and $C_{UN} = 5$

Figure 5.10: Solution Instances of *Guinea* Experiment

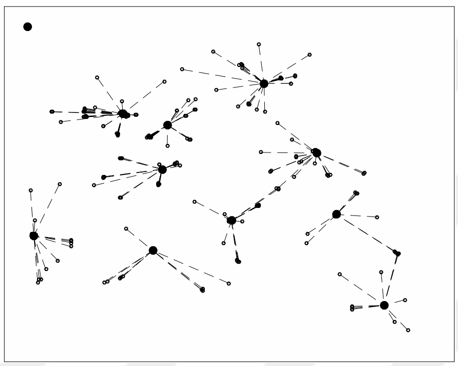
In the *Guinea* experiment, the solution times increase significantly compared to the *LeonaG* and*BOL* experiment. The average CPU time that the instances that belong to the first batch take is about 1600 seconds and the instances that belong to the second batch take is about 650 seconds. Heuristic algorithm finds 6.8% and 25.8% better solutions than the 2LDN on the average and at the best case, respectively. Yet, we noticed that the heuristic algorithm performed tremendous improvements mostly on the instances with decentralized solutions. Even so, the solutions obtained besides the decentralized ones are still, on the average, 6.6% better than 2LDN.

The reason that the heuristic algorithm provides such improvements over the solutions of 2LDN is the size of the data. Although, we experiment with a time limit of 90000 seconds, on average, 2LDN reports 30% optimality gap. Therefore, we conclude that as the size of the data set increases, the dominance of the heuristic algorithm upon 2LDN increases.

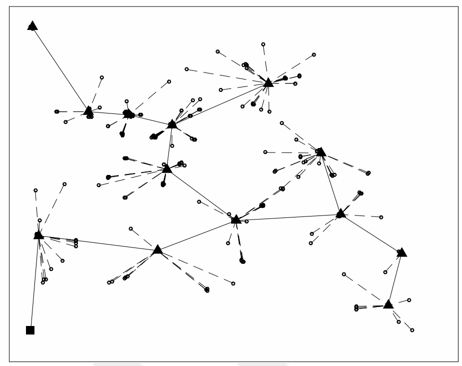
Table 5.6: Results of the Experiment with *Guinea*

| Scenario Setting | | | | | 2LDN | | | | | Heuristic Algorithm | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **n** | $C_{CF}$ | $C_{DF}$ | $C_{LN}$ | $C_{UN}$ | **Objective** | **# of CF** | **# of DF** | **Gap (%)** | **Time (sec.)** | **Objective** | **# of CF** | **# of DF** | **Time (sec.)** | **Comparison (%)** |
| 230 | 5000 | 10000 | 1 | 2.5 | 536805 | 0 | 25 | 4.52 | 90000 | 532009 | 0 | 21 | 819.41 | 0.9 |
| 230 | 5000 | 10000 | 1 | 5 | 532009 | 0 | 21 | 0 | 750.89 | 532009 | 0 | 21 | 819.88 | 0 |
| 230 | 5000 | 15000 | 1 | 2.5 | 651685 | 0 | 21 | 19.68 | 90000 | 628363 | 0 | 18 | 1152.52 | 3.6 |
| 230 | 5000 | 15000 | 1 | 5 | 661728 | 0 | 25 | 10.82 | 90000 | 628363 | 0 | 18 | 1111.71 | 5 |
| 230 | 5000 | 20000 | 1 | 2.5 | 776766 | 7 | 19 | 32.03 | 90000 | 705052 | 0 | 15 | 1643.85 | 9.2 |
| 230 | 5000 | 20000 | 1 | 5 | 787052 | 0 | 25 | 23.67 | 90000 | 705052 | 0 | 15 | 1360.84 | 10.4 |
| 230 | 5000 | 30000 | 1 | 2.5 | 950391 | 8 | 18 | 43.67 | 90000 | 818206 | 3 | 8 | 2424.94 | 13.9 |
| 230 | 5000 | 30000 | 1 | 5 | 1002195 | 6 | 17 | 39.41 | 90000 | 822941 | 0 | 11 | 1930.22 | 17.9 |
| 230 | 5000 | 50000 | 1 | 2.5 | 975195 | 30 | 0 | 46.37 | 90000 | 956837 | 22 | 0 | 2580.78 | 1.9 |
| 230 | 5000 | 50000 | 1 | 5 | 1386416 | 8 | 18 | 55.41 | 90000 | 1028271 | 0 | 9 | 2573.05 | 25.8 |
| 230 | 50000 | 100000 | 1 | 2.5 | 1399397.5 | 2 | 6 | 22.12 | 90000 | 1364139 | 2 | 5 | 446.39 | 2.5 |
| 230 | 50000 | 100000 | 1 | 5 | 1394829 | 0 | 7 | 18.11 | 90000 | 1394829 | 0 | 7 | 457.86 | 0 |
| 230 | 50000 | 150000 | 1 | 2.5 | 1497758.5 | 8 | 0 | 27.61 | 90000 | 1489759 | 8 | 0 | 468.9 | 0.5 |
| 230 | 50000 | 150000 | 1 | 5 | 1860089 | 4 | 4 | 38.3 | 90000 | 1683449 | 2 | 5 | 460.69 | 9.5 |
| 230 | 50000 | 200000 | 1 | 2.5 | 1501398 | 9 | 0 | 27.88 | 90000 | 1489759 | 8 | 0 | 670.52 | 0.8 |
| 230 | 50000 | 200000 | 1 | 5 | 1922790 | 7 | 1 | 42.39 | 90000 | 1916829 | 5 | 2 | 672.18 | 0.3 |
| 230 | 50000 | 300000 | 1 | 2.5 | 1490220 | 8 | 0 | 27.6 | 90000 | 1489759 | 8 | 0 | 832.46 | 0 |
| 230 | 50000 | 300000 | 1 | 5 | 1944851 | 8 | 0 | 42.29 | 90000 | 1945709 | 7 | 0 | 838.84 | 0 |
| 230 | 50000 | 500000 | 1 | 2.5 | 1489386.5 | 8 | 0 | 25.45 | 90000 | 1489759 | 8 | 0 | 992.77 | 0 |
| 230 | 50000 | 500000 | 1 | 5 | 1946765 | 8 | 0 | 43.24 | 90000 | 1945709 | 7 | 0 | 1032.5 | 0.1 |

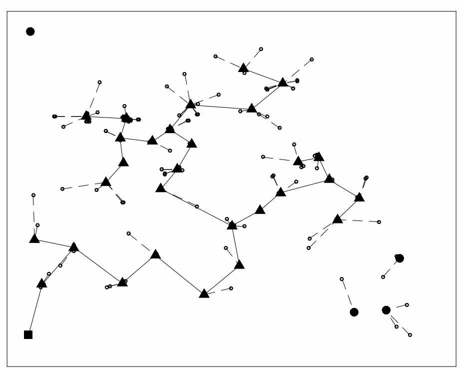
Similarly, our heuristic outperforms 2LDN in the experiments conducted with the *LGA* data which are reported in Table 5.7. In these experiments, in the 40% of instances, the heuristic algorithm finds a decentralized solution and 35% of the instances result with centralized networks. The solutions of the remaining 25% of the instances include hybrid systems. Example graphs for the decentralized, the centralized, and the hybrid systems are provided in Figure 5.11 (a-c), respectively.



(a) $C_{CF} = 50k$, $C_{DF} = 100k$
$C_{LN} = 1$, $C_{UN} = 5$



(b) $C_{CF} = 50k$, $C_{DF} = 150k$
$C_{LN} = 1$, $C_{UN} = 2.5$



(c) $C_{CF} = 5k$, $C_{DF} = 30k$, $C_{LN} = 1$ and
$C_{UN} = 2.5$

Figure 5.11: Solution Instances of *LGA* Experiment

Table 5.7: Results of the Experiment with $LGA$

| Scenario Setting | | | | | 2LDN | | | | | Heuristic Algorithm | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **n** | $C_{CF}$ | $C_{DF}$ | $C_{LN}$ | $C_{UN}$ | **Objective** | **# of CF** | **# of DF** | **Gap (%)** | **Time (sec.)** | **Objective** | **# of CF** | **# of DF** | **Time (sec.)** | **Comparison (%)** |
| 274 | 5000 | 10000 | 1 | 2.5 | 683766 | 0 | 40 | 2.46 | 90000 | 683572 | 0 | 40 | 1296.39 | 0.0 |
| 274 | 5000 | 10000 | 1 | 5 | 683572 | 0 | 40 | 0.01 | 34940 | 683572 | 0 | 40 | 1215.77 | 0.0 |
| 274 | 5000 | 15000 | 1 | 2.5 | 887718 | 0 | 37 | 16.95 | 90000 | 861782 | 0 | 32 | 2802.22 | 2.9 |
| 274 | 5000 | 15000 | 1 | 5 | 891991 | 0 | 36 | 7.8 | 90000 | 861782 | 0 | 32 | 1811.62 | 3.4 |
| 274 | 5000 | 20000 | 1 | 2.5 | 1067647 | 0 | 36 | 28.62 | 90000 | 1016068 | 0 | 29 | 8973.27 | 4.8 |
| 274 | 5000 | 20000 | 1 | 5 | 1069942 | 0 | 36 | 16.6 | 90000 | 1016068 | 0 | 29 | 2121.11 | 5.0 |
| 274 | 5000 | 30000 | 1 | 2.5 | 1321778 | 38 | 11 | 42.08 | 90000 | 1186840 | 29 | 4 | 5345.03 | 10.2 |
| 274 | 5000 | 30000 | 1 | 5 | 1377271 | 0 | 32 | 32.31 | 90000 | 1275602 | 0 | 23 | 7439.68 | 7.4 |
| 274 | 5000 | 50000 | 1 | 2.5 | 1244328 | 46 | 1 | 37.72 | 90000 | 1212205 | 32 | 1 | 5906.5 | 2.6 |
| 274 | 5000 | 50000 | 1 | 5 | 1949043 | 63 | 6 | 52.47 | 90000 | 1673562 | 0 | 16 | 6980.39 | 14.1 |
| 274 | 50000 | 100000 | 1 | 2.5 | 2261760 | 15 | 1 | 21.21 | 90000 | 2184179 | 12 | 1 | 869.72 | 3.4 |
| 274 | 50000 | 100000 | 1 | 5 | 2353691 | 0 | 13 | 19.9 | 90000 | 2293837 | 0 | 11 | 874.82 | 2.5 |
| 274 | 50000 | 150000 | 1 | 2.5 | 2270363 | 15 | 0 | 21.31 | 90000 | 2192306.5 | 13 | 0 | 1317.91 | 3.4 |
| 274 | 50000 | 150000 | 1 | 5 | 2997725 | 3 | 10 | 37.38 | 90000 | 2620287 | 9 | 1 | 1310.52 | 12.6 |
| 274 | 50000 | 200000 | 1 | 2.5 | 2259452 | 14 | 0 | 20.95 | 90000 | 2192306.5 | 13 | 0 | 1881.53 | 3.0 |
| 274 | 50000 | 200000 | 1 | 5 | 2798088 | 14 | 0 | 34.98 | 90000 | 2641757 | 10 | 0 | 1876.4 | 5.6 |
| 274 | 50000 | 300000 | 1 | 2.5 | 2263213 | 16 | 0 | 20.96 | 90000 | 2192306.5 | 13 | 0 | 2663.13 | 3.1 |
| 274 | 50000 | 300000 | 1 | 5 | 2877220 | 16 | 0 | 36.8 | 90000 | 2641757 | 10 | 0 | 2669.18 | 8.2 |
| 274 | 50000 | 500000 | 1 | 2.5 | 2298378 | 15 | 0 | 23.46 | 90000 | 2192306.5 | 13 | 0 | 2890.27 | 4.6 |
| 274 | 50000 | 500000 | 1 | 5 | 2925740 | 17 | 0 | 37.44 | 90000 | 2641757 | 10 | 0 | 3017.98 | 9.7 |

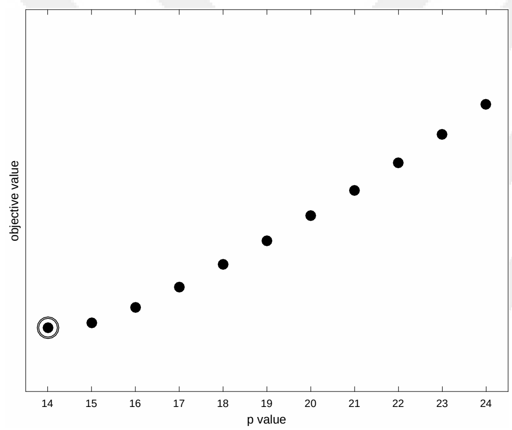
We conclude that once the size of the instance increases, the heuristic algorithm provides better solutions than the exact formulation within a reasonable time. However, we recognize that all of the instances that our heuristic outperforms 2LDN significantly, has a decentralized solution. Thus, we also decided to compare our heuristic with the SPLP. When all facilities are forced to be decentralized ones, the problem becomes the SPLP. Therefore the solution objective of the SPLP is an upper bound to our objective. We conduct experiments with *LeonaG*, *BOL*, *Guinea* and *LGA* instances and the results are provided in Table C.1.

As expected, for the instances with decentralized solutions, SPLP is observed to be a better method than our heuristic algorithm. However, for the instances that our heuristic obtains a centralized or a hybrid system, which form the 62.5% of the instances experimented with, the results found by our heuristic algorithm are better than the SPLP. For those instances, the heuristic algorithm finds a solution, on average, 32% better than 2LDN for the *LeonaG* data, 30% better for the *BOL* data, 32% better for the *Guinea* data, and 40% better for the *LGA* data. Hence, we conclude that the proposed heuristic algorithm is valid for designing two level networks with a tree-star configuration.

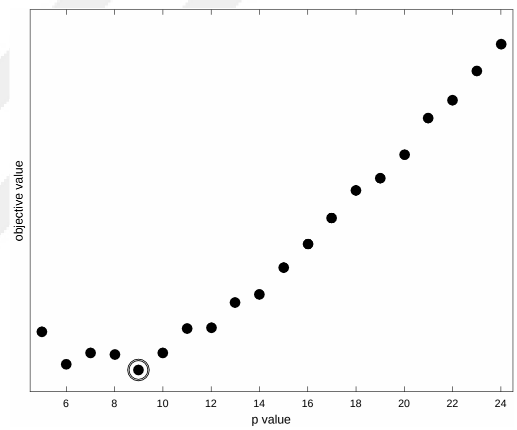
## 5.3 Iteration Performance of the Heuristic Algorithm

In this section, we analyze the iterations performed within the heuristic algorithm. We observed several common patterns in terms of the change in the objective values found at iterations. We analyze all instances except for the second batches of *LeonaG*, *BOL* and *Guinea* because there are not enough number of iterations to deduce a pattern, as can be seen in Table B.1.

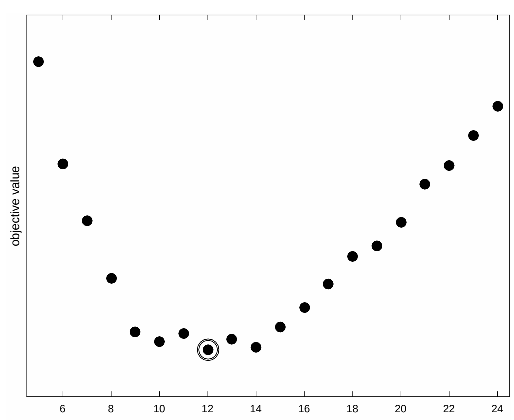
We observe three different patterns in the objective function for the different p values as shown in Figure 5.12. Each dot in the figure, represents the objective value found for corresponding $p$ value at each iteration. The minimum objective value achieved during the iterations, which is reported as the solution for the instance, is marked with larger circles. The first pattern is an increasing convex curve as in Figure 5.12a. The second one is increasing curve with fluctuations which violates convexity, as in Figure 5.12b and the third one is as in Figure 5.12c. We denote these patterns as P1, P2 and P3. After all, 70% of the instances of the *LeonaG* experiment have a P1 pattern, 20% have a P2 pattern and 10% have a P3 pattern.




(a) $C_{CF} = 5k$, $C_{DF} = 10k$
$C_{LN} = 1$, $C_{UN} = 2.5$




(b) $C_{CF} = 50k$, $C_{DF} = 500k$
$C_{LN} = 1$, $C_{UN} = 5$

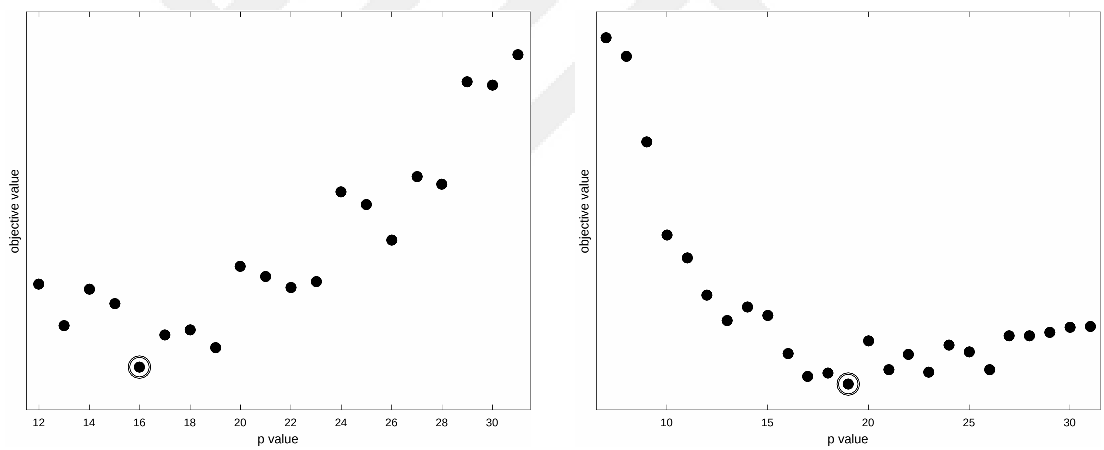



(c) $C_{CF} = 50k$, $C_{DF} = 500k$
$C_{LN} = 1$, $C_{UN} = 2.5$

Figure 5.12: Iteration Performances of some *LeonaG* instances

To continue with $BOL$ experiment we observed the same patterns stated above, as well as two additional ones. The first one is an increasing pattern with great fluctuation as in Figure 5.13a, and the second one is a decreasing curve with fluctuations as in Figure 5.13b. We denote these patterns as P4 and P5. As a result, 50% of the instances of the $BOL$ experiment have a P1 pattern, 20% have a P2 pattern, 20% have a P4 pattern and 10% have a P5 pattern.

Furthermore, we observe that 60% of the instances of the *Guinea* experiment have the P1 pattern and 30% have the P2 pattern. Lastly, 45% of the instances of the $LGA$ experiment have a P1 pattern, 25% have the P5 pattern, 15% have the P3 pattern and 10% have the P2 pattern.




(a) $C_{CF} = 5k$, $C_{DF} = 30k$
$C_{LN} = 1$, $C_{UN} = 2.5$

(b) $C_{CF} = 50k$, $C_{DF} = 500k$
$C_{LN} = 1$, $C_{UN} = 2.5$

Figure 5.13: Iteration Performances of some $BOL$ instances

In a nutshell, 54% of the instances we experiment has P1 pattern, which means that the iterations performed after the first iteration are redundant. Moreover, those instances are the ones which have decentralized solutions and solutions can be obtained by SPLP in a shorter time. In addition, 20% of the instances we experiment have P2 pattern where after a small increase in p, the iterations becomes redundant.

There is a possibility of reducing the computational time that heuristic algorithm requires, by implementing a neighborhood search to the heuristic algorithm to decrease the number of iterations it performs without compromising the best value it can achieve.

An additional way of decreasing the computational time could be implementing a preprocessing step before deciding the type of the selected facilities and the upper-level connections. After deciding the locations of the facilities in Step 2, it is possible to identify the types of the selected facilities according to following rule:

- If the distance between a facility and the source node is smaller than $(C_{DF} - C_{CF})/C_{UN}$, then that facility has to be a centralized facility.

Then, those identified facilities would be given as an input for the PCST problem. Since the number of decisions made in Step 3 is decreased, the computational time that Step 3 requires as well as the total solution time of the heuristic algorithm would be decreased.

# Chapter 6

# Conclusion

In this thesis, we study the problem of designing a two level hybrid network with a tree-star topology. First we provide single-commodity flow based and multi-commodity based formulations. As the problem is NP-hard, we also propose an iterative heuristic algorithm that solves numerous subproblems. The heuristic algorithm sequentially designs the first and the second level of the network for alternative number of facilities and selects the best objective value.

We conduct experiments with both synthetic and real data that we obtained from an open source tool called Network Planner to compare the performance of the heuristic algorithm with the exact formulation we selected for the problem.

In the experiments performed with synthetic data, we have compared heuristic algorithm's solutions with the optimal solutions obtained with using the exact formulation. While generating the synthetic data our aim was simulating different cases in terms of spatial distribution patterns. In consequence to these experiments, we observed that heuristic algorithm finds a solution, at most, 5.57% worse than the exact formulation and is able to design centralized, decentralized and hybrid systems at presence of different patterns. In addition, we also provide the modified and the extended version of the heuristic approach to increase its efficiency of detecting different patterns. In 53% of the instances a hybrid network is obtained as an optimal solution. For those instances heuristic approaches find solutions, on average, 0.89% close to the optimal. The remaining 25% and 22% of the instances have centralized and decentralized networks as an optimal design. The heuristic approaches finds the optimal solution for all of the decentralized networks and finds a solution, on average, 2.68% close to the optimal solution for the centralized networks.

Afterwards, in the experiments conducted with real data we used wide a range of parameter combinations. The reason we used a wide range of parameters was to examine the behavior of the problem under different cost parameters as well as the response of the heuristic algorithm to those changing environments and provided a sensitivity analysis of the parameters. The real instances were too complex to obtain an optimal solution with the exact formulation unlike the synthetic data. As a result, heuristic algorithm finds a solution, at most, 2.28% worse than the exact formulation while it finds a better solutions up to 25.8%. In addition, as the size of the dataset increases, the percentage of the improvements provided by the heuristic algorithm increases as well.

In real life applications, there are limitations on the maximum distance of the connections established between the demand nodes and the facilities due to feasibility and reliability concerns. Similarly, facilities can have capacity constraints regarding the total number of customers that they can serve. Thus, as a further research direction, upper bounds can be defined for the number of demand nodes that a facility can serve and the maximum distance that a lower-level connection can be established.

# Bibliography

[1] C. S. Revelle, H. A. Eiselt, and M. S. Daskin, "A bibliography for some fundamental problem categories in discrete location science," *European Journal of Operational Research*, vol. 184, no. 3, pp. 817–848, 2008.

[2] S. Bastani and N. Kazemzadeh, "Hierarchical location problem," in *Facility Location*, pp. 219–241, Springer, 2009.

[3] I. Contreras and E. Fernández, "General network design: A unified view of combined location and network design problems," *European Journal of Operational Research*, vol. 219, no. 3, pp. 680–697, 2012.

[4] C. Ortiz Astorquiza, *Multi-level facility location problems*. PhD thesis, Concordia University, 2017.

[5] P. Hines, S. Blumsack, and M. Schläpfer, "When are decentralized infrastructure networks preferable to centralized ones?," 2017.

[6] L. Jia and L. Tong, "Renewables and storage in distribution systems: Centralized vs. decentralized integration," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 665–674, 2016.

[7] A. Kargarian, J. Mohammadi, J. Guo, S. Chakrabarti, M. Barati, G. Hug, S. Kar, and R. Baldick, "Toward distributed/decentralized dc optimal power flow implementation in future electric power systems," *IEEE Transactions on Smart Grid*, 2016.

[8] N. Turken, J. Carrillo, and V. Verter, “Facility location and capacity acquisition under carbon tax and emissions limits: To centralize or to decentralize?,” *International Journal of Production Economics*, vol. 187, pp. 126–141, 2017.

[9] D. Mourtzis, M. Doukas, and F. Psarommatis, “A multi-criteria evaluation of centralized and decentralized production networks in a highly customer-driven environment,” *CIRP Annals-Manufacturing Technology*, vol. 61, no. 1, pp. 427–430, 2012.

[10] M. Shahraeini, M. H. Javidi, and M. S. Ghazizadeh, “Comparison between communication infrastructures of centralized and decentralized wide area measurement systems,” *IEEE Transactions on Smart Grid*, vol. 2, no. 1, pp. 206–211, 2011.

[11] D. Manolis, T. Pappa, C. Diakaki, I. Papamichail, and M. Papageorgiou, “Simulation study of centralized vs decentralized approaches to the signal control of large-scale urban networks,” tech. rep., 2017.

[12] A. Balakrishnan, T. L. Magnanti, and P. Mirchandani, “A dual-based algorithm for multi-level network design,” *Management Science*, vol. 40, no. 5, pp. 567–581, 1994.

[13] J. G. Klincewicz, “Hub location in backbone/tributary network design: A review,” *Location Science*, vol. 6, no. 1, pp. 307–335, 1998.

[14] S. Chamberland, B. Sansò, and O. Marcotte, “Topological design of two-level telecommunication networks with modular switches,” *Operations Research*, vol. 48, no. 5, pp. 745–760, 2000.

[15] S. M. Faccin, C. Wijting, J. Kenckt, and A. Damle, “Mesh wlan networks: concept and system design,” *IEEE Wireless Communications*, vol. 13, no. 2, pp. 10–17, 2006.

[16] G. Park, Y. Lee, and J. Han, “A two-level location–allocation problem in designing local access fiber optic networks,” *Computers & Operations Research*, vol. 51, pp. 52–63, 2014.

[17] B. Yazar, O. Arslan, O. E. Karaşan, and B. Y. Kara, "Fiber optical network design problems: A case for turkey," *Omega*, vol. 63, pp. 23–40, 2016.

[18] J. F. Bigotte, D. Krass, A. P. Antunes, and O. Berman, "Integrated modeling of urban hierarchy and transportation network planning," *Transportation Research Part A: Policy and Practice*, vol. 44, no. 7, pp. 506–522, 2010.

[19] A. M. Costa, P. M. França, and C. Lyra Filho, "Two-level network design with intermediate facilities: An application to electrical distribution systems," *Omega*, vol. 39, no. 1, pp. 3–13, 2011.

[20] A. S. Kocaman, W. T. Huh, and V. Modi, "Initial layout of power distribution systems for rural electrification: A heuristic algorithm for multilevel network design," *Applied Energy*, vol. 96, pp. 302–315, 2012.

[21] M. T. Melo, S. Nickel, and F. Saldanha-Da-Gama, "Facility location and supply chain management–a review," *European Journal of Operational Research*, vol. 196, no. 2, pp. 401–412, 2009.

[22] S.-h. Chung, Y.-s. Myung, and D.-w. Tcha, "Optimal design of a distributed network with a two-level hierarchical structure," *European Journal of Operational Research*, vol. 62, no. 1, pp. 105–115, 1992.

[23] S. J. Ratick, J. P. Osleeb, and D. Hozumi, "Application and extension of the moore and revelle hierarchical maximal covering model," *Socio-Economic Planning Sciences*, vol. 43, no. 2, pp. 92–101, 2009.

[24] H. K. Smith, P. R. Harper, C. N. Potts, and A. Thyle, "Planning sustainable community health schemes in rural areas of developing countries," *European Journal of Operational Research*, vol. 193, no. 3, pp. 768–777, 2009.

[25] A. I. Barros, R. Dekker, and V. Scholten, "A two-level network for recycling sand: a case study," *European Journal of Operational Research*, vol. 110, no. 2, pp. 199–214, 1998.

[26] G. Şahin and H. Süral, "A review of hierarchical facility location models," *Computers & Operations Research*, vol. 34, no. 8, pp. 2310–2331, 2007.

[27] R. Z. Farahani, M. Hekmatfar, A. B. Arabani, and E. Nikbakhsh, "Hub location problems: A review of models, classification, solution techniques, and applications," *Computers & Industrial Engineering*, vol. 64, no. 4, pp. 1096–1109, 2013.

[28] C.-H. Lee, H.-B. Ro, and D.-W. Tcha, "Topological design of a two-level network with ring-star configuration," *Computers & Operations Research*, vol. 20, no. 6, pp. 625–637, 1993.

[29] J.-G. Kim and D.-W. Tcha, "Optimal design of a two-level hierarchical network with tree-star configuration," *Computers & Industrial Engineering*, vol. 22, no. 3, pp. 273–281, 1992.

[30] B. Gendron, P.-V. Khuong, and F. Semet, "Comparison of formulations for the two-level uncapacitated facility location problem with single assignment constraints," *Computers & Operations Research*, vol. 86, pp. 86–93, 2017.

[31] C. Ortiz-Astorquiza, I. Contreras, and G. Laporte, "Formulations and approximation algorithms for multi-level uncapacitated facility location," *INFORMS Journal on Computing*, 2017.

[32] S. L. Hakimi, "Optimum distribution of switching centers in a communication network and some related graph theoretic problems," *Operations Research*, vol. 13, no. 3, pp. 462–475, 1965.

[33] C. S. ReVelle and R. W. Swain, "Central facilities location," *Geographical analysis*, vol. 2, no. 1, pp. 30–42, 1970.

[34] O. Kariv and S. L. Hakimi, "An algorithmic approach to network location problems. ii: The p-medians," *SIAM Journal on Applied Mathematics*, vol. 37, no. 3, pp. 539–560, 1979.

[35] K. E. Rosing, C. ReVelle, and H. Rosing-Vogelaar, "The p-median and its linear programming relaxation: An approach to large problems," *Journal of the Operational Research Society*, pp. 815–823, 1979.

[36] R. L. Church, "Cobra: a new formulation of the classic p-median location problem," *Annals of Operations Research*, vol. 122, no. 1-4, pp. 103–120, 2003.

[37] G. Cornuejols, G. L. Nemhauser, and L. A. Wolsey, "A canonical representation of simple plant location problems and its applications," *SIAM Journal on Algebraic Discrete Methods*, vol. 1, no. 3, pp. 261–272, 1980.

[38] S. Elloumi, "A tighter formulation of the p-median problem," *Journal of combinatorial optimization*, vol. 19, no. 1, pp. 69–83, 2010.

[39] A. A. Kuehn and M. J. Hamburger, "A heuristic program for locating warehouses," *Management Science*, vol. 9, no. 4, pp. 643–666, 1963.

[40] M. L. Balinski, "On finding integer solutions to linear programs," tech. rep., Mathematica Princeton NJ, 1964.

[41] A. S. Manne, "Plant location under economies-of-scaledecentralization and computation," *Management Science*, vol. 11, no. 2, pp. 213–235, 1964.

[42] J. F. Stollsteimer, "A working model for plant numbers and locations," *Journal of Farm Economics*, vol. 45, no. 3, pp. 631–645, 1963.

[43] S. Chopra and M. R. Rao, "The steiner tree problem i: Formulations, compositions and extension of facets," *Mathematical Programming*, vol. 64, no. 1, pp. 209–229, 1994.

[44] D. S. Johnson, "The np-completeness column: An ongoing gulde," *Journal of Algorithms*, vol. 3, no. 4, pp. 381–395, 1982.

[45] M. Minkoff, *The prize collecting Steiner tree problem.* PhD thesis, Massachusetts Institute of Technology, 2000.

[46] D. Bienstock, M. X. Goemans, D. Simchi-Levi, and D. Williamson, "A note on the prize collecting traveling salesman problem," *Mathematical Programming*, vol. 59, no. 1, pp. 413–420, 1993.

[47] M. X. Goemans and D. P. Williamson, "A general approximation technique for constrained forest problems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 296–317, 1995.

[48] M. X. Goemans and D. P. Williamson, "The primal-dual method for approximation algorithms and its application to network design problems," *Approximation algorithms for NP-hard problems*, pp. 144–191, 1997.

[49] E. W. Weisstein, "Great circle." http://mathworld.wolfram.com/GreatCircle.html. Accessed April, 2017.

[50] G. Cornuéjols, G. L. Nemhauser, and L. A. Wolsey, "The uncapacitated facility location problem," tech. rep., Carnegie-mellon univ pittsburgh pa management sciences research group, 1983.

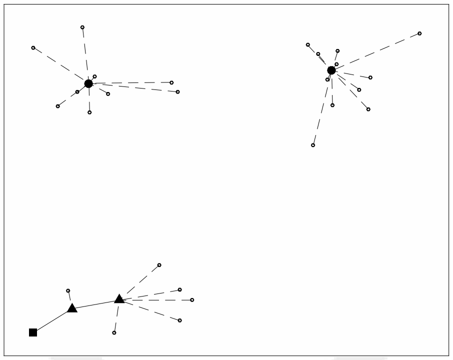
[51] D. S. Johnson, M. Minkoff, and S. Phillips, "The prize collecting steiner tree problem: theory and practice," in *SODA*, vol. 1, p. 4, 2000.

[52] "Network planner." http://optimus.modilabs.org/. Accessed May, 2017.

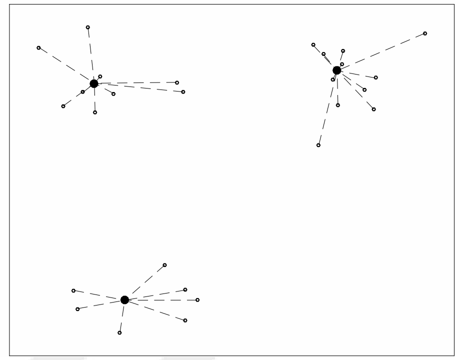
[53] "Utm - universal transverse mercator." http://geokov.com/education/utm.aspx. Accessed April, 2017.

# Appendix A

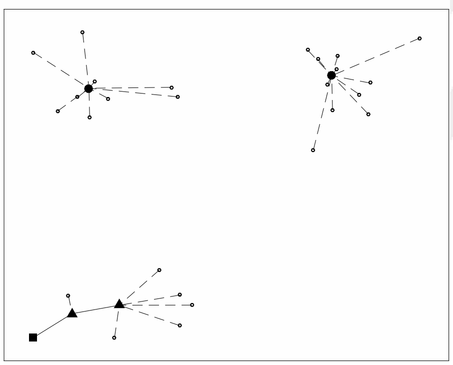
# RANDOM DATA RESULTS



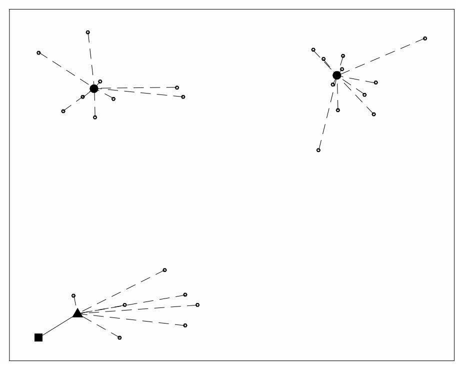
(a) $C_{CF} = 2k$, $C_{DF} = 10k$
$C_{LN} = 1$, $C_{UN} = 2$



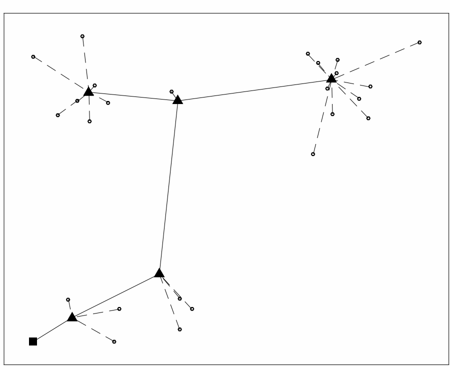
(b) $C_{CF} = 2k$, $C_{DF} = 10k$
$C_{LN} = 1$, $C_{UN} = 4$



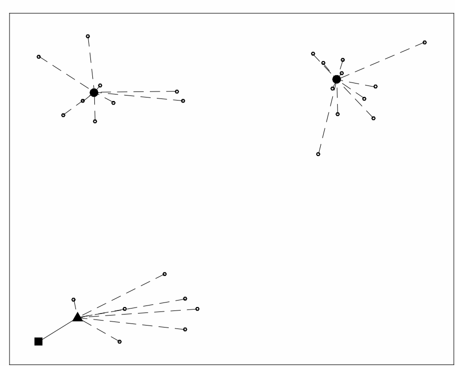
(c) $C_{CF} = 2k$, $C_{DF} = 15k$
$C_{LN} = 1$, $C_{UN} = 2$



(d) $C_{CF} = 2k$, $C_{DF} = 15k$
$C_{LN} = 1$, $C_{UN} = 4$

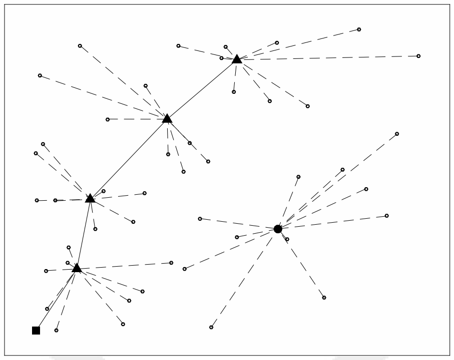


(e) $C_{CF} = 2k$, $C_{DF} = 20k$
$C_{LN} = 1$, $C_{UN} = 2$

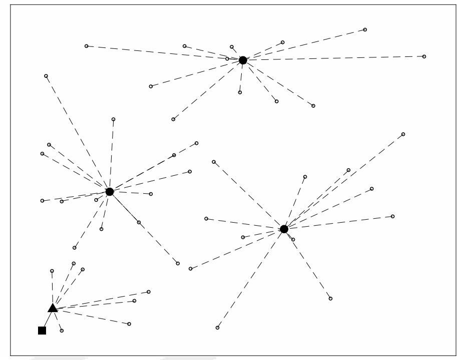


(f) $C_{CF} = 2k$, $C_{DF} = 20k$
$C_{LN} = 1$, $C_{UN} = 4$

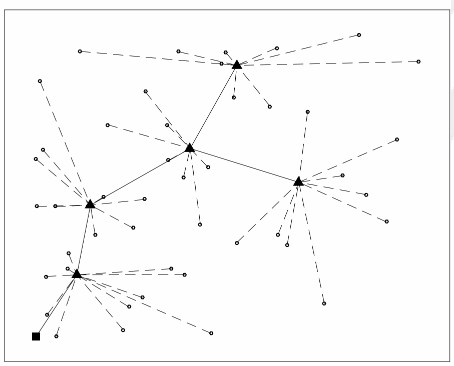
Figure A.1: Results of the Type II (Clustered) Random Data with $n = 30$



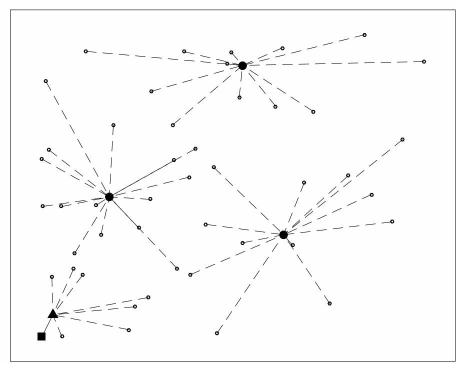
(a) $C_{CF} = 5k$, $C_{DF} = 15k$
$C_{LN} = 1$, $C_{UN} = 2.5$



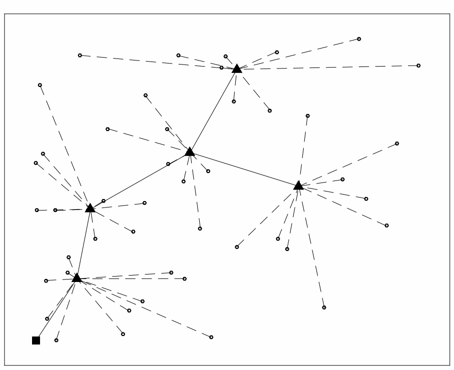
(b) $C_{CF} = 5k$, $C_{DF} = 15k$
$C_{LN} = 1$, $C_{UN} = 5$



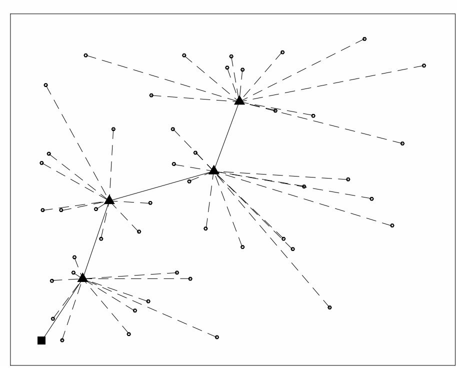
(c) $C_{CF} = 5k$, $C_{DF} = 20k$
$C_{LN} = 1$, $C_{UN} = 2.5$



(d) $C_{CF} = 5k$, $C_{DF} = 20k$
$C_{LN} = 1$, $C_{UN} = 5$

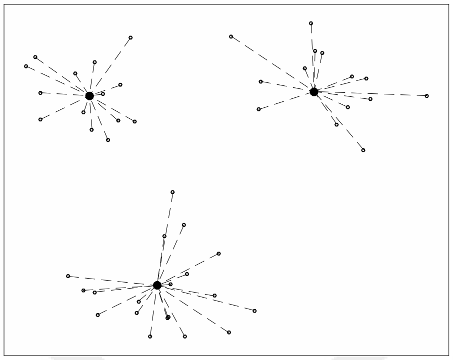


(e) $C_{CF} = 5k$, $C_{DF} = 30k$
$C_{LN} = 1$, $C_{UN} = 2.5$

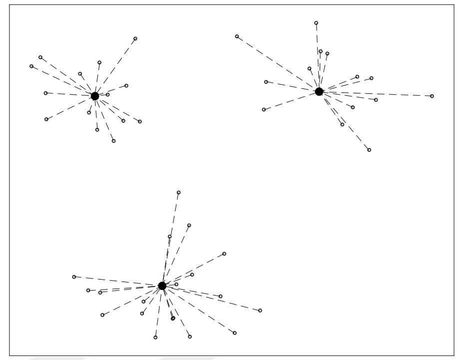


(f) $C_{CF} = 5k$, $C_{DF} = 30k$
$C_{LN} = 1$, $C_{UN} = 5$

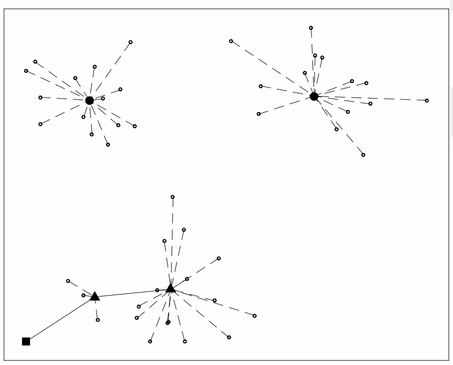
Figure A.2: Results of the Type I (Uniform) Random Data with $n = 50$



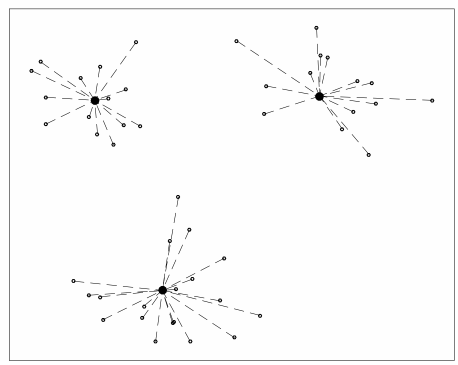
(a) $C_{CF} = 5k$, $C_{DF} = 15k$
$C_{LN} = 1$, $C_{UN} = 2.5$



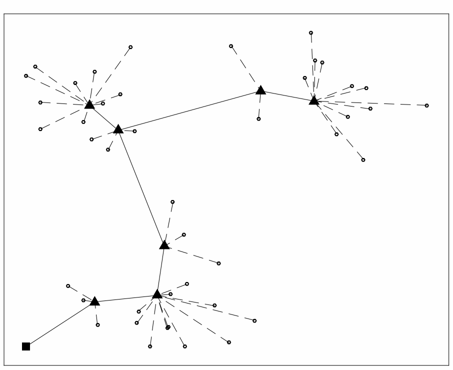
(b) $C_{CF} = 5k$, $C_{DF} = 15k$
$C_{LN} = 1$, $C_{UN} = 5$



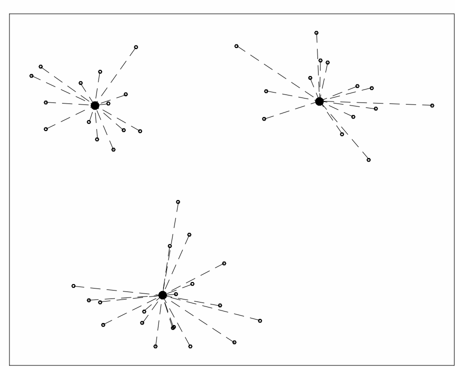
(c) $C_{CF} = 5k$, $C_{DF} = 20k$
$C_{LN} = 1$, $C_{UN} = 2.5$



(d) $C_{CF} = 5k$, $C_{DF} = 20k$
$C_{LN} = 1$, $C_{UN} = 5$

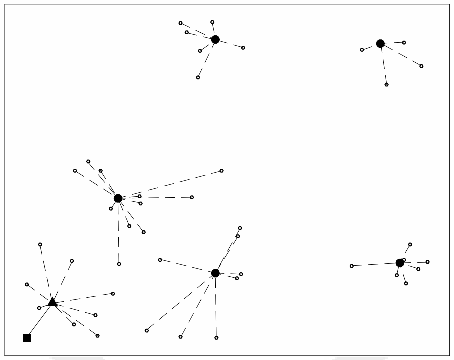


(e) $C_{CF} = 5k$, $C_{DF} = 30k$
$C_{LN} = 1$, $C_{UN} = 2.5$

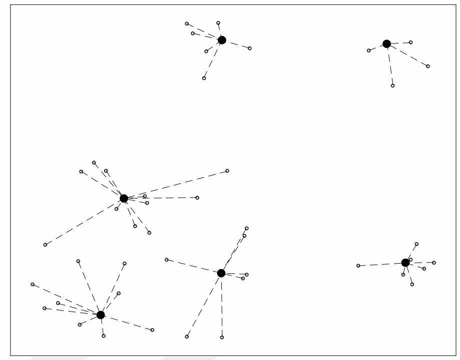


(f) $C_{CF} = 5k$, $C_{DF} = 30k$
$C_{LN} = 1$, $C_{UN} = 5$

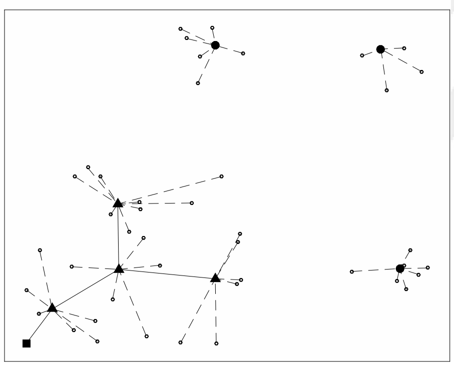
Figure A.3: Results of the Type II (Clustered) Random Data with $n = 50$



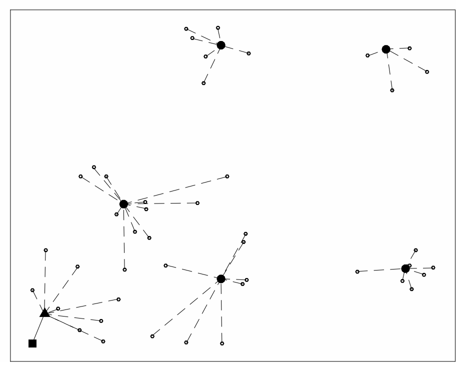
(a) $C_{CF} = 5k$, $C_{DF} = 15k$
$C_{LN} = 1$, $C_{UN} = 2.5$



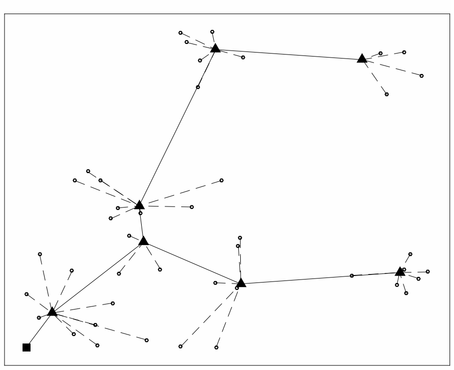
(b) $C_{CF} = 5k$, $C_{DF} = 15k$
$C_{LN} = 1$, $C_{UN} = 5$



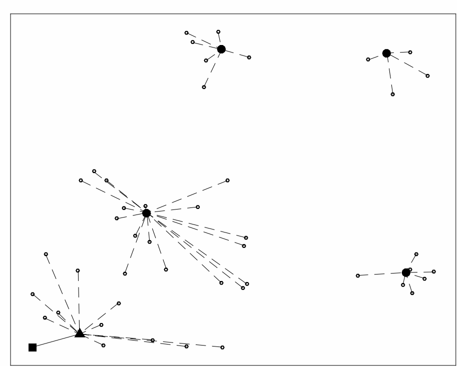
(c) $C_{CF} = 5k$, $C_{DF} = 20k$
$C_{LN} = 1$, $C_{UN} = 2.5$



(d) $C_{CF} = 5k$, $C_{DF} = 20k$
$C_{LN} = 1$, $C_{UN} = 5$



(e) $C_{CF} = 5k$, $C_{DF} = 30k$
$C_{LN} = 1$, $C_{UN} = 2.5$



(f) $C_{CF} = 5k$, $C_{DF} = 30k$
$C_{LN} = 1$, $C_{UN} = 5$

Figure A.4: Results of the Type III (Mixed) Random Data with $n = 50$

# Appendix B

# ITERATION AND CPU TIME BREAKDOWN OF THE HEURISTIC ALGORITHM

Table B.1: Iteration numbers, *pmin* and *pmax* values of the experiments conducted with *Leona, BOL, Guinea* and *LGA* reported in Tables 5.4, 5.5, 5.6 and 5.7

| *LeonaG* | | | *BOL* | | | *Guinea* | | | *LGA* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ***pmax*** | ***pmin*** | **# of iterations** | ***pmax*** | ***pmin*** | **# of iterations** | ***pmax*** | ***pmin*** | **# of iterations** | ***pmax*** | ***pmin*** | **# of iterations** |
| 25 | 14 | 11 | 32 | 21 | 11 | 30 | 21 | 9 | 59 | 40 | 19 |
| 25 | 14 | 11 | 32 | 21 | 11 | 30 | 21 | 9 | 59 | 40 | 19 |
| 25 | 10 | 15 | 32 | 16 | 16 | 30 | 18 | 12 | 59 | 32 | 27 |
| 25 | 10 | 15 | 32 | 16 | 16 | 30 | 18 | 12 | 59 | 32 | 27 |
| 25 | 8 | 17 | 32 | 15 | 17 | 30 | 15 | 15 | 59 | 29 | 30 |
| 25 | 8 | 17 | 32 | 15 | 17 | 30 | 15 | 15 | 59 | 29 | 30 |
| 25 | 7 | 18 | 32 | 12 | 20 | 30 | 11 | 19 | 59 | 23 | 36 |
| 25 | 7 | 18 | 32 | 12 | 20 | 30 | 11 | 19 | 59 | 23 | 36 |
| 25 | 5 | 20 | 32 | 7 | 25 | 30 | 9 | 21 | 59 | 16 | 43 |
| 25 | 5 | 20 | 32 | 7 | 25 | 30 | 9 | 21 | 59 | 16 | 43 |
| 5 | 3 | 2 | 7 | 5 | 2 | 9 | 7 | 2 | 16 | 11 | 5 |
| 5 | 3 | 2 | 7 | 5 | 2 | 9 | 7 | 2 | 16 | 11 | 5 |
| 5 | 2 | 3 | 7 | 4 | 3 | 9 | 7 | 2 | 16 | 9 | 7 |
| 5 | 2 | 3 | 7 | 4 | 3 | 9 | 7 | 2 | 16 | 9 | 7 |
| 5 | 2 | 3 | 7 | 4 | 3 | 9 | 6 | 3 | 16 | 7 | 9 |
| 5 | 2 | 3 | 7 | 4 | 3 | 9 | 6 | 3 | 16 | 7 | 9 |
| 5 | 1 | 4 | 7 | 2 | 5 | 9 | 5 | 4 | 16 | 5 | 11 |
| 5 | 1 | 4 | 7 | 2 | 5 | 9 | 5 | 4 | 16 | 5 | 11 |
| 5 | 1 | 4 | 7 | 2 | 5 | 9 | 4 | 5 | 16 | 4 | 12 |
| 5 | 1 | 4 | 7 | 2 | 5 | 9 | 4 | 5 | 16 | 4 | 12 |

Table B.2: CPU Time Breakdown of the Experiments conducted with *Leona*, *BOL*, *Guinea* and *LGA* reported in Tables 5.4, 5.5, 5.6 and 5.7

| *LeonaG* | | | *Bol* | | | *Guinea* | | | *LGA* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Total Algoritm Time (sec)** | **Total Optimizer Time (sec)** | **Ratio** | **Total Algoritm Time (sec)** | **Total Optimizer Time (sec)** | **Ratio** | **Total Algoritm Time (sec)** | **Total Optimizer Time (sec)** | **Ratio** | **Total Algoritm Time (sec)** | **Total Optimizer Time (sec)** | **Ratio** |
| 73.41 | 25.19 | 34% | 165.86 | 45.17 | 27% | 819.41 | 183.88 | 22% | 1296.39 | 408.83 | 32% |
| 72.07 | 22.73 | 32% | 158.97 | 35.45 | 22% | 819.88 | 194.09 | 24% | 1215.77 | 335.69 | 28% |
| 260.15 | 192.06 | 74% | 491.23 | 321.61 | 65% | 1152.52 | 290.9 | 25% | 2802.22 | 1547.28 | 55% |
| 92.11 | 30.69 | 33% | 223.9 | 60.07 | 27% | 1111.71 | 240.77 | 22% | 1811.62 | 551.72 | 30% |
| 263.49 | 188.46 | 72% | 438.56 | 264.69 | 60% | 1643.85 | 537.99 | 33% | 8973.27 | 7460.06 | 83% |
| 103.76 | 34.64 | 33% | 259.58 | 83.13 | 32% | 1360.84 | 311.45 | 23% | 2121.11 | 610.94 | 29% |
| 259.76 | 180.38 | 69% | 511.83 | 293.16 | 57% | 2424.94 | 998.6 | 41% | 5345.03 | 3378.91 | 63% |
| 343.65 | 260.3 | 76% | 539.02 | 316.8 | 59% | 1930.22 | 562.9 | 29% | 7439.68 | 5442.27 | 73% |
| 274.43 | 191.97 | 70% | 696.16 | 372.86 | 54% | 2580.78 | 973.74 | 38% | 5906.5 | 3258.28 | 55% |
| 272.08 | 187.7 | 69% | 618.73 | 303.84 | 49% | 2573.05 | 997.22 | 39% | 6980.39 | 4318.6 | 62% |
| 21.97 | 4.89 | 22% | 97.16 | 15.35 | 16% | 446.39 | 79.13 | 18% | 869.72 | 183.72 | 21% |
| 22.44 | 5.28 | 24% | 92.25 | 14.06 | 15% | 457.86 | 85.47 | 19% | 874.82 | 178.42 | 20% |
| 26.31 | 6.61 | 25% | 157.48 | 22.37 | 14% | 468.9 | 88.32 | 19% | 1317.91 | 258.73 | 20% |
| 26.89 | 6.3 | 23% | 163.37 | 28.68 | 18% | 460.69 | 86.42 | 19% | 1310.52 | 253.45 | 19% |
| 27.22 | 7.16 | 26% | 176.35 | 25.93 | 15% | 670.52 | 121.14 | 18% | 1881.53 | 354.2 | 19% |
| 26.74 | 6.86 | 26% | 180.11 | 25.74 | 14% | 672.18 | 118.98 | 18% | 1876.4 | 355.43 | 19% |
| 44.45 | 21.71 | 49% | 295.02 | 41.17 | 14% | 832.46 | 153.85 | 18% | 2663.13 | 479.81 | 18% |
| 45.41 | 22.2 | 49% | 300.99 | 42.23 | 14% | 838.84 | 155.14 | 18% | 2669.18 | 476.57 | 18% |
| 45.12 | 22.49 | 50% | 313.18 | 42.91 | 14% | 992.77 | 176.11 | 18% | 2890.27 | 515.58 | 18% |
| 46 | 22.95 | 50% | 316.57 | 44.45 | 14% | 1032.5 | 182.98 | 18% | 3017.98 | 520.13 | 17% |

# Appendix C

# HEURISTIC ALGORITHM vs SPLP COMPARISON

Table C.1: Comparison of the Heuristic Algorithm with SPLP for *LeonaG*

| Parameters | | | | Heuristic Algorithm | | | | SPLP | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $C_{CF}$ | $C_{DF}$ | $C_{LN}$ | $C_{UN}$ | **Objective** | **# of CF** | **# of DF** | **Time (sec.)** | **Objective** | **# of DF** | **Time (sec.)** | **Comparison (%)** |
| 5000 | 10000 | 1 | 2.5 | 318231 | 0 | 14 | 73.41 | 318231 | 14 | 0.63 | 0 |
| 5000 | 10000 | 1 | 5 | 318231 | 0 | 14 | 72.07 | 318231 | 14 | 0.55 | 0 |
| 5000 | 15000 | 1 | 2.5 | 378347 | 0 | 10 | 260.15 | 378347 | 10 | 0.59 | 0 |
| 5000 | 15000 | 1 | 5 | 378347 | 0 | 10 | 92.11 | 378347 | 10 | 0.61 | 0 |
| 5000 | 20000 | 1 | 2.5 | 419803 | 9 | 3 | 263.49 | 424207 | 8 | 0.62 | -1 |
| 5000 | 20000 | 1 | 5 | 424207 | 0 | 8 | 103.76 | 424207 | 8 | 0.63 | 0 |
| 5000 | 30000 | 1 | 2.5 | 427161 | 12 | 0 | 259.76 | 503332 | 7 | 0.67 | -17.8 |
| 5000 | 30000 | 1 | 5 | 501782 | 1 | 6 | 343.65 | 503332 | 7 | 0.6 | -0.3 |
| 5000 | 50000 | 1 | 2.5 | 427161 | 12 | 0 | 274.43 | 613255 | 5 | 0.67 | -43.6 |
| 5000 | 50000 | 1 | 5 | 571367 | 9 | 0 | 272.08 | 613255 | 5 | 0.68 | -7.3 |
| 50000 | 100000 | 1 | 2.5 | 703614 | 3 | 0 | 21.97 | 772841 | 3 | 0.68 | -9.8 |
| 50000 | 100000 | 1 | 5 | 772841 | 0 | 3 | 22.44 | 772841 | 3 | 0.66 | 0 |
| 50000 | 150000 | 1 | 2.5 | 703614 | 3 | 0 | 26.31 | 894591 | 2 | 0.8 | -27.1 |
| 50000 | 150000 | 1 | 5 | 784386 | 3 | 0 | 26.89 | 894591 | 2 | 0.82 | -14 |
| 50000 | 200000 | 1 | 2.5 | 703614 | 3 | 0 | 27.22 | 994591 | 2 | 1.12 | -41.4 |
| 50000 | 200000 | 1 | 5 | 784386 | 3 | 0 | 26.74 | 994591 | 2 | 1.23 | -26.8 |
| 50000 | 300000 | 1 | 2.5 | 703614 | 3 | 0 | 44.45 | 1106447 | 1 | 1.4 | -57.3 |
| 50000 | 300000 | 1 | 5 | 784386 | 3 | 0 | 45.41 | 1106447 | 1 | 1.45 | -41.1 |
| 50000 | 500000 | 1 | 2.5 | 703614 | 3 | 0 | 45.12 | 1306447 | 1 | 1.46 | -85.7 |
| 50000 | 500000 | 1 | 5 | 784386 | 3 | 0 | 46 | 1306447 | 1 | 1.49 | -66.6 |

Table C.1 (cont.): Comparison of the Heuristic Algorithm with SPLP for *BOL*

| Parameters | | | | Heuristic Algorithm | | | | SPLP | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $C_{CF}$ | $C_{DF}$ | $C_{LN}$ | $C_{UN}$ | **Objective** | **# of CF** | **# of DF** | **Time (sec.)** | **Objective** | **# of DF** | **Time (sec.)** | **Comparison (%)** |
| 5000 | 10000 | 1 | 2.5 | 318231 | 0 | 14 | 73.41 | 318231 | 14 | 0.63 | 0 |
| 5000 | 10000 | 1 | 5 | 318231 | 0 | 14 | 72.07 | 318231 | 14 | 0.55 | 0 |
| 5000 | 15000 | 1 | 2.5 | 378347 | 0 | 10 | 260.15 | 378347 | 10 | 0.59 | 0 |
| 5000 | 15000 | 1 | 5 | 378347 | 0 | 10 | 92.11 | 378347 | 10 | 0.61 | 0 |
| 5000 | 20000 | 1 | 2.5 | 419803 | 9 | 3 | 263.49 | 424207 | 8 | 0.62 | -1 |
| 5000 | 20000 | 1 | 5 | 424207 | 0 | 8 | 103.76 | 424207 | 8 | 0.63 | 0 |
| 5000 | 30000 | 1 | 2.5 | 427161 | 12 | 0 | 259.76 | 503332 | 7 | 0.67 | -17.8 |
| 5000 | 30000 | 1 | 5 | 501782 | 1 | 6 | 343.65 | 503332 | 7 | 0.6 | -0.3 |
| 5000 | 50000 | 1 | 2.5 | 427161 | 12 | 0 | 274.43 | 613255 | 5 | 0.67 | -43.6 |
| 5000 | 50000 | 1 | 5 | 571367 | 9 | 0 | 272.08 | 613255 | 5 | 0.68 | -7.3 |
| 50000 | 100000 | 1 | 2.5 | 703614 | 3 | 0 | 21.97 | 772841 | 3 | 0.68 | -9.8 |
| 50000 | 100000 | 1 | 5 | 772841 | 0 | 3 | 22.44 | 772841 | 3 | 0.66 | 0 |
| 50000 | 150000 | 1 | 2.5 | 703614 | 3 | 0 | 26.31 | 894591 | 2 | 0.8 | -27.1 |
| 50000 | 150000 | 1 | 5 | 784386 | 3 | 0 | 26.89 | 894591 | 2 | 0.82 | -14 |
| 50000 | 200000 | 1 | 2.5 | 703614 | 3 | 0 | 27.22 | 994591 | 2 | 1.12 | -41.4 |
| 50000 | 200000 | 1 | 5 | 784386 | 3 | 0 | 26.74 | 994591 | 2 | 1.23 | -26.8 |
| 50000 | 300000 | 1 | 2.5 | 703614 | 3 | 0 | 44.45 | 1106447 | 1 | 1.4 | -57.3 |
| 50000 | 300000 | 1 | 5 | 784386 | 3 | 0 | 45.41 | 1106447 | 1 | 1.45 | -41.1 |
| 50000 | 500000 | 1 | 2.5 | 703614 | 3 | 0 | 45.12 | 1306447 | 1 | 1.46 | -85.7 |
| 50000 | 500000 | 1 | 5 | 784386 | 3 | 0 | 46 | 1306447 | 1 | 1.49 | -66.6 |

Table C.1 (cont.): Comparison of the Heuristic Algorithm with SPLP for *Guinea*

| Parameters | | | | Heuristic Algorithm | | | | SPLP | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $C_{CF}$ | $C_{DF}$ | $C_{LN}$ | $C_{UN}$ | **Objective** | **# of CF** | **# of DF** | **Time (sec.)** | **Objective** | **# of DF** | **Time (sec.)** | **Comparison (%)** |
| 5000 | 10000 | 1 | 2.5 | 532009 | 0 | 21 | 819.41 | 532009 | 21 | 10.75 | 0 |
| 5000 | 10000 | 1 | 5 | 532009 | 0 | 21 | 819.88 | 532009 | 21 | 9.66 | 0 |
| 5000 | 15000 | 1 | 2.5 | 628363 | 0 | 18 | 1152.52 | 628363 | 18 | 12.92 | 0 |
| 5000 | 15000 | 1 | 5 | 628363 | 0 | 18 | 1111.71 | 628363 | 18 | 6.99 | 0 |
| 5000 | 20000 | 1 | 2.5 | 705052 | 0 | 15 | 1643.85 | 705052 | 15 | 13.2 | 0 |
| 5000 | 20000 | 1 | 5 | 705052 | 0 | 15 | 1360.84 | 705052 | 15 | 12.5 | 0 |
| 5000 | 30000 | 1 | 2.5 | 818206 | 3 | 8 | 2424.94 | 822941 | 11 | 9.98 | -0.6 |
| 5000 | 30000 | 1 | 5 | 822941 | 0 | 11 | 1930.22 | 822941 | 11 | 21.18 | 0 |
| 5000 | 50000 | 1 | 2.5 | 956837 | 22 | 0 | 2580.78 | 1028271 | 9 | 9.58 | -7.5 |
| 5000 | 50000 | 1 | 5 | 1028271 | 0 | 9 | 2573.05 | 1028271 | 9 | 12.98 | 0 |
| 50000 | 100000 | 1 | 2.5 | 1364139 | 2 | 5 | 446.39 | 1394829 | 7 | 14.36 | -2.2 |
| 50000 | 100000 | 1 | 5 | 1394829 | 0 | 7 | 457.86 | 1394829 | 7 | 16.2 | 0 |
| 50000 | 150000 | 1 | 2.5 | 1489759 | 8 | 0 | 468.9 | 1744829 | 7 | 20.73 | -17.1 |
| 50000 | 150000 | 1 | 5 | 1683449 | 2 | 5 | 460.69 | 1744829 | 7 | 21.83 | -3.6 |
| 50000 | 200000 | 1 | 2.5 | 1489759 | 8 | 0 | 670.52 | 2067926 | 6 | 21.23 | -38.8 |
| 50000 | 200000 | 1 | 5 | 1916829 | 5 | 2 | 672.18 | 2067926 | 6 | 24.4 | -7.9 |
| 50000 | 300000 | 1 | 2.5 | 1489759 | 8 | 0 | 832.46 | 2583885 | 5 | 25.87 | -73.4 |
| 50000 | 300000 | 1 | 5 | 1945709 | 7 | 0 | 838.84 | 2583885 | 5 | 27.19 | -32.8 |
| 50000 | 500000 | 1 | 2.5 | 1489759 | 8 | 0 | 992.77 | 3436257 | 4 | 33.4 | -130.7 |
| 50000 | 500000 | 1 | 5 | 1945709 | 7 | 0 | 1032.5 | 3436257 | 4 | 34.55 | -76.6 |

Table C.1 (cont.): Comparison of the Heuristic Algorithm with SPLP for $LGA$

| Parameters | | | | Heuristic Algorithm | | | | SPLP | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $C_{CF}$ | $C_{DF}$ | $C_{LN}$ | $C_{UN}$ | **Objective** | **# of CF** | **# of DF** | **Time (sec.)** | **Objective** | **# of DF** | **Time (sec.)** | **Comparison (%)** |
| 5000 | 10000 | 1 | 2.5 | 683572 | 0 | 40 | 1296.39 | 683572 | 40 | 12.23 | 0 |
| 5000 | 10000 | 1 | 5 | 683572 | 0 | 40 | 1215.77 | 683572 | 40 | 12.19 | 0 |
| 5000 | 15000 | 1 | 2.5 | 861782 | 0 | 32 | 2802.22 | 861782 | 32 | 16.96 | 0 |
| 5000 | 15000 | 1 | 5 | 861782 | 0 | 32 | 1811.62 | 861782 | 32 | 11.31 | 0 |
| 5000 | 20000 | 1 | 2.5 | 1016068 | 0 | 29 | 8973.27 | 1016068 | 29 | 10.42 | 0 |
| 5000 | 20000 | 1 | 5 | 1016068 | 0 | 29 | 2121.11 | 1016068 | 29 | 8.67 | 0 |
| 5000 | 30000 | 1 | 2.5 | 1186840 | 29 | 4 | 5345.03 | 1275602 | 23 | 14.86 | -7.5 |
| 5000 | 30000 | 1 | 5 | 1275602 | 0 | 23 | 7439.68 | 1275602 | 23 | 15.14 | 0 |
| 5000 | 50000 | 1 | 2.5 | 1212205 | 32 | 1 | 5906.5 | 1673562 | 16 | 10.62 | -38.1 |
| 5000 | 50000 | 1 | 5 | 1673562 | 0 | 16 | 6980.39 | 1673562 | 16 | 13.09 | 0 |
| 50000 | 100000 | 1 | 2.5 | 2184179 | 12 | 1 | 869.72 | 2293837 | 11 | 14.27 | -5 |
| 50000 | 100000 | 1 | 5 | 2293837 | 0 | 11 | 874.82 | 2293837 | 11 | 18.33 | 0 |
| 50000 | 150000 | 1 | 2.5 | 2192306.5 | 13 | 0 | 1317.91 | 2765295 | 9 | 19.18 | -26.1 |
| 50000 | 150000 | 1 | 5 | 2620287 | 9 | 1 | 1310.52 | 2765295 | 9 | 20.59 | -5.5 |
| 50000 | 200000 | 1 | 2.5 | 2192306.5 | 13 | 0 | 1881.53 | 3138311 | 7 | 19 | -43.2 |
| 50000 | 200000 | 1 | 5 | 2641757 | 10 | 0 | 1876.4 | 3138311 | 7 | 17.27 | -18.8 |
| 50000 | 300000 | 1 | 2.5 | 2192306.5 | 13 | 0 | 2663.13 | 3770383 | 5 | 27.78 | -72 |
| 50000 | 300000 | 1 | 5 | 2641757 | 10 | 0 | 2669.18 | 3770383 | 5 | 29.62 | -42.7 |
| 50000 | 500000 | 1 | 2.5 | 2192306.5 | 13 | 0 | 2890.27 | 4603902 | 4 | 35.26 | -110 |
| 50000 | 500000 | 1 | 5 | 2641757 | 10 | 0 | 3017.98 | 4603902 | 4 | 38.02 | -74.3 |