A MULTI-LEVEL CONTINUOUS MINIMAX LOCATION PROBLEM WITH REGIONAL DEMAND

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

BY

AMIN FARIDYAHYAEI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

SEPTEMBER 2017

Approval of the thesis:

A MULTI-LEVEL CONTINUOUS MINIMAX LOCATION PROBLEM WITH REGIONAL DEMAND

submitted by **AMIN FARIDYAHYAEI** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Yasemin Serin _____
Head of Department, **Industrial Engineering**

Assist. Prof. Dr. Mustafa Kemal Tural _____
Supervisor, **Industrial Engineering Department, METU**

Examining Committee Members:

Assoc. Prof. Dr. Canan Sepil
Industrial Engineering Department, METU _____

Assist. Prof. Dr. Mustafa Kemal Tural
Industrial Engineering Department, METU _____

Assist. Prof. Dr. Banu Lokman
Industrial Engineering Department, METU _____

Assist. Prof. Dr. Bahar Çavdar
Industrial Engineering Department, METU _____

Assist. Prof. Dr. Özlem Karsu
Industrial Engineering Department, Bilkent University _____

Date: _____

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

Name, Last Name: AMIN FARIDYAHYAEI

Signature :

ABSTRACT

A MULTI-LEVEL CONTINUOUS MINIMAX LOCATION PROBLEM WITH REGIONAL DEMAND

Faridyahyaei, Amin

M.S., Department of Industrial Engineering

Supervisor : Assist. Prof. Dr. Mustafa Kemal Tural

September 2017, 134 pages

The minimax facility location problem seeks for the optimal locations of the facilities in the plane so that the maximum Euclidean distance between the demanding entities (given points in the plane) and their corresponding nearest facilities is minimized. In the solutions, remote entities (irrespective of their weights) tend to pull the facilities toward themselves which may result in larger distances for the other entities.

In this thesis, we consider a multi-level minimax location problem which allows some of the entities to be covered in outer levels and thereby reducing their impact on the facility locations. We assume that associated with each entity, there is a weight which represents its importance, e.g., weights might represent populations if the entities are districts or cities. Additionally, we consider entities as regions in the plane consisting of an infinite number of points, therefore, this problem is a multi-level version of the minimax location problem with continuous demand. Based on the nature of the problem, the farthest point of each region to its nearest facility is important and Euclidean distance is utilized in distance calculations.

In this thesis, firstly, we model the single and multi-facility versions of the considered problem as mixed integer second order cone programming (MISOCP) problems. Secondly, we perform computational experiments on artificially generated instances to see the limits of the mathematical programming formulations. Then, we propose

several heuristics and compare them with the MISOCP formulations in terms of solution quality and computational time. Finally, all these solution approaches are tested on the case study of Istanbul.

Keywords: Minimax Problem, Second Order Cone Programing, Regional Demand, Facility Location, Location-allocation

ÖZ

BÖLGESEL TALEPLİ ÇOK SEVİYELİ SÜREKLİ MİNİMAX YER SEÇİMİ PROBLEMİ

Faridyahyaei, Amin
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi : Yrd. Doç. Dr. Mustafa Kemal Tural

Eylül 2017 , 134 sayfa

Minimax yer seçimi problemi, talep birimleri (düzlemde verilen noktalar) ve bunlara karşılık gelen en yakın tesisler arasındaki maksimum Öklid mesafesini enaza indirecek düzlemdeki tesislerin en uygun yerlerini arar. Çözümlerde, uzak birimler (ağırlıklarından bağımsız olarak) tesisleri kendilerine doğru çekme eğilimindedir ve bu da diğer birimler için daha büyük mesafelere sebep olabilir.

Bu tezde, bazı birimlerin dış seviyelerde kapsanmasına izin verilen ve böylece tesis yerleri üzerindeki etkilerini azaltan çok seviyeli bir minimax yer seçimi problemi göz önüne alıyoruz. Her birimle ilişkili olarak, önemini temsil eden bir ağırlık olduğunu varsayıyoruz; örneğin, talep birimleri ilçeler veya şehirler ise, ağırlıklar nüfusu temsil edebilir. Ek olarak, birimleri sonsuz sayıda noktadan oluşan düzlemdeki bölgeler olarak ele alıyoruz; bu nedenle, bu problem, sürekli minimax yer seçimi problemin çok seviyeli bir versiyonudur. Problemin doğasına bağlı olarak, her bölgenin en yakın tesise en uzak noktası önem arz etmekte ve mesafe hesaplamalarında Öklid uzaklığı kullanılmaktadır.

Bu tezde, ilk olarak, çalıştığımız problemin tek ve çok tesisli versiyonlarını karışık tam sayılı ikinci dereceden konik programlama (KTİDKP) problemleri olarak modelliyoruz. İkinci olarak, matematiksel programlama formülasyonlarının sınırlarını görmek için yapay olarak üretilen örnekler üzerinde hesaplama deneyleri yapıyoruz.

Daha sonra, birkaç sezgisel çözüm yöntemi öneriyoruz ve bunları KTİDKP formülasyonları ile çözüm kalitesi ve hesaplama süresi açısından karşılaştırıyoruz. Son olarak, bütün bu çözüm yöntemlerini İstanbul vaka çalışması üzerinde test ediyoruz.

Anahtar Kelimeler: Minimax Problemi, İkinci Dereceden Konik Programlama, Bölgesel Talep, Tesis Yeri Seçimi, Konum-tahsis

To my family for their endless support and encouragement

ACKNOWLEDGMENTS

Foremost, I would like to thank my thesis supervisor Assist. Prof. Dr. Mustafa Kemal Tural for his guidance, support, and encouragement. His office door was always open whenever I ran into a trouble spot or had a question about my thesis. I learned a lot from him and it was a big honor for me to work with him.

I would like to thank my friends for sharing with me their experiences and memories specially Saleh Farham, Mahdi Farshchi, Arsham Atashi and Azar Arghavanian for their helps and guidance.

I also wish to appreciate my thesis committee members Assoc. Prof. Dr. Canan Sepil, Assist. Prof. Dr. Banu Lokman, Assist. Prof. Dr. Bahar Cavdar and Assist. Prof. Dr. Ozlem Karsu for their invaluable feedbacks.

Finally, and the most importantly, I declare my deepest appreciation to my family for their endless love, support and encouragement.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xv

LIST OF FIGURES xviii

CHAPTERS

1 INTRODUCTION 1

2 LITERATURE REVIEW 7

2.1 Facility Location Problem Models 8

2.1.1 Minisum Location Problems 9

2.1.2 Minimax Location Problems 9

2.1.3 Covering Location Problems 10

2.2 Number of Facilities 11

2.3 Solution Space 11

2.3.1 Discrete Facility Location Problems 11

2.3.2 Continuous Facility Location Problems 12

2.4 Structure of Demand Entities 13

2.5 Structure of Covering Level 14

3 PROBLEM FORMULATION 15

3.1 Mathematical formulation of the single facility case 16

3.1.1 Notations of the single facility case 16

3.1.2 SOCP formulation for the single level problem (classical Minimax) 16

3.1.3 MISOCP formulation for bi-level problem 18

3.1.4 MISOCP formulation for k-level problem 19

3.1.5 Choosing the value of M_h 20

3.1.6 Time complexity of the single facility case 28

3.2 Mathematical formulation of the multi-facility case 29

3.2.1 MISOCP formulation of the multi-facility, k-level problem 29

4 HEURISTIC SOLUTION APPROACHES 33

4.1 Heuristics for the single facility case 33

4.1.1 Critical Level Heuristic (CLH) 34

4.1.2 Location Leveling Heuristic (LLH) 37

4.1.3 Critical Level and Demand Exchange Heuristic (CLDEH) 40

4.1.4 Critical Level, Upper Demand Exchange Heuristic (CLUDEH) 45

4.1.5 Descent Critical Level, Demand Exchange Heuristic (DCLDEH) 49

4.1.6 Area Abstraction Heuristic (AAH) 51

4.2 Heuristics for the multi-facility case 55

4.2.1 Location-Allocation-Leveling Heuristic (LALH) . 58

4.2.2 Location-Allocation-Leveling, Critical Level Heuristic (LAL-CLH) 61

4.2.3 Location-Allocation-Leveling, Location-Leveling Heuristic (LAL-LLH) 62

4.2.4 Location-Allocation-Leveling, Critical Level Demand Exchange Heuristic (LAL-CLDEH) 64

4.2.5 Location-Allocation-Leveling, Critical Level Upper Demand Exchange Heuristic (LAL-CLUDEH) 65

4.2.6 Location-Allocation-Leveling, Descent Critical Level Demand Exchange Heuristic (LAL-DCLDEH) . . 66

4.2.7 Location-Allocation-Leveling, Area Abstraction Heuristic (LAL-AAH) 67

5 COMPUTATIONAL STUDIES 69

5.1 Computational study for the single facility case 69

5.1.1 Instance generation 69

5.1.2 Parameter settings of the heuristics 70

5.1.3 Computational results 79

5.1.3.1 Exact solution 79

5.1.3.2 Heuristics' solutions 81

5.2 Computational study of the multi-facility case 99

5.2.1 Instance generation 99

5.2.2 Parameter settings of the heuristics 101

5.2.3 Computational results 103

5.2.3.1 Exact solution 104

5.2.3.2 Heuristic's solutions 107

6 CASE STUDY OF ISTANBUL 119

7 CONCLUSION AND FUTURE STUDY DIRECTIONS 127

REFERENCES 131

LIST OF TABLES

TABLES

Table 3.1 Comparison of the proposed approaches for Big-M 27

Table 4.1 Comparison between the CLDEH versions 45

Table 4.2 Comparison between the CLUDEH versions 47

Table 4.3 Comparison between the DCLDEH versions 51

Table 5.1 Randomly Generated Instances 71

Table 5.2 CLH parameter setting 72

Table 5.3 Parameter setting for CLDEH 73

Table 5.4 Parameter setting for CLDEH 73

Table 5.5 Parameter setting for CLDEH 74

Table 5.6 Parameter setting for DCLDEH 75

Table 5.7 Parameter setting for DCLDEH 76

Table 5.8 Parameter setting for AAH 76

Table 5.9 Parameter setting for AAH 77

Table 5.10 Parameter setting for AAH 77

Table 5.11 Parameter setting for AAH 78

Table 5.12 Parameter setting for all heuristics 78

Table 5.13 Optimal or the best found objective function values by the MISOCP formulation and the associated computation times (in seconds) 80

Table 5.14 The best found objective value between MISOCP formulation and heuristics 82

Table 5.15 Average of the % deviation of heuristics from best found solution . . 83

Table 5.16 Minimum of the % deviation of heuristics from best found solution . 84

Table 5.17 Computation time (in seconds) of the heuristics associated to the solutions of the Table 5.16 86

Table 5.18 Number of initial locations checked by the heuristics in a time limit . 88

Table 5.19 Average % deviation of the heuristics for running them in a time limit 89

Table 5.20 Minimum % Deviation of the heuristics for running them in a time limit 90

Table 5.21 Random Generated Instances 100

Table 5.22 LALH parameter setting 102

Table 5.23 LALH parameter setting 102

Table 5.24 LAL-CLDEH parameter setting 103

Table 5.25 LAL-CLDEH parameter setting 104

Table 5.26 Optimal or the best found objective function values by MISOCP formulation and associated running times (in seconds) 105

Table 5.27 Optimal or the best found objective function values by MISOCP formulation and associated running times (in seconds) 106

Table 5.28 Minimum % deviation of the heuristics from the best found solution 108

Table 5.29 Minimum % deviation of the heuristics from the best found solution 109

Table 5.30 Average % deviation of the heuristics from the best found solution . 110

Table 5.31 Average % deviation of the heuristics from the best found solution . 111

Table 5.32 Running time (in seconds) to obtain solutions of Tables 5.28 and 5.29 112

Table 5.33 Running time (in seconds) to obtain solutions of Tables 5.28 and 5.29 113

Table 6.1 Districts of Istanbul 121

Table 6.2 Objective function of the MISOCP formulation and minimum of the % deviation of heuristics from best found solution for single facility case . 122

Table 6.3 Objective function of the MISOCP formulation and average of the % deviation of heuristics from best found solution for single facility case . 122

Table 6.4 Comparison between the computation time (in second) of the MISOCP formulation and heuristics' solution related to Table 6.2. 122

Table 6.5 Objective function value of the MISOCP formulation and minimum of % deviation of heuristics from best found solution for multi-facility case 124

Table 6.6 Objective function value of the MISOCP formulation and average of % deviation of heuristics from best found solution for multi-facility case . 124

Table 6.7 Computation time (in second) comparison between the MISOCP formulation and heuristics' solution related to the Table 6.5. 125

LIST OF FIGURES

FIGURES

Figure 1.1 Comparison of unweighted, weighted and multi-level Minimax location problem for the single facility case 4

Figure 3.1 Proposition 2 23

Figure 4.1 The steps of the CLH heuristic 36

Figure 4.2 The steps of the LLH heuristic 39

Figure 4.3 The steps of the CLDEH heuristic 44

Figure 4.4 The steps of the CLUDEH heuristic 48

Figure 4.5 The steps of the AAH heuristic 53

Figure 4.6 The steps of the LAL algorithm 57

Figure 4.7 The steps of the LALH heuristic 59

Figure 4.8 The steps of the LAL-CLH heuristic 63

Figure 5.1 Deviations of heuristics for all instances 91

Figure 5.2 Facility locations found with heuristics for instance number 10 . . . 91

Figure 5.3 Facility locations found with heuristics for instance number 15 . . . 92

Figure 5.4 Facility locations found with heuristics for instance number 24 . . . 92

Figure 5.5 Computation time of heuristics for all instances 93

Figure 5.6 Computation time of CLH with respect to the number of regions and levels 93

Figure 5.7 Computation time of LLH with respect to the number of regions and levels 94

Figure 5.8 Computation time of CLDEH with respect to the number of regions and levels 94

Figure 5.9 Computation time of CLUDEH with respect to the number of regions and levels 95

Figure 5.10 Computation time of DCLDEH with respect to the number of regions and levels 95

Figure 5.11 Computation time of AAH with respect to the number of regions and levels 96

Figure 5.12 Performance of heuristics in 100 replications of instance number 15 97

Figure 5.13 Facility location found in each iteration of heuristics for instance number 10 98

Figure 5.14 Minimum % deviation of LAL-CLDEH, LAL-DCLDEH and LAL-AAH for all instances 114

Figure 5.15 Minimum % deviation of LALH, LAL-CLH, LAL-LLH and LAL-CLUDEH for all instances 115

Figure 5.16 Comutation time of LALH (in seconds) for all instances 115

Figure 5.17 Facility locations found with the heuristics for the instance number 21 116

Figure 5.18 Facility locations found with the heuristics for the instance number 53 116

Figure 5.19 Covering levels for the instance number 41 117

Figure 6.1 Districts map of Istanbul 120

Figure 6.2 Single facility locations found for single-level (green circle), bi-level (red star) and three-level (blue square) cases of Istanbul 123

Figure 6.3 Multi facility locations found for single-level(Green), bi-level(Red) and three-level(Blue) cases of Istanbul 126

CHAPTER 1

INTRODUCTION

Every person encounters with location decisions in his daily life frequently. Where to place something and its accessibility are the main questions in all location decisions. The answer of these questions are influenced by different criteria which should be determined by the decision maker. Typically, minimizing of the cost is the common aim of location decisions. Private sector considers minimizing financial costs as the main objective of the location decisions. However, governments usually consider social costs as well. For instance, in locating hospitals or fire stations for a district, some other social factors are involved too. In general, Facility Location Problem (FLP) seeks to position one or more facilities in an optimal way to satisfy requirements of some demands as well as criteria of decision maker.

For each scenario in location problems, some fixed and variable costs are associated like setup costs or transportation costs. Transportation costs are commonly measured by distance or required time for traveling from one facility to corresponding demand entities. Hence, distance or time is utilized as one of the main cost criteria in modeling of the facility location problems. There are 3 main models for FLP which are Minisum Location Problems, Minimax Location Problems and Covering Location Problems.

1. Minisum problem: The objective function of this model is minimization of the sum of weighted distance between demand entities and facility(s).

2. Minimax problem: It focuses on minimizing the maximum distance between demands and its closest facility. In the public sector where satisfying all de-

mands has the same importance for the decision maker, minimax model comes to use.

3. Covering problem: This problem can be considered in two different models. Maximal Covering Location Problem (MCLP) which aims at providing as much coverage as possible for demand entities with a fixed covering radius and number of facilities. However, Set Covering Problem (SCP) focuses on total covering of the demand entities with minimum number of facilities.

In facility location problems, 3 different kinds of distance measures are often used which are rectilinear distance, Euclidean distance, and Chebyshev distance. Utilizing each of these distance measures could change not only formulation of the model but also the complexity of the problem. In this thesis, we are studying a pre-positioning problem in humanitarian logistics, which is a public and social oriented area of study.

Humanitarian logistics consider supply chain management for natural disasters or complex emergencies. It includes some pre-disaster during disaster and post-disaster actions in order to alleviate the suffering of affected people [21]. One of the main pre-disaster activities is stock pre-positioning in some distribution centers to increase efficiency of the relief operations. Consequently, location and number of these centers could influence quality of relief services provided to vulnerable people [2].

Typically, several abstractions are done in modeling of such a problem. One of the most common abstractions is representing all demands as some points while demands (affected people in this case) are distributed continuously on the plane. For obtaining more realistic solutions for the problem, considering demands as regions instead of points seems reasonable. Hence, in this thesis demand regions are considered as demand entities. Suzuki and Drezner [47] studied for the first time Minimax location problem considering demand regions instead of points and proposed a geometric based heuristic for the problem. Then, demand regions are applied for other location models by others. More details on employing demand regions in facility location problems can be found in [16].

Drezner and Wesolowski [20] presented a new approach in dealing with demand entities. They considered lots of points on the plane to represent demand entities and then

grouped them to decrease the complexity of the problem. They suggested three distance measures to calculate the distance between group of points and facilities which are the distance of the closest point of the group to the facility, the distance of the farthest point and average distance. Then such an idea is also used in demand regions to estimate the distance from the facilities. For instance, [6] has utilized closest distance in a Minimax problem while [31] and [45] have applied farthest and average distance for a Minisum location problem, respectively. Based on the nature of the studied problem in this thesis, the Euclidean distance between the facility and the farthest point of the demand region is utilized to take the worst-case scenario into account.

The most important contribution in this thesis is considering multi-level coverage for the Minimax problem. Such an idea has not been studied in the literature before and we investigate its effectiveness on randomly generated instances as well as a real case study of the Istanbul.

In the classical Minimax location problem, remote entities (irrespective of their weights) tend to pull the facilities toward themselves which may result in larger distances for the other entities. For confronting this problem, weighted Minimax location problem was proposed. It reduces the effect of less important demand entities on the solution but does not eliminate it completely. The multi-level Minimax location problem studied in this thesis considers several coverage levels with specific covering percentages and in each level provides coverage for that particular percent of the demand entities. This model ensures covering important entities in the inner levels and others in outer levels. The multi-level Minimax location problem results in lower maximum distance between demand entities and facilities in the inner levels by reducing the effect of demand regions in the outer levels on the facility location.

Consider Figure 1.1 which illustrates comparison of 3 mentioned versions of the Minimax location problem. Polygons and circles in the figure represent demand regions and corresponding weight of each region has been indicated on it. The empty circle (blue) represents the optimal solution for the unweighted Minimax location problem while empty square (purple) and fill circle (red) ones are related to weighted and multi-level versions, respectively. It is clear that the optimal solution of the unweighted model is close to all demand entities irrespective of their weights. However,

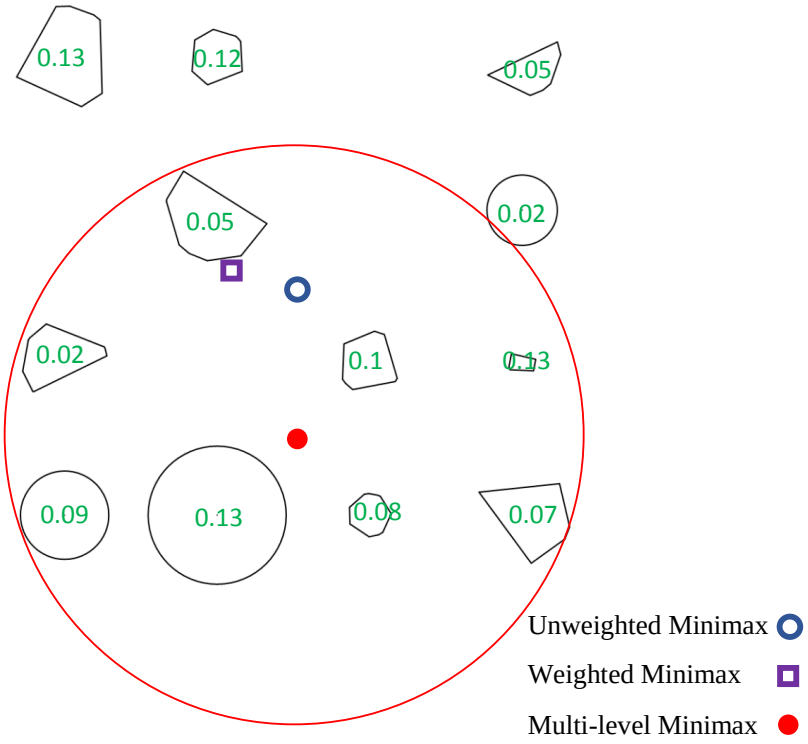


Figure 1.1: Comparison of unweighted, weighted and multi-level Minimax location problem for the single facility case

the optimal location of the facility in the weighted version is farther away from the demand entities on the right hand side which have lower weights. The multi-level case in this instance considers 2 levels. 65 percent of coverage is provided in the first level which is indicated by the inner circle and the others are covered in the second level. Note that the location of the facility is affected by the entities in both the inner and the outer levels. However, the impact of the outer level can be controlled by the decision maker as will be explained later.

In summary, the problem studied in this thesis is locating several facilities on the continuous plane in order to cover all convex demand regions where the impact of the demand regions inside of a level on the facility location is controlled by the decision maker.

Euclidean distance makes this problem nonlinear. Second order cone programming (SOCP) is utilized to obtain the mathematical programming formulation of the problem over some second order cone constraints. SOCP guarantees getting the optimal

solution for nonlinear models which can be formulated by it. Since our formulation contains allocation of the demand entities for the various levels and facilities, it has binary variables. Hence, Mixed Integer SOCP (MISOCP) is applied in this thesis. SOCP is solvable in polynomial time but binary variables increase the complexity of the problem dramatically. Small instances in both single and multi-facility version of the problem could be solved by SOCP in reasonable time but for big instances, several heuristics are proposed and the results of heuristic are compared with optimal solutions of small instances to assess quality of the heuristic algorithms.

Heuristics are developed by employing the idea of location-allocation algorithm. The proposed heuristics in both single facility and multi one allocate demands to levels and facility(s) iteratively to get near optimal solutions. Additionally, in the single facility case Area Abstraction heuristic (AAH) is also developed which is a geometric based algorithm.

Finally, the mathematical programming formulations and introduced heuristic solution approaches are applied for the Istanbul case to assess performance of the model for real data.

The outline of the thesis is as follows. Firstly, an overview of location problems and classifications related to our model are given in Chapter 2. Then, in Chapter 3, the mathematical modeling of the problem is presented. Chapter 4 describes the developed heuristics for both single and multi-facility case. Computational results for randomly generated instances are reported in Chapter 5. In Chapter 6, Istanbul case is presented and finally Chapter 7 concludes the thesis.

CHAPTER 2

LITERATURE REVIEW

Selecting of the optimal location is one of the initial and most crucial decisions in all businesses which could guarantee its prosperity for many years. Hence, it has been studied in different sciences such as Geographic Information System, Industrial Engineering, Management Science and etc. This causes a big challenge in providing a comprehensive overview of the facility location problem.

Human beings have been involved with location decisions for accommodation, agriculture and other aspects since their earliest life days. Therefore, there is not any specific time as the origin of studying location problems but the ideas of Fermat at the 17th century had a significant impact on the formation of the modern facility location problem.

He considered the problem of locating a new point on the plane given to the 3 prepositioned ones, such that sum of the distance between them is minimized. This problem was solved by Torricelli with applying the focal property of the ellipses and called as Fermat – Torricelli point . Then, Alfred Weber (1909) formulated the general case of this problem mathematically, which leads in the future massive amounts of the researches in this area.

Weber problem seeks the optimal location of a facility in the plane (x^*, y^*) considering the position of n fixed demand points (a_i, b_i) with weights of w_i where the Euclidean distance between facility and each demand is $d_i(x, y)$.

$$\begin{aligned} \min_{x,y} \quad & W(x,y) = \sum_{i=1}^{n} W_i d_i(x,y) \\ \text{s.t.} \quad & d_i(x,y) = \sqrt{(x-a_i)^2 + (y-b_i)^2} \end{aligned} \tag{2.1}$$

Weiszfeld [51] proposed an iterative algorithm to solve the Weber problem efficiently which is called as Weiszfeld procedure. Then, Hakimi [26] presented a seminal paper for modeling of the different facility location problems that would be discussed in the following.

As it was mentioned above, due to the great amount of researches in this area, classification of Facility Location Problems (FLP) is difficult. According to the scope of our research, we present an overview of location problems in the following manner. Firstly, FLP is considered based on the modeling of the problem in Section 2.1. In Section 2.2, second classification is provided given to the number of assumed facilities for the problem. The third category considers the solution space of the problem and it is explained in Section 2.3. The fourth classification of FLP is based on the structure of the demand entities which is mentioned in Section 2.4 and finally, in Section 2.5, the last category is described given to the form of the covering demand entities.

2.1 Facility Location Problem Models

Modeling of the problem is one of the first steps in the evaluating of that problem. The objective function should be stated in the quantitative form to be able of utilizing the mathematic formulations and finding the optimal values for the problem. The most common objective function for the FLP is minimizing costs. Cost can be stated as a function of time or distance between the facility(s) and the demand entities. Different approaches are proposed in the modeling of the location problems but the main categorization for FLP models contains 3 classes of Minisum Location Problems, Minimax Location Problems and Covering Location Problems [15].

2.1.1 Minisum Location Problems

Minisum problem focuses on minimizing the sum of weighted distance between demand entities and facility(s) which is also called as P-median problem. Both Fermat – Torricelli and Weber problems consider the weighted distance but the first systematical formulation for P-median problem was done by Hakimi [26]. Two different distance measures are usually used in the location problems which utilizing each of these distances can make some differences in the modeling of the problem. Consider n distinct points on the plane $P_i = (a_i, b_i)$

1. Rectilinear Distance: It calculates the distance between two points by summing of the absolute differences for both X and Y coordinates of points.
   $d_i(X, P_i) = |X - a_i| + |Y - b_i|$
   Minisum location problem with rectilinear distance has been studied in lots of papers such as [43] and [28].

2. Euclidean Distance: This measures the straight distance between two points.
   $d_i(X, P_i) = \sqrt{(X - a_i)^2 + (Y - b_i)^2}$
   Most of the studies use Euclidean distance as their measure, since it could represents the spatial location problems, perfectly. Weber problem is one of the outstanding studies that has applied this distance.

2.1.2 Minimax Location Problems

This kind of FLP aims at minimizing the maximum distance between demands and their closest facility [26]. The primal application of Minimax problem was in the positioning of the public facilities where the minimum distance of farthest customer and its corresponding facility should be achieved. It is also called as P-center problem and has both weighted [18], [36], [19] and unweighted versions [26], [50] in the literature.

2.1.3 Covering Location Problems

The third class in FLP models is covering location problems. Covering problem is one of the interesting areas for researchers because of its extensive applications in real world life especially for service and emergency facilities. Covering problems are classified in two categories.

1. Set covering problem: This model tries to minimize location cost satisfying the specific level of coverage. Hence, this problem aims at providing total coverage with the lowest number of facilities [40].

2. Maximal coverage location problem (MCLP): It maximizes the coverage of demands within a desired service standard S by locating a fixed number of facilities. Accordingly, the problem optimizes provided amount of coverage based on specific number of facilities. MCLP was first proposed by Church and ReVelle [10] in 1974 and then applied for lots of problems in recent years.

Covering problems have been widely used to assist facility placement for provision of various types of services. Zanjani Farahani [24] summarized main applications of MCLP in different areas consisting Emergency medical services, Data management, Community warning, Health care and so on. Two types of covering problems have been widely studied, the location set covering problem (LSCP) and the MCLP. While LSCP aims at covering a region of interest entirely, the MCLP tries to maximize the coverage when the budget is not sufficient for a complete coverage of a region. Continuous space siting has been of longstanding interest. Firstly, Weber studied finding the location in the plane for siting a single industry to minimize transportation costs. This problem, then, is called as Weber facility location problem and Church used such an idea to introduce planar maximal covering (PMC) location problem. Church claimed that a discrete set of locations can contain an optimal solution to the continuous-space MCLP by using Circle intersect points set (CIPS) but in his study, demand was represented as discrete points. Murray and Tong [39] extended CIPS method for other kinds of demand objects (points, lines and/or polygons) by introducing Finite Dominating Set (FDS). Furthermore, Mean-Shift algorithm has been also used for finding candidate locations set [29].

2.2 Number of Facilities

One of the initial decisions for the business administrators is deciding about the number of facilities which they want to open. All location models that were explained before, have both single and multi facility versions. Based on [33] and [15] single facility problem has lower complexity in comparison with the multi-facility case. The single facility version of P-center problem is called as 1-center. Lots of geometric ([42], [30], [23], [8], [52]) and mathematical ([22], [41]) methods for its solving have been introduced. Multi-facility location problems are the extension of the single facility version. In the multi-facility case, the allocation of demand entities to the facilities will appear. Allocation procedure creates the binary variables in the problem. Hence, the common method for the multi-facility case is utilizing Location Allocation (LA) algorithm. It was proposed by Cooper [12] and aims at finding optimal locations of m facilities by allocating the demands in a way that the distance between demands and facilities be minimized. In fact, LA solves a single facility model inside of its solution procedure. hence, studying single facility location problems are as important as multi facility ones.

2.3 Solution Space

Next categorization of the facility location problems is based on the solution space which could be discrete or continuous. In the discrete case, there are finite number of locations that can host facilities. Therefore, the problem tries to find the best point(s) among these candidate locations. In the continuous case, all points on the plane are potential locations for the facilities. In the following, discrete and continuous versions of location models are explained briefly.

2.3.1 Discrete Facility Location Problems

Most of the location problems including P-center, P-median and Covering were shown to be NP-hard [35] and [15]. Consequently, lots of studies are done to find more efficient solutions for both discrete and continuous versions of FLP. One of the primal

researches on discrete version of P-median has been done by Christofides [9]. He presents a mathematical method (Lagrangian relaxation) and abstracts solution space to get objective function in lower time. For more researches on discrete P-median you can see [37].

Another discrete location problem is P-center one. Complexity of this problem is $O(n^p)$ where n is the number of candidate locations and p is the number of facilities. This complexity is gained by checking all candidate locations for P facilities one by one to find optimal scenario. Other algorithms with lower complexity were introduced for this problem which try to decrease search space for the problem. An other idea in dealing with discrete P-center problem is converting it to the set covering problem [15] and [38].

2.3.2 Continuous Facility Location Problems

Since our proposed problem is a continuous minimax, continuous version of FLP, which considers all points on the plane as potential facility location (planar problem), has been considered more meticulously. The main continuous P-median problem is Weber problem. The mathematical method of Weiszfeld as a solution approach for solving of the Weber problem was discussed before. Another approach in coping with weber problem is utilizing Location Allocation Model [27].

Minimax continuous problem could be considered in two classes of single facility and multi-facility. 1-center continuous problem usually has been studied by geometric methods. Geometric interpretation of the 1-center problem is locating of a facility such that all regions are covered with minimum covering radius. The main geometric approach for 1-center problem is smallest enclosing circle [53]. Another well known method for 1-center problem has been proposed by Plastria [44]. He has presented the minimal covering circle algorithm which selects two of demands randomly and figures an enclosing circle of these demands. Then, algorithm adds other demands one by one to get total coverage. For P-center continuous problem, the main approach is using Voronoi diagram heuristic. This method was firstly introduced by Suzuki and Okabe [48] and other extensions were added by further researches such as Wei & et al. [50]. The algorithm of this heuristic is similar to the location allocation heuristic.

It figures the voronoi diagrams of the region based on random initial locations and then assigns the center of each voronoi as initial location of next iteration.

Continuous covering problems have also been investigated in so many researches. The first solution for continuous Maximal Covering Location Problem was proposed by Church [11]. He proved that a discrete set of locations can contain an optimal solution to the continuous-space MCLP. The idea was based on the approach which is called Circle intersect points set (CIPS). The extension of CIPS was introduced by Murray and Tong [39] which is called as finite dominating set (FDS). CISP just considers the demands as points but FDS can aslo deal with demand regions. FDS determines the boundary of each demand region by its specific algorithm and then claims that intersection of these boundaries would be the candidate locations for problem. Therefore, it converts the Continuous covering problem to the discrete one.

2.4 Structure of Demand Entities

Researchers usually abstract demand entities to the several points to represent entire demand region. Such an idea makes problem simple and manageable. However, improving computational capabilities increases expectation to present more real models of the problem. Point representation of demand entities can decrease complexity of the problem dramatically, but it provides not accurate solutions for the problem.

Considering demand entities as region is a realistic approach in dealing with location problems, specially when 1) demand entities are not fixed on the plane, 2) there is uncertainty about demands, and 3) there is a huge number of demand points that grouping them as regions reduces complexity of the problem.

Considering demand entities as regions requires new distance measures for demands. Three distance measures have been proposed for regional demands which are as follows [17].

1. Closest distance: This measure is useful in considering product distribution where the aim is delivering good to the closest supplier and he would distribute product to other suppliers.

2. Farthest distance: In the worst-case scenario, where a service should be provided to the farthest client as well, this measure is applied. As an example, hospital and fire stations can be mentioned [17].

3. Expected distance: This measure is the common distance in the literature. when all of the members of a region have the same importance for the decision maker, expected distance is utilized. As an example Bennett and Mirakhor [3] have applied expected distance for minisum problem and have considered the centroid of the region as origin of calculation of distance.

2.5 Structure of Covering Level

As it was mentioned in Section 2.3, minimax and set covering problems have very close relation with each other. In set covering problem, the aim is providing total coverage with smallest number of facilities and in P-center problem, the aim is presenting total coverage with minimum distance. The initial assumption for set covering problem is the fixed range of coverage for each facility. Recently, some researchers introduced variable covering radius instead of the fixed one. Berman & et al. [4] considered a function as the covering radius that is monotonically increasing and called it as variable covering radius. In the other paper, Berman & et al [5] proposed gradual covering problem. They challenged the definite value of covering radius where demands are covered inside of it completely but by increasing epsilon value of distance between demands and facilities, coverage would be partial. They assumed two distances to deal with this challenge. Furthermore, Karasakal and Karasakal [32] extended gradual covering idea in a manner that coverage would be reduced after specific radius, in monotonic trend until completing decay. An other idea can be considering multi covering levels instead of just single coverage radius and level. This idea has not been studied sufficiently. One of the numerated researches in this scope has been done by Carrizosa and Plastria [7]. They have considered two covering levels for discrete minimax problem and have presented an iterative algorithm for solving such a problem.

CHAPTER 3

PROBLEM FORMULATION

Minimax (P-center) problem is a specific type of location problem, which aims at minimizing the maximum distance of the facility (facilities) to all demand entities. The common procedure in dealing with demand entities which are regions is representing them as some points and modelling the problem considering these points. However, such an abstraction underestimates the objective function of the problem. Assuming demand entities as regions provide more realistic outcomes for the problem. As mentioned in Chapter 1, several distance measures could be applied for finding the distance of facility from demand regions. Based on the nature of the studied problem in this thesis, the Euclidean distance between the facility and the farthest point of the demand region is utilized to take the worst-case scenario into account. Furthermore, in this thesis several covering levels are defined for the Minimax location problem where each of these levels provides specific covering percentage of demand regions. It is assumed that the facility provides particular covering range for each level, which is determined by corresponding parameters. For each demand region, there is a weight which represents its importance and the sum of these weights are equal to one. Facility should cover pre-specified percentage of weights of demand regions in each level inside its associated covering range for that level. Hence, the model covers important demand regions in inner levels and others in outer levels where the impact of the outer level can be controlled by the decision maker with parameters of the problem. Consequently, the objective function of the studied model in this thesis is minimizing the covering radius of the facility such that it could provide the specified covering percentages for each level.

3.1 Mathematical formulation of the single facility case

3.1.1 Notations of the single facility case

Regions are considered in two different convex forms which are polygons and disks. Suppose there are n polygons and m disks such that index i and j are used to enumerate them respectively. Additionally, K covering levels are assumed for the problem which are indicated by index h. Covering range of each level, $c_h r + d_h$, is specified by two parameters; radius coefficient (c_h) and covering domain (d_h) of that level. The notations of mathematical formulation of the problem are as follows.

Parameters :

E_i : Set of corners' locations of polygon region i, $i = 1, 2, ..., n$

C_j : Center of disk region j, $j = 1, 2, ..., m$

R_j : Radius of disk region j

α_i : Weight of polygon region i

β_j : Weight of disk region j

c_h : Coefficient of level h coverage

d_h : Domain of level h coverage

P_h : Covering percentage of level h

Decision variables:

fp_i : Farthest distance between the facility and polygon i

fd_j : Farthest distance between the facility and disk j

X : Location of a facility

y_{ih} : Binary variable which equals to 1 (0) if polygon i is covered in level h (other levels)

z_{jh} : Binary variable which equals to 1 (0) if disk j is covered in level h (other levels)

r : Covering radius of the facility

3.1.2 SOCP formulation for the single level problem (classical Minimax)

Firstly, the mathematical formulation for classical minimax problem with demand regions is presented and in the next sections the bi-level and K-level extensions of the

problem would be discussed.

$$
\begin{aligned}
&\text{minimize} && r \\
&\text{subject to} && r \geq fp_i; i = 1, \ldots, n && (3.1)\\
& && r \geq fd_j; j = 1, \ldots, m && (3.2)\\
& && fp_i \geq \|W - X\|,\ W \in E_i; i = 1, \ldots, n && (3.3)\\
& && fd_j \geq \|C_j - X\| + R_j,\ j = 1, \ldots, m && (3.4)
\end{aligned}
$$

Constraints 3.1 and 3.2 provide maximum value of the farthest distance from the demand regions to the facility between both circles and polygons. Constraints 3.3 and 3.4 ensure covering of demand regions inside corresponding covering radius of the facility for polygons and disks, respectively.

In the constraints 3.3 and 3.4 Euclidean distance is utilized which makes the problem nonlinear. In order to deal with nonlinearity of the problem, SOCP is applied to obtain the mathematical programming formulation of the problem over some second order cone constraints.

Second order cone programing (SOCP)

SOCP is a type of convex optimization problems which minimizes a linear objective function over some second order cone constraints. This method can formulate problems with quadratic objective function and linear constraints as well [34]. Its general formulation is as follows where A and C_i are matrices; f_i are scalars and d_i, c, e_i, b are column vectors.

$$
\begin{aligned}
&\text{minimize} && c^T x && (3.5)\\
&\text{subject to} && \|C_i x + d_i\| \leq e_i^T x + f_i,\ i = 1, \ldots, m \\
& && Ax = b
\end{aligned}
$$

SOCP problems are solvable in polynomial time, hence, it is very applicable in quadratic programs. Additionally, SOCP guarantees getting optimal objective function for nonlinear problems which could be formulated by it.

3.1.3 MISOCP formulation for bi-level problem

As it mentioned above, binary variables are utilized in the multi-level Minimax location problem for assigning the demand regions for each covering level of the facility. Hence, the mathematical formulation of the problem would be Mixed Integer SOCP (MISOCP).

Binary variables of the K-level problem are defined in the notation section which 2 indices of *ih* or *jh* enumerate them. For the bi-level Minimax location problem, binary variables can be defined in another way to have fewer number of binary variables for the problem. This modification of the model could help in decreasing the computational time of the problem. Accordingly, the definition of binary variables are changed as bellow.

y_i : Binary variable which equals to 1 (0) if polygon i is covered in level 1 (level 2)

z_j : Binary variable which equals to 1 (0) if disk j is covered in level 1 (level 2)

It should be stated that in this thesis, the coefficient of the first level (c_1) and its domain (d_1) are considered as 1 and 0, respectively. Therefore, for the bi-level problem, the coefficient of the second level is named as c and the domain of the second level is called as d. Now the bi-level Minimax location can be formulated mathematically by utilizing MISOCP formulation as follows.

$$\text{minimize} \quad r \tag{3.6}$$

$$\text{subject to} \quad r + (1 - y_i)M \geq \|W - X\|,\ W \in E_i; i = 1, \ldots, n \tag{3.7}$$

$$cr + d \geq \|W - X\|,\ W \in E_i; i = 1, \ldots, n \tag{3.8}$$

$$r + (1 - z_j)M \geq \|C_j - X\| + R_j,\ j = 1, \ldots, m \tag{3.9}$$

$$cr + d \geq \|C_j - X\| + R_j,\ j = 1, \ldots, m \tag{3.10}$$

$$\sum_{i=1}^{n} \alpha_i y_i + \sum_{j=1}^{m} \beta_j z_j \geq P \tag{3.11}$$

$$y_i, z_j \in \{0, 1\}, r \geq 0 \tag{3.12}$$

In this formulation, objective function of 3.6 refers to minimizing of the maximum distance between facility and demand entities. Constraint 3.7 ensures the distance of all polygons of level 1 from facility should be less than objective function. Also,

next constraint 3.8 represents that the location of facility is in a way which all polygons would be covered in second level. Constraints 3.9 and 3.10 are defining same conditions for disks which are presented for polygons in equations 3.7 and 3.8 respectively. Constraint 3.11 ensures that weight of all polygons and disks located in first level should be greater than or equal to P and last constraint 3.12 imposes boundary conditions of the problem.

3.1.4 MISOCP formulation for k-level problem

Here, the k-level extension of the Minimax location problem for single facility is proposed. Suppose there are K levels in the problem, which are enumerated by index h. The formulation of the problem is as follows,

$$\text{minimize} \quad r \tag{3.13}$$

$$\text{subject to} \quad c_h r + d_h + (1 - y_{ih})M_h \geq \|W - X\|,\ W \in E_i; i = 1, \ldots, n$$

$$h = 1, \ldots, K \tag{3.14}$$

$$c_h r + d_h + (1 - z_{jh})M_h \geq \|C_j - X\| + R_j,\ j = 1, \ldots, m$$

$$h = 1, \ldots, K \tag{3.15}$$

$$\sum_{i=1}^{n} \alpha_i y_{ih} + \sum_{j=1}^{m} \beta_j z_{jh} \geq P_h,\ h = 1, \ldots, K \tag{3.16}$$

$$y_{ih} \leq y_{ih+1},\ h = 1, \ldots, K-1; i = 1, \ldots, n \tag{3.17}$$

$$z_{jh} \leq z_{jh+1},\ h = 1, \ldots, K-1; j = 1, \ldots, m \tag{3.18}$$

$$y_{iK} = 1,\ i = 1, \ldots, n \tag{3.19}$$

$$z_{jK} = 1,\ j = 1, \ldots, m \tag{3.20}$$

$$y_{ih}, z_{jh} \in \{0, 1\}, r \geq 0 \tag{3.21}$$

The constraints 3.14 and 3.15 ensure total coverage of demands which are positioned in level h for polygon regions and disks respectively. The M_h is a big value that holds constraints 3.14 and 3.15 be satisfied in the situation that demand i and j are covered in other levels. Constraint 3.16 represents that weight of all polygons and disks located in level h should be greater than or equal to P_h for each level h. Constraints 3.17

and 3.18 ensure the covered regions in previous levels, remain covered in next levels as well. The constraints 3.19 and 3.20 represent total coverage at the last level for all polygons and disk respectively. Finally, constraint 3.21 imposes binary and boundary conditions on decision variables.

3.1.5 Choosing the value of M_h

MISOCP formulation of the problem contains big-M values, namely M_h, for each level. Big-M formulations are usually weak and determining the best big-M value is very crucial, as the bigger the big-M value the weaker the formulation. Three approaches are proposed for obtaining the M_h, best big-M value in level h, for the single facility version of the problem. All of these three approaches utilize the bounds of covering radii (r) in order to find M_h. Hence, lower and upper bounds for covering radius should be specified.

Proposition 1: The optimal objective function value of the single level minimax problem is an upper bound for the covering range of the level one and a lower bound for the covering range of the level K.

$$c_1 r^* + d_1 \leq l^* \leq c_K r^* + d_K \tag{3.22}$$

Proof: The SOCP formulation for the single level minimax problem was presented in section 3.1.2. Consider l^* and X^* as optimal objective function and optimal location of the single level minimax problem. l^* equals the farthest distance between X^* and the farthest demand region from it. Similarly, considering X^* as a facility location and allocating the closest demand regions to this location one by one until reaching the covering percentage of the first level (P_1) forms first level of the location X^* for the k-level problem. Forming other levels in the same manner provides a feasible solution for the k-level problem. Since covering all of the demand regions in the first level is not required, it is clear that the covering range of the first level $c_1 r + d_1$ for the location X^* is less than equal to l^*. Furthermore, X^* and r are a feasible solution for the k-level problem and it could be claimed that optimal objective function value for

the k-level problem (r^*) is less than equal to r. Therefore we have,

$$c_1 r^* + d_1 \leq c_1 r + d_1 \leq l^* \tag{3.23}$$

On the other hand, in the last level (K), all regions should be covered completely. Accordingly, the covering range of the last level of the k-level problem is satisfying all constraints of the single level problem. Hence, the covering range of the last level ($c_K r^* + d_K$) of the k-level problem associated with its optimal location is a feasible solution for the single level problem which could not be less than l^*. As a result, the following equation is obtained:

$$l^* \leq c_K r^* + d_K \tag{3.24}$$

Approach 1

Considering the constraints 3.14 and 3.15, the problem contains M_h when demand regions are not covered in the hth level. In fact, M_h guarantees satisfying the constraints 3.14 and 3.15 where y_{ih} or z_{jh} is zero. Therefore we have,

$$\begin{aligned} c_h r^* + d_h + (1-0)M_h &\geq \|W - X\| \\ c_h r^* + d_h + (1-0)M_h &\geq \|C_j - X\| + R_j \end{aligned} \tag{3.25}$$

On the other hand, in the K-level Minimax location problem, the last level (K) covers all of the demand regions. Therefore the Euclidean distance of demand regions from the optimal facility location should be less than equal to the covering range of the last level.

$$\begin{aligned} c_K r^* + d_K &\geq \|W - X\| \\ c_K r^* + d_K &\geq \|C_j - X\| + R_j \end{aligned} \tag{3.26}$$

Considering the equations 3.25 and 3.26, it is clear that the constraints of the K-level problem which contains M_h are satisfied when the left hand side of the equations 3.25 be greater than equal to the left hand side of the equations 3.26.

$$c_h r^* + d_h + (1-0)M_h \geq c_K r^* + d_K \Rightarrow M_h \geq (c_K - c_h)r^* + (d_K - d_h) \tag{3.27}$$

Above equation provides an upper bound for M_h. However, since r^* is a decision variable, its value is not in hand before solving of the problem. We suggest an upper

bound for r^* to replace it in 3.27. Considering the first Proposition, an upper bound for r^* is as below.

$$c_1 r^* + d_1 \leq l^* \Rightarrow r^* \leq \frac{l^* - d_1}{c_1} \tag{3.28}$$

Replacing above obtained upper bound of r^* in the equation 3.27 would give the big-M value of the first approach ($M1_h$).

$$M1_h = (c_K - c_h)\frac{l^* - d_1}{c_1} + (d_K - d_h) \tag{3.29}$$

Approach 2

Considering the equations 3.14 and 3.15, M_h would appear in the constraints when y_{ih} or z_{jh} is equal to zero, then,

$$\begin{aligned} M_h &\geq \|W - X\| - (c_h r^* + d_h) \\ M_h &\geq \|C_j - X\| + R_j - (c_h r^* + d_h) \end{aligned} \tag{3.30}$$

Again, there is a decision variable r^* in the 3.30 and it is required to replace the right hand side of the equation by an equivalent term. Since a lower M_h is preferred, an upper bound of $\|W - X\|$ and a lower bound of r^* will next be found.

Proposition 2: The optimal location of the facility for k-level minimax problem lies inside of the convex hull of the regions' corners.

Proof: It can be proven by contradiction that optimal solution for k-level minimax location problem lies inside the convex hull of the demand regions. Assume an arbitrary point x outside of the convex hull as an optimal location for the problem. Moving in the direction of convex hull by Δ, gives another point which is closer to any arbitrary demand point in the convex hull than the assumed optimal location. This is a contradiction (See Figure 3.1). Based on the separating Hyperplane theorem, there exists a hyperplane such that x is in one side of the hyperplane and the convex hull of the demand regions is on the other side. A line containing x which is orthogonal to the separating hyperplane is called as separating axis. In Figure 3.1, it is shown that by shifting x on the separating axis towards the convex hull by a small Δ, we obtain another point x' that is closer to every point in the convex hull. Since the separating axis

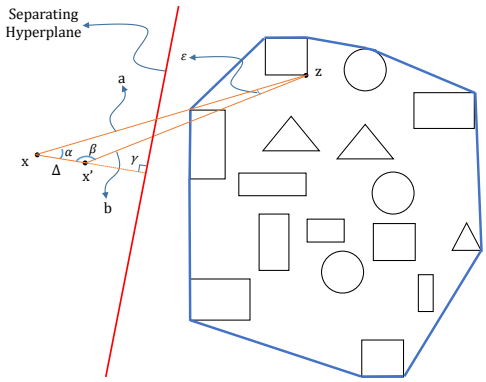


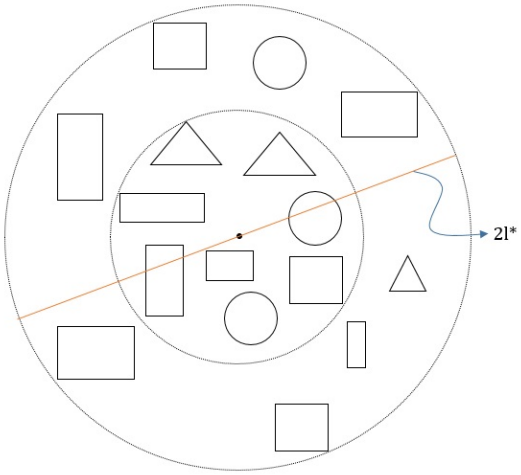
Figure 3.1: Proposition 2

is orthogonal to the separating hyperplane, γ is 90° and thereupon α is strictly less than 90°. Furthermore, considering the small value of Δ, the angle ϵ is very small. Based on the sine role in triangles, it can be shown that $a < b$ as follows,

$$\frac{a}{b} = \frac{sin(\beta)}{sin(\alpha)} = \frac{sin(180 - \alpha - \epsilon)}{sin(180 - \alpha)} \geq 1$$

The convex hull of the demand regions is a minimum convex set containing all of the regions. Also, considering Proposition 2, it can be concluded that the optimal location for the k-level minimax problem lies inside of the enclosing circle of the all demand regions.

Consequently, Euclidean distance between the optimal location of the facility and regions should be less than diameter of the circle with center of the optimal location and radius of l^*,



Therefore we have,

$$\|W - X\| \leq 2l^* \tag{3.31}$$
$$\|C_j - X\| + R_j \leq 2l^*$$

Additionally, considering the total coverage of all regions at the last level;

$$\|W - X\| \leq c_K r^* + d_K \tag{3.32}$$
$$\|C_j - X\| + R_j \leq c_K r^* + d_K$$

Replacing r^* by the obtained upper bound in equation 3.28 gives;

$$\|W - X\| \leq c_K(\frac{l^* - d_1}{c_1}) + d_K \tag{3.33}$$
$$\|C_j - X\| + R_j \leq c_K(\frac{l^* - d_1}{c_1}) + d_K$$

Consequently, an upper bound for the first term of the right hand side of the Equation 3.30 is $\min\left\{2l^*, c_K(\frac{l^* - d_1}{c_1}) + d_K)\right\}$.

For the second term of the right hand side of the Equation 3.30, a lower bound is required. Considering the first Proposition, a lower bound for r^* can be obtained as follows.

$$c_K r^* + d_K \geq l^* \Rightarrow r^* \geq \frac{l^* - d_K}{c_K} \tag{3.34}$$

Now, by replacing the above obtained bounds in Equation 3.30, we obtain another upper bound value for big-M ($M2_h$) as bellow:

$$M2_h = \max\left\{0, \min\left\{2l^*, c_K(\frac{l^* - d_1}{c_1}) + d_K\right\}\right\} - \max\left\{0, c_h(\frac{l^* - d_K}{c_K}) + d_h\right\} \tag{3.35}$$

Approach 3

In the third approach, a heuristic algorithm is proposed for determining an upper bound of the objective function value of the K-level problem. This new upper bound of r^* provides new values for $M1_h$ and $M2_h$.

By solving the single level single facility minimax location problem using the SOCP formulation, this approach first finds an initial location of the facility. Then, it assigns the nearest regions to the first level one by one until the covering percentage of the first

level is reached. Once the covering percentage of the first level is reached, regions are assigned to the second level. This procedure continues until all regions are assigned to levels. The critical distance of each level (r_h) is obtained as follows;

r_h = (*the farthest distance of the regions in level h from the initial location* − d_h)/c_h.

Based on Equations 3.13 and 3.14, in each level the distance between the facility location and farthest point of the regions of that level should be less than equal to $c_h r + d_h$. As a result, the above equation presents critical distance of each level for any arbitrary facility location. Maximum of critical distances for all levels provides the optimal covering level for that point. Since an arbitrary initial location is not necessarily the optimal location, this heuristic claims that an upper bound of r is equal to $\max\{r_h\}; h = 1, \ldots, K$.

$$r^* \leq \max_{h=1,\ldots,K}\{r_h\}. \tag{3.36}$$

The pseudocode of Approach 3 for finding an upper bound for M_h is given below where x represents the initial location.

Algorithm Heuristic for M

Begin
Solve the single level problem by the SOCP formulation to find the location x
Sort all regions based on the farthest distance to x
for $h = 1$ to K **do**
    Set $N = \varnothing$, $Wl_h = 0$, A = set of all demand regions.
    repeat
        Select nearest region (R) to the facility from A/N.
        $N = N \cup \{R\}$
        $Wl_h = Wl_h$+weight of R
        Set r_h = (farthest distance of the region R from the location $x - d_h$)/c_h
    until sum of weight of the regions in level h reaches to P_h
end for
$\max(r_h) \leftarrow$ Upper bound of r
End

Similar to the the second Approach, utilizing the upper bound of the r^* suggests a new upper bound for big-M ($M3_h$).

Based on the equations 3.34 and 3.36 we have,

$$\frac{l^* - d_k}{c_k} \leq r^* \leq \max(r_h) \tag{3.37}$$

Recalling the equation 3.30, M_h is as follows.

$$M_h \geq \|W - X\| - (c_h r^* + d_h)$$

Also, according to the equations 3.31 and 3.32, it is shown that

$$\|W - X\| \leq 2l^*$$

$$\|W - X\| \leq c_K r^* + d_K$$

Now, by utilizing the obtained new upper bound of r^* which is found in the equation 3.37 and replacing it in the above formulation, we obtain an upper bound for $\|W - X\|$ as follows.

$$\|W - X\| \leq \min(2l^*, c_K \max(r_h) + d_K) \tag{3.38}$$

On the other hand, it was shown in the first Proposition that $\frac{l^* - d_k}{c_k} \leq r^*$. Thus, we can conclude that

$$c_h r^* + d_h \geq c_h(\frac{l^* - d_k}{c_k}) + d_h \tag{3.39}$$

Consequently, the updated upper bound of the second Approach ($M'2_h$) is as below;

$$M'2_h = \max\{0, \min\{2l^*, c_K \max(r_h) + d_K\}\} - \max\left\{0, c_h(\frac{l^* - d_K}{c_K}) + d_h\right\} \tag{3.40}$$

Furthermore, the first Approach can also be updated by the new obtained upper bound of r^*. Given to the equation 3.27 we have;

$$M_h \geq (c_K - c_h)r^* + (d_K - d_h)$$

By replacing the $\max(r_h)$ as an upper bound of r^* in the above equation, we get the updated version of the upper bound of the first Approach ($M'1_h$).

$$M'1_h = (c_K - c_h)\max(r_h) + (d_K - d_h) \tag{3.41}$$

Finally, the obtained value of M_h in third Approach ($M3_h$) is the minimum of $M'1_h$ and $M'2_h$.

$$M3_h = \min\{M'1_h, M'2_h\} \tag{3.42}$$

Preliminary computational experiment: Comparison between 3 proposed approaches for big-M

The three approaches of finding the best big-M value in level h have been tested on several randomly generated instances. This computational experiment presents the obtained M_h values for each of these approaches and illustrates their effects on the computational time of the problem. Five instances have been considered in this experiment which are 100 demand regions and 5 levels, 144 demand regions and 4 levels, 196 demand regions and 3 levels, 225 demand regions and 5 levels and finally, 400 demand regions and 2 levels. More details on the other parameters of the model for solving instances will be explained in the Chapter 5. The 4th instance with 225 demand regions could not be solved in the time limit of the 1 hour and accordingly, the reported percent gap of the found solution during 1 hour has presented in the Table 3.1. For other instances, the big-M values of each instances as well as its associated computational time have been indicated in the Table 3.1.

In all of these instances the third approach has the lowest values for the M_h and

Table 3.1: Comparison of the proposed approaches for Big-M

Instances		Approaches	Big-M values					Time (seconds) or optimality gap (%)
			level 1	level 2	level 3	level 4	level 5	
1	(100,5)	M1	1739.87	1567.12	1396.07	727.39	0	1368.07
		M2	997.77	953.9	911.72	745.67	568.17	1150.57
		M3	429.6	385.73	343.55	177.5	0	1052.29
2	(144,4)	M1	1072.24	889.36	708.18	0	-	134.29
		M2	989.02	922.27	857.22	601.93	-	211.52
		M3	387.09	320.34	255.29	0	-	92.18
3	(196,3)	M1	346.28	172.29	0	-	-	36.57
		M2	562.71	453.64	346.28	-	-	15.25
		M3	216.43	107.36	0	-	-	14.11
4	(225,5)	M1	1856.17	1671.98	1489.49	776.2	0	45.61%
		M2	1064.62	1017.92	972.92	795.86	606.3	31.26%
		M3	458.31	411.62	366.62	189.55	0	22.17%
5	(400,2)	M1	189.06	0	-	-	-	19.65
		M2	334.49	189.06	-	-	-	54.36
		M3	145.43	0	-	-	-	12.79

consequently the lowest computational time as well. This table indicates that after the third approach, the first one is the best one with lowest M_h values. This table also shows the relation between M_h value and computational time which is a positive relation.

In order to get the smallest possible value of the M_h, the minimum of these three approaches' value is considered as M_h in our computational study.

$$M_h = \min\{M1_h, M2_h, M3_h\}$$

3.1.6 Time complexity of the single facility case

Time complexity of an algorithm refers to the total time required by that algorithm to complete its running procedure. It is calculated by counting the number of operations which a program executes during its running time. Time complexity of both discrete and continuous versions of the P-center problem is discussed in the literature. Single facility minimax problem could be stated as finding the location of a circle's center and radius which could cover all demand entities. In our problem, considering the different covering levels, the binary variables are utilized in the model. We are applying the mixed-integer SOCP algorithm for solving of the problem. The common algorithm in solving integer problems is using branch and bound ($B\&B$) technique. Time complexity in such problems depend on the complexity of its relaxation, number of integer variables and the applied algorithm for $B\&B$ technique. Accordingly, obtaining the complexity of the integer problems is not simple. Complexity of the SOCP formulation could be achieved by $O(a^2 \sum_{i=1}^{A} a_i)$ where a is the total number of decision variables in the problem, A is the number of constraints and a_i refers to dimension of the each constraint [34].

In this specific problem, the integer variables are used in the formulation to assign demand entities to levels. If the assignment of demand regions to the levels be known, the problem could be reformulated in normal SOCP version which has above mentioned complexity. The number of possible allocations of demand entities to levels could be obtained by applying Stirling numbers of the second kind formula. This formula determines total number of ways to partition a set of A elements into B

nonempty subsets [1]. Stirling numbers of the second kind is indicated by S(A,B).

$$S(A,B) = \frac{1}{B!}\sum_{i=0}^{B}(-1)^i \cdot \binom{B}{i} \cdot (B-i)^A \tag{3.43}$$

In our problem, there is a set of $(n+m)$ demand regions which should be partitioned into K levels. Consequently, the worst case time complexity to search all solutions is the multiplication of SOCP complexity and total number of possible allocations which is,

$$O(K^3(n+m)^2(m+n+L)) \times \frac{1}{K!}\sum_{i=0}^{K}(-1)^i \cdot \binom{K}{i} \cdot (K-i)^{n+m} \tag{3.44}$$

Where K, n, m, L are number of levels, polygons, disks and corners of polygons respectively.

3.2 Mathematical formulation of the multi-facility case

In this section, formulation of the Multi-Level Continuous Minimax Problem (ML-CMP) is extended to the multi-facility case. Here, besides all earlier mentioned constraints for the single facility version, new constraints for assigning demand regions to the facilities are added to the formulation.

3.2.1 MISOCP formulation of the multi-facility, k-level problem

Allocating of demand entities to the facilities is done by binary variables. Hence, the new index t, which enumerates facilities, is added to the model. Let y_{ith} be a binary variable which equals 1 (0) if polygon i is covered by facility t in level h (other levels) and similarly let z_{jth} be a binary variable which equals 1 (0) if disk j is covered by facility t in level h (other levels). Also, r_t represents the covering radii of the facility t.

Now, the MISOCP model for the multi-facility, k-level problem can be stated as fol-

lows,

$$\text{minimize} \quad R \tag{3.45}$$

subject to:

$$R \geq r_t,\ t = 1, \ldots, T \tag{3.46}$$

$$c_h r_t + d_h + (1 - y_{ith}) M_{ith} \geq \|W - X_t\|,\ i = 1, \ldots, n; W \in E_i; h = 1, \ldots, K$$
$$t = 1, \ldots, T \tag{3.47}$$

$$c_h r_t + d_h + (1 - z_{jth}) M_{jth} \geq \|C_j - X_t\| + R_j,\ j = 1, \ldots, m; h = 1, \ldots, K$$
$$t = 1, \ldots, T \tag{3.48}$$

$$\sum_{i=1}^{n} \sum_{t=1}^{T} \alpha_i y_{ith} + \sum_{j=1}^{m} \sum_{t=1}^{T} \beta_j z_{jth} \geq P_h,\ h = 1, \ldots, K \tag{3.49}$$

$$\sum_{t=1}^{T} y_{ith} \leq 1,\ i = 1, \ldots, n; h = 1, \ldots, K - 1 \tag{3.50}$$

$$\sum_{t=1}^{T} z_{jth} \leq 1,\ j = 1, \ldots, m; h = 1, \ldots, K - 1 \tag{3.51}$$

$$y_{ith} \leq y_{ith+1},\ h = 1, \ldots, K - 1; i = 1, \ldots, n; t = 1, \ldots, T \tag{3.52}$$

$$z_{jth} \leq z_{jth+1},\ h = 1, \ldots, K - 1; j = 1, \ldots, m; t = 1, \ldots, T \tag{3.53}$$

$$\sum_{t=1}^{T} y_{itK} = 1,\ i = 1, \ldots, n \tag{3.54}$$

$$\sum_{t=1}^{T} z_{jtK} = 1,\ j = 1, \ldots, m \tag{3.55}$$

$$r_t \geq 0,\ t = 1, \ldots, T \tag{3.56}$$

$$y_{ith}, z_{jth} \in \{0, 1\} \tag{3.57}$$

The maximum value between covering radius of all facilities is the objective function value for this problem and constraint 3.46 represents such an equation. Constraint 3.47 ensures that all polygon regions should be covered completely inside the corresponding level by the associated facility. Constraint 3.48 extends the constraint 3.47 to the disk regions as well. Equation 3.49 guarantees covering of the predetermined percentage of all regions in each level. All regions should be covered by the facilities and each of these regions should be covered by exactly one facility at least in the last covering level of the associated facility. Constraints 3.54 and 3.55 make sure that each facility is covered by exactly one facility at the last covering level of that facility.

Constraints 3.50 and 3.51 ensure that each region can not be covered by more than 1 facility in other levels as well. Constraints 3.52 and 3.53 indicate that the covered regions in previous levels, also remain covered in the next levels for both polygon and disk regions. Finally, constraints 3.56 and 3.57 impose binary and boundary conditions on the decision variables.

Since constraints 3.54 and 3.55 represent that regions are covered by exactly one facility in the last level and constraints 3.52 and 3.53 guaranty that the binary variable value of the previous levels can not be more than value of the last level, it can be shown that the regions can not be covered by more than one facility in the last level. Therefore, the constraints 3.50 and 3.51 are redundant and have been written in the formulation to explain the model clearly.

Similar to the single facility case, this formulation also contains Big-M values and determining the best big-M value is crucial. According to the constraints 3.47 and 3.48, M_{ith} and M_{jth} appear in the formulation when y_{ith} or z_{jth} is zero. In such a situation, the constraints 3.47 and 3.48 can be written as below.

$$M_{ith} \geq \|W - X_t\| - (c_h r_t + d_h) \tag{3.58}$$
$$M_{jth} \geq \|C_j - X_t\| + R_j - (c_h r_t + d_h)$$

Since a lower big-M value is preferred, an upper bound for $\|W - X_t\|$ and $\|C_j - X_t\| + R_j$ and a lower bound for $c_h r_t + d_h$ is required. In the multi-facility version of the problem, it is possible that a facility does not cover any demand region. Hence, the optimal covering radius for a facility r_t could be zero. As a result, the lower bound for $c_h r_t + d_h$ is d_h.

We explained by Proposition 2 in the second approach of choosing the best big-M value for the single facility case that the optimal location of the facility should lie inside the convex hull of the regions' corners. We can extend that proposition for multi-facility case with the same proof as well. Therefore, it can be claimed that the farthest distance between any region and any facility is less than equal to the distance between that region and enclosing circle of all demand regions. Consequently, we consider the farthest distance between each demand region and the enclosing circle of all demand regions as an upper bound for $\|W - X_t\|$ and $\|C_j - X_t\| + R_j$. Suppose the farthest distance of a polygon region i and a disk region j from the enclosing circle of

regions be D_i and D_j, respectively. Then, the best big-M value for the multi-facility case of the problem is as follows.

$$M_{ith} = D_i - d_h,\ i = 1, \ldots, n; h = 1, \ldots, K; t = 1, \ldots, T \tag{3.59}$$
$$M_{jth} = D_j - d_h,\ j = 1, \ldots, m; h = 1, \ldots, K; t = 1, \ldots, T$$

CHAPTER 4

HEURISTIC SOLUTION APPROACHES

MISOCP formulation of the problem contains big-M values which make it weak. Therefore this formulation is unable to solve large instances to optimality. During its solution by branch-and-bound (B&B) technique, the B&B tree becomes too large and out of memory error appears. Consequently, we propose several heuristics in this chapter to be able to solve large-size instances in reasonable time. Heuristics provide an alternative approach to overcoming the limitations of exact methods. In contrast to an exact method, a heuristic is a general approach that solves a problem in reasonable time with no guarantee on the optimality of the solution.

4.1 Heuristics for the single facility case

In this chapter six heuristics, five of which are SOCP based, are proposed for the single facility case of the problem. In the first five heuristics, an SOCP problem is solved in each iteration. Since SOCP problems are solvable in polynomial time, computational times of the heuristics are reasonable when the number of iterations is not too many. The last heuristic abstracts the search region in each iteration until it approaches to an acceptable answer. These heuristics are explained in the following sections in detail.

4.1.1 Critical Level Heuristic (CLH)

The main idea for the next 5 heuristics are based on the Location-Allocation (LA) algorithm. This method has been used in different multi-facility location problems in the literature [24]. Procedure of LA is as finding some locations for facilities and allocating all demand entities to these facilities (allocation step) and then finding new facility locations keeping the allocations the same (location step). Allocation and location steps alternate until convergence is achieved. In our heuristics, we begin at a location and determine the corresponding levels (leveling step) for that location. Next, based on these levels new facility location is found and the algorithm iterates between location and leveling steps until convergence or some other stopping conditions are reached.

Leveling Step:

In this step, firstly, nearest demand regions to the in hand location are determined. Then, these demand regions are added to the first level one by one until sum of weights of the added regions reaches the corresponding coverage percentage of the first level. The selected regions form the first level and in this manner the outer levels are formed as well.

Critical Level:

Critical level refers to the level that specifies the objective function of the problem. All levels have particular coverage domain which is depended on its parameters (c_h and d_h) but the important factor is covering radius of that level (r_h). This radius could be obtained by the following formula and the maximum of these radii of all levels determines the objective function of the problem.

x : Current location of the facility

f_{hx} : Distance of the farthest point of the demand regions in level h from x

$$r_h = (f_{hx} - d_h)/c_h$$

$$r = \max(r_h)$$

$$\textit{critical level or levels} = \arg \max_{h=1,\ldots,K} (r_h)$$

The main part of the SOCP based algorithms is related to updating the facility location. In the critical level heuristic (CLH), after starting the algorithm from an initial location, leveling step is performed. Then, critical level or levels for that location are determined. If several critical levels exist in an iteration, one of them is selected randomly. Next, all regions which are not located in the chosen critical level are ignored and the problem is reduced to a single level single facility problem. An SOCP problem is solved and the new facility location for the regions in the selected critical level is found. This location is taken as the initial point for the next iteration. The pseudocode of the CLH heuristic is given in Algorithm CLH.

Algorithm CLH

Begin
Consider a random point x_0 inside the convex hull of the demand regions as the initial facility location.
while a stopping condition is reached **do**
    Sort all regions based on the farthest distance to x_0
    for $h = 1$ to K **do**
        Set $N = \varnothing$, $Wl_h = 0$, A = set of all demand regions.
        repeat
            Select nearest region (R) to the facility x_0 from A/N.
            $N = N \cup \{R\}$
            $Wl_h = Wl_h$+weight of R
        until sum of the weights of the regions in level h $\geq P_h$
        Set $r_h = (f_{hx} - d_h)/c_h$
    end for
    Set $r^* = \max(r_h)$.
    Set critical level or levels = $\arg\max_{h=1,\ldots,K}(r_h)$
    if several critical levels exist **then**
        Select one of them randomly.
    end if
    Solve an SOCP problem for the demand regions in the critical level to get a new facility location x_0
end while
End

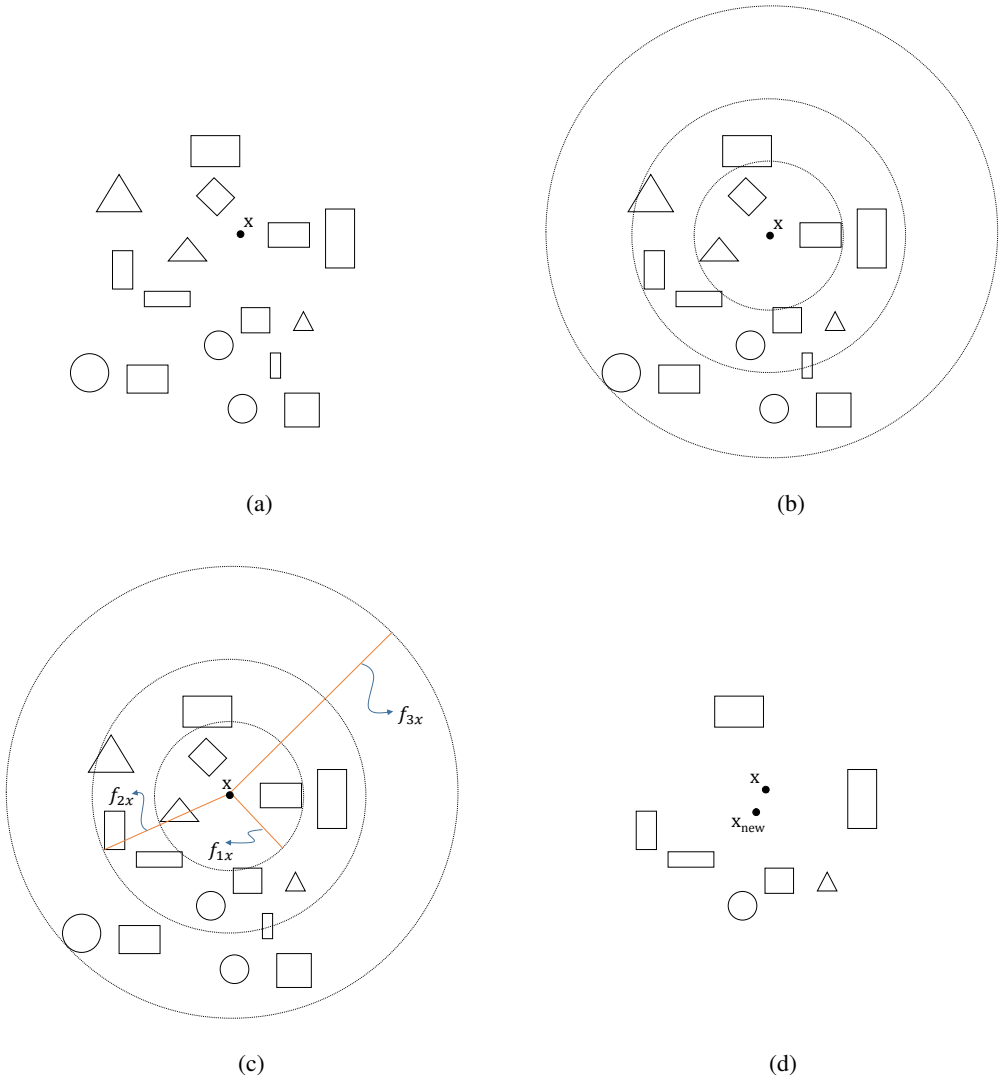


Figure 4.1: The steps of the CLH heuristic

Line 2 is initialization of the algorithm which is done by randomly selecting a point from the convex hull of the demand regions. In lines 4 to 13 leveling step is done. Lines 14 to 18 determine the critical level. In line 19, an SOCP model is solved for the regions inside critical level to update the facility location.

There are two stopping criteria for the CLH heuristic. The first one is maximum number of iterations and the second one is convergence. Getting three successive close solutions is considered as the criterion for the convergence.

Each iteration (starting at line 3 and coming back to line 3) of this algorithm is solvable in polynomial time. The CLH heuristic gets stuck quickly. Getting the same

facility location in the location step provides the same levels and as a result the same locations for the next iterations. Hence, the convergence condition is activated quickly before searching the solution space sufficiently. Furthermore, the performance of this algorithm depends on the initial location. Therefore, the algorithm can be run with several initial locations (replications) with the hope of obtaining better solutions. Starting from good initial facility locations results in better solutions. One good initial facility location for the algorithm could be the SOCP solution for the single level single facility minimax problem. The Figure 4.1 displays steps of the CLH heuristic. Demand regions and the random initial location for a facility is shown in Figure (a). Leveling step for the considered facility location is presented in Figure (b). Next figure (c), illustrates the determination procedure of the critical level. Finally, Figure (d) displays the demand regions which are located inside critical level. An SOCP problem is solved for these regions to update facility location. The new facility location is shown in the Figure (d) as well.

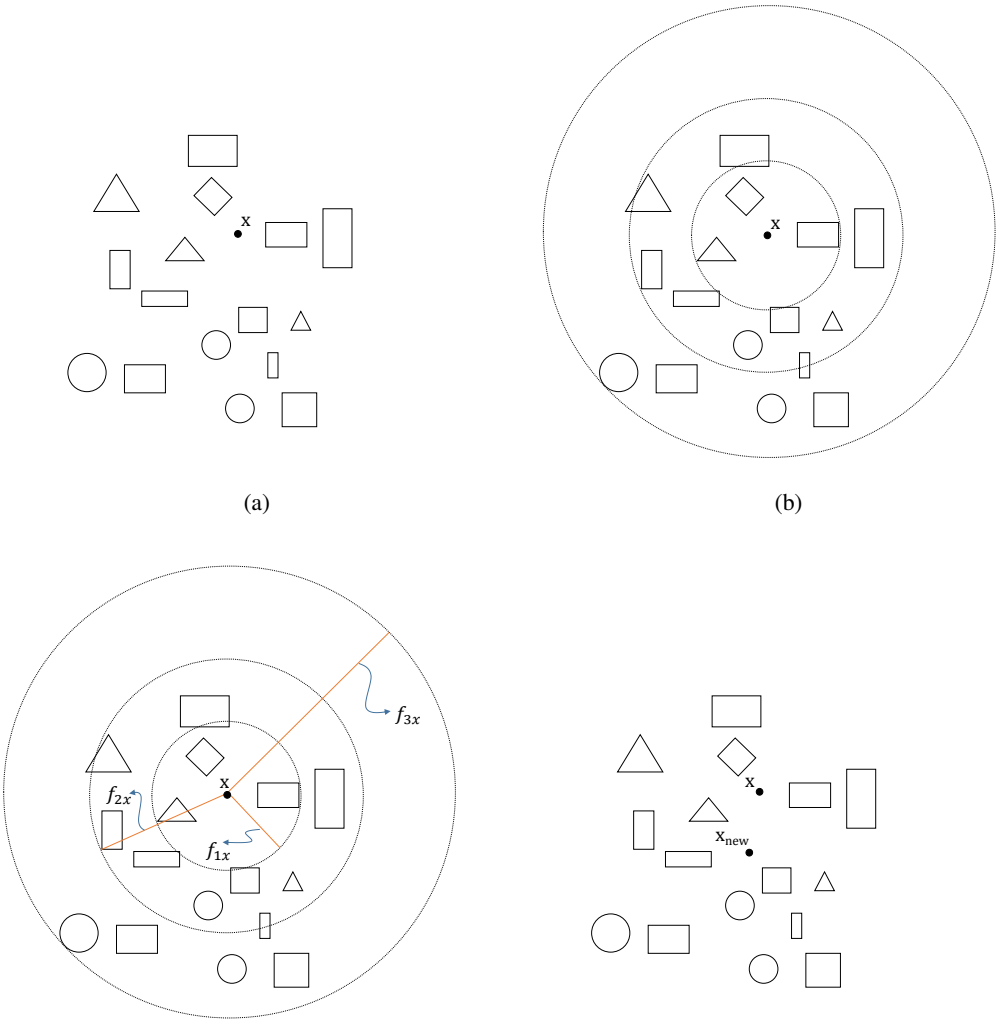
4.1.2 Location Leveling Heuristic (LLH)

The next proposed heuristic is the Location Leveling Heuristic (LLH). The LLH works similar to the CLH but considers all the demand regions in the location step. The CLH finds the new location by solving an SOCP problem for the demand regions located in the critical level. The LLH, similar to the CLH, begins at an initial location and performs the leveling step for that initial location. After the leveling step, demand regions are allocated to their corresponding levels and consequently the values of the binary variables (y_{ih} and z_{jh}) are determined. The binary variables are fixed with the obtained values and the MISOCP problem becomes as an SOCP problem which is solved to obtain the new location for the next iteration. The pseudocode of the LLH heuristic is given in Algorithm LLH.

Algorithm LLH

Begin
Consider a random point x_0 inside the convex hull of the demand regions as the initial facility location.
while a stopping condition is reached **do**
    Sort all regions based on the farthest distance to x_0
    for $h = 1$ to K **do**
        Set $N = \varnothing$, $Wl_h = 0$, A = set of all demand regions.
        repeat
            Select nearest region R to the facility x_0 from A/N.
            $N = N \cup \{R\}$
            $Wl_h = Wl_h$+weight of R
        until sum of weights of the regions in level h $\geq P_h$
        Set $r_h = (f_{hx} - d_h)/c_h$
    end for
    Set $r^* = \max(r_h)$.
    Get values of the binary variables based on the leveling step.
    Solve an SOCP problem for the multi-level minimax problem and get the new location x_0
end while
End

In this heuristic after some iterations, it is possible to encounter with a situation where allocation of the demand regions to the levels are the same as the previous iterations. Hence, we will obtain the same location as in the previous iterations and get stuck. However, LLH mostly gets stuck later than CLH considering the fact that the possibility of having the same allocations for all demand regions is less than having the same allocations for just demand regions located in the critical level. Again, in the LLH the stopping criteria are both reaching the maximum number of iterations and achieving convergence which is getting three successive close enough solutions. Steps of the LLH heuristic are illustrated in the Figure 4.2. Demand regions and the initial facility location is shown in Figure (a). Covering levels of the considered facility location is presented in Figure (b). Figure (c), displays the procedure of calculating objective



(a) (b)

(c) (d)

Figure 4.2: The steps of the LLH heuristic

function value for the considered facility location. After determining of the values of the binary variables, an SOCP problem is solved for the problem and the new location is found for the facility which is shown in Figure (d).

4.1.3 Critical Level and Demand Exchange Heuristic (CLDEH)

This heuristic is also based on the location-allocation algorithm. Here again we iterate between the location-leveling steps. The Critical Level and Demand Exchange Heuristic (CLDEH) is introduced to cope with the disadvantages of the first algorithm (CLH). The main weakness of the CLH heuristic is getting stuck. This algorithm performs the initialization and leveling steps as done in the CLH. However, the convergence condition in this algorithm is eliminated to let the heuristic iterate and explore new solutions. After finding a local solution, the CLDEH exchanges some demand regions of the critical level by demand regions which are located in other levels to explore new areas of the solution space.

Exchanging problem elements is a famous approach for getting out of stuck in heuristics. One of the first applications of the elements exchange was in the Traveling Salesman Problem (TSP). 2-opt exchange heuristic for TSP is proposed in 1958 by Croes [14]. He developed a method that by considering an initial solution eliminates two edges or nodes and inserts them in other place and reconnects the nodes to get a new solution. This idea then was extended to location-allocation problems as well. Cooper [13] in 1964 proposed a heuristic which was a general scheme for all location-allocation problems for finding new improving solutions after getting stuck in the local optima. This heuristic suggests alternating problem elements in order to obtain another good solution. Additionally, Teitz and Bart [49] in 1968 introduced a special exchange or swap heuristic for the discrete Minisum problem. We are using such an idea in this thesis.

Demand region exchanging step is performed by utilizing Simulated Annealing (SA) metaheuristic. We select some regions randomly to remove from critical level and add some other random demand regions form other levels until satisfying the associated covering percentage of the critical level. If the obtained new solution is better than the incumbent (current) solution, this new location is accepted and will be the initial

location for the next iteration. Otherwise, with a specific probability, non-improving solution will be accepted. This specific probability is decreased iteration by iteration with a temperature reduction function to reduce the probability of accepting non-improving solutions in the next iterations. The pseudocode of the CLDEH heuristic is given in algorithm CLDEH.

Algorithm CLDEH

Begin
Consider an initial random point x_0 inside the convex hull of the demand regions.
Consider an initial temperature t and a temperature reduction function α
Perform the leveling step for x_0 and evaluate objective function value $f(x_0)$ at x_0.
while a stopping condition is reached **do**
    while maximum number of iterations for SA procedure is reached **do**
        Remove one demand region randomly from the critical level and add demand regions until reaching the covering percentage of the critical level.
        solve SOCP for the critical level to get new location x
        Sort all regions based on the farthest distance to x
        for $h = 1$ to K **do**
            Perform leveling step.
            Set $r_h = (f_{hx} - d_h)/c_h$
        end for
        Set $f(x) = \max(r_h)$
        Set critical level or levels = $\arg\max_{h=1,\ldots,K}(r_h)$
        if several critical levels exist **then**
            Select one of them randomly.
        end if
        if $f(x) < f(x_0)$ **then**
            Set $x_0 = x$
        end if
        if $f(x) \geq f(x_0)$ **then**
            Generate a random number b in range $(0, 1)$
            if $b < \exp((f(x_0) - f(x))/t_0)$ **then**
                Set $x_0 = x$
            end if
        end if
    end while
    $t = \alpha(t)$
end while
End

Lines 1 to 4 initialize the algorithm with some starting parameters. In lines 6 to 28 simulated annealing algorithm is performed. Firstly, a new neighborhood solution for the initial location is obtained by exchanging demand regions. Then, in lines 8 to 18 corresponding objective function and critical level for the new solution is found. Condition for acceptance of this solution is listed in lines 19 to 27. This procedure is repeated for a specific number of iterations. In line 29 the initial temperature is decreased to reduce the possibility of accepting non-improving solutions in the next iterations of the SA procedure. Simulated annealing lines (6 to 28) are repeated until a stopping condition is reached.

As the initial temperature approaches zero, the probability of accepting non-improving solutions will approach zero too. Thus, one of the stopping criteria is terminating replications when t becomes small. The second stopping criterion is considering a maximum number of replications for the SA algorithm. This heuristic also depends on the initial location, therefore, the solution of single level minimax problem for all demand entities could be one good starting point for the algorithm. Figure 4.3 illustrates the steps of the CLDEH heuristic as well. In the Figures (a) and (b) the initial random location of the facility x_0 and its corresponding covering levels are shown. The Figure (c) determines the critical level for the x_0. Suppose that the second level is the critical level for the x_0. Hence, the removed region should be from the second level. Hence, the located regions in the second level are displayed in Figure (d). The dashed region in the Figure (d) is removed and several regions are added for the critical level which is shown in Figure (e). Considering the current regions of the second level, an SOCP problem is solved for the second level and the new location is found for the facility which is shown in Figure (e).

Two versions for CLDEH is considered in this thesis. One of them is removing 1 demand region in the exchanging procedure and the second one is removing 2 demand regions. A preliminary computational experiment is done to select one of these versions for the rest of thesis.

Six randomly generated instances are considered for this comparison and solved with two versions of CLDEH for 20 different initial locations (replications). Average and minimum objective function values and computation times over 20 replications are

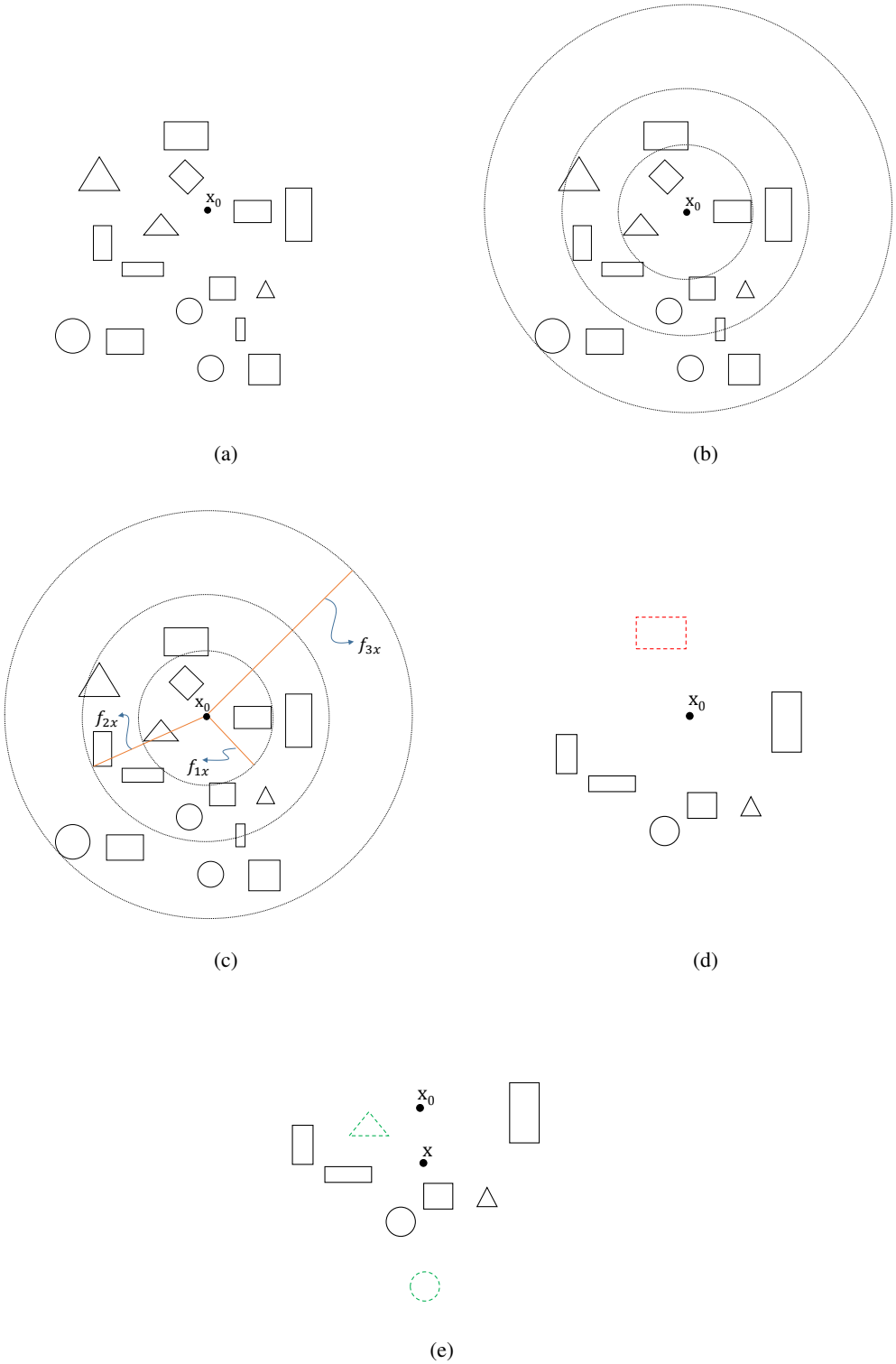


Figure 4.3: The steps of the CLDEH heuristic

Table 4.1: Comparison between the CLDEH versions

Number of demand regions	Number of levels	CLDEH version 1				CLDEH version 2			
		Average		Min		Average		Min	
		Obj. value	Running time (s)	Obj. value	Running time (s)	Obj. value	Running time (s)	Obj. value	Running time (s)
100	3	377.48	3.74	376.98	3.67	377.62	4.17	376.02	3.84
	4	368.49	4.21	364.52	4.04	369.35	4.67	364.52	4.18
196	3	355.88	5.6	355.88	5.49	355.88	6.15	355.88	5.71
	4	341.92	11.19	339.47	10.94	342.29	11.3	339.47	10.6
400	3	387.27	13.72	387.27	13.54	387.27	13.86	387.27	13.13
	4	360.86	23.74	358.53	23.02	361.09	23.4	359.16	22.09

found and displayed in Table 4.1. Computational time for both versions are close to each other. The objective function value for the first version in some instances is better than that for the second version. Briefly, minimum values are the same but the average values are slightly better for the first version. Hence, the first version which is removing one demand region is considered as the final version of CLDEH.

4.1.4 Critical Level, Upper Demand Exchange Heuristic (CLUDEH)

In exchanging the demand regions in the previous heuristic (CLDEH), random selection is utilized. However, choosing a proper demand region as an added entity to the critical level could affect the quality of the heuristic solution. Therefore, a new heuristic is proposed for the problem. The Critical Level Upper Demand Exchange Heuristic (CLUDEH) suggests selecting an entering demand region from one upper level. Exploitation of the good solutions and exploration of the new promising regions are two main features of a perfect heuristic. The CLDEH heuristic allows the algorithm to insert demand regions from any outer level which provides exploration feature for the heuristic. Forcing the algorithm to insert demand regions from one upper level helps in exploiting of the good solutions. Hence, the CLUDEH heuristic is introduced. The CLUDEH also uses simulated annealing in exchanging demand regions for getting neighborhood solutions. The pseudocode of the CLUDEH heuristic is given in algorithm CLUDEH.

Algorithm CLUDEH

Begin
Consider an initial random point x_0 inside the convex hull of the demand regions.
Consider an initial temperature t and a temperature reduction function α.
Perform the leveling step for x_0 and evaluate objective function value $f(x_0)$.
while a stopping condition is reached **do**
    while maximum number of iterations for SA procedure is reached **do**
        Remove one demand region randomly from the critical level.
        if Critical level not be the last level **then**
            Add demands from an upper level to the critical level until reaching its covering percentage.
        else
            Add demands from a random level to the critical level until reaching its covering percentage.
        end if
        solve SOCP for the critical level to get new location x
        Sort all regions based on the farthest distance to x
        for $h = 1$ to K **do**
            Perform leveling step.
            Set $r_h = (f_{hx} - d_h)/c_h$
        end for
        Set $f(x) = \max(r_h)$
        Set critical level or levels = $\arg\max_{h=1,\ldots,K}(r_h)$
        if several critical levels exist **then**
            Select one of them randomly.
        end if
        if $f(x) < f(x_0)$ **then**
            Set $x_0 = x$
        end if
        if $f(x) \geq f(x_0)$ **then**
            Generate a random number b in range (0, 1)
            if $b < \exp((f(x_0) - f(x))/t_0)$ **then**
                Set $x_0 = x$
            end if
        end if
    end while
    $t = \alpha(t)$
end while
End

Table 4.2: Comparison between the CLUDEH versions

Number of demand regions	Number of levels	CLUDEH version 1				CLUDEH version 2			
		Average		Min		Average		Min	
		Obj. value	Running time (s)	Obj. value	Running time (s)	Obj. value	Running time (s)	Obj. value	Running time (s)
100	3	386.38	4.26	376.98	4.09	386.67	4.17	376.98	4.08
	4	374.61	4.48	367.66	4.35	377.89	4.33	367.66	4.17
196	3	355.88	5.87	355.88	4.82	355.88	5.79	355.88	5.72
	4	341.9	11.75	337.34	11.46	342.41	11.28	339.47	11.04
400	3	387.27	12.97	387.27	9.8	387.27	12.42	387.27	7.82
	4	365.95	24.43	358.68	23.87	364.15	23.59	358.19	23.26

Stopping conditions for this algorithm is again maximum number of iterations as well as convergence condition for the simulated annealing procedure. Convergence criterion in SA is cooling the system by decreasing the temperature until near zero. Initial location affects the performance of the heuristic therefore, CLUDEH can be replicated with several different initial locations.

Similar to the CLDEH heuristic, the Figure 4.4 represents the steps of the CLUDEH heuristic. The main difference between the Figures 4.3 and 4.4 is in the located level of the inserted demand regions to the critical level. In the Figure (e) of the 4.4, it is shown that the inserted demand regions (dashed regions) are from the third level which is an upper level of the critical level (second level).

One idea for improving this heuristic can be removing 2 demand regions instead of one. We have tested this extension for CLUDEH to select the most efficient version for CLUDEH. In Table 4.2, the computational results for comparing these two versions are provided. Again instances with 100, 196, and 400 demand regions and 3 and 4 levels are generated randomly. These six randomly generated instances are solved for 20 different initial locations with both versions of CLUDEH. The computational results for both versions are similar to each other but for some instances the first version performs better. As a result, the first version of CLUDEH is considered as the final version in the rest of this thesis.

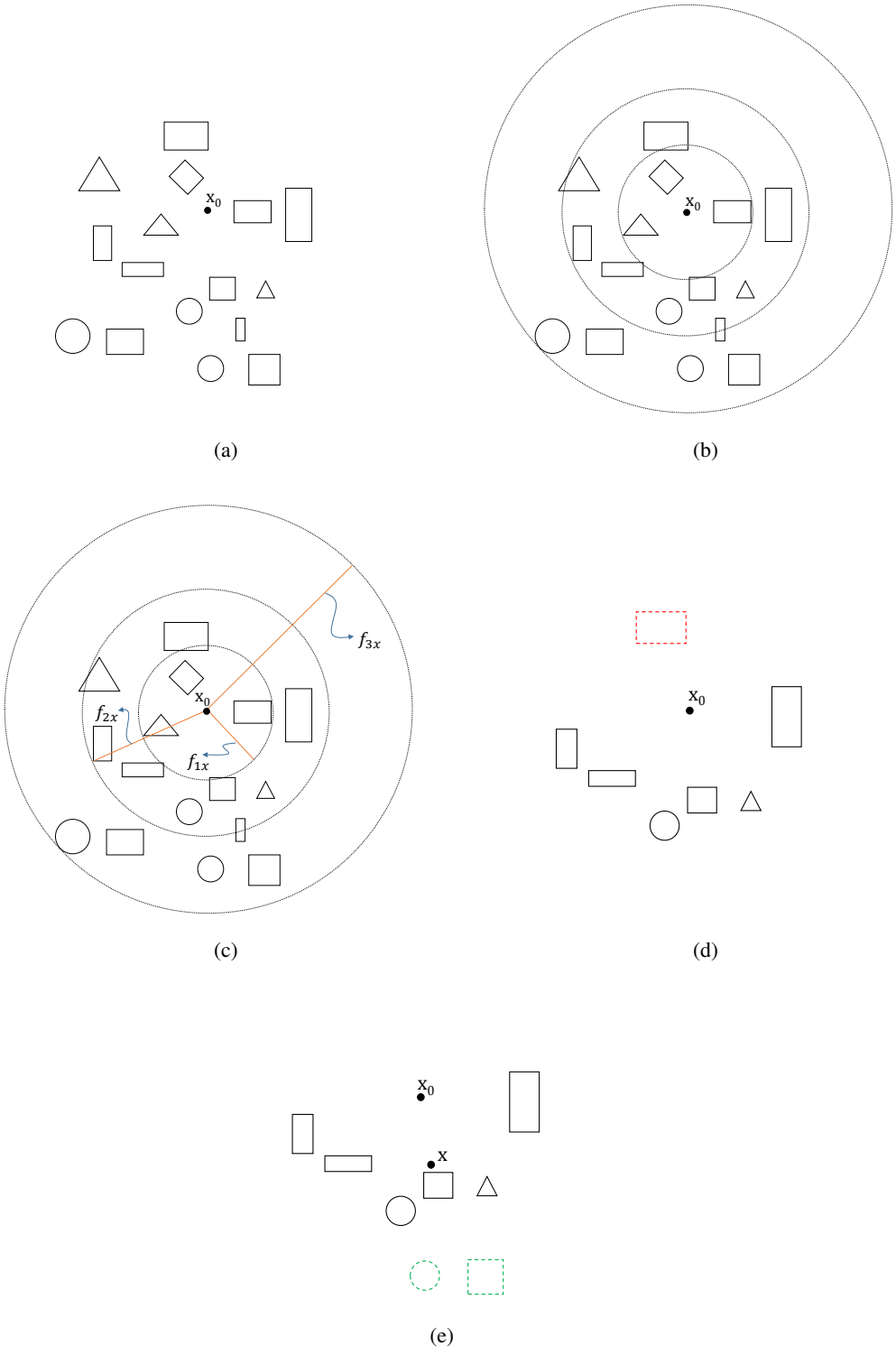


Figure 4.4: The steps of the CLUDEH heuristic

4.1.5 Descent Critical Level, Demand Exchange Heuristic (DCLDEH)

Unlike the two previous heuristics, we introduce a new heuristic which does not accept any non-improving solutions. This heuristic is similar to the steepest descent algorithm. The steepest decent algorithm evaluates all solutions to find the best improving one. However, this heuristic accepts the first improving solution. Descent Critical Level Demand Exchange Heuristic (DCLDEH), firstly, selects a random point as the initial location of the heuristic and then based on this point, it performs the leveling step to determine the critical level. The demand regions exchange procedure is applied in the DCLDEH as well. However, here the algorithm just moves to the improving neighborhood locations. The pseudocode of DCLDEH heuristic is given in algorithm DCLDEH.

Lines 1 and 2 initialize the heuristic with starting point of x_0. Line 4 enumerates the iterations of the algorithm. In the lines 5 to 9 the covering levels for each initial location are formed. Additionally, the corresponding objective function and critical level for the x_0 is determined in lines 10 to 14. Lines 15 to 18 represent the demand exchange procedure for getting out of stuck and lines 19 to 23 force the heuristic to move to the just improving solutions. Finally line 24 finds the new initial location by solving an SOCP formulation for the next iteration. This heuristic does not increase the index of the iteration until getting an improving solution. Therefore, another parameter is define to enumerate number of cycles inside the heuristic. The maximum number of cycles is one of the stopping conditions for the problem. The second stopping criterion is the convergence condition which is defined as getting three successive close enough solutions.

For selecting the entering demand regions in the exchange procedure, two approaches were presented in the previous heuristics. CLDEH suggests random selection while CLUDEH chooses the input demand regions from one upper level. In both of these two approaches, two versions were proposed. Consequently, 4 versions can be considered for exchanging demand regions in the current heuristic. These version are as follows.

Algorithm DCLDEH

Begin
Consider an initial random point x_0 inside the convex hull of the demand regions.
while a stopping condition is reached **do**
    for $iter = 1$ to $Niter$ **do**
        Sort all regions based on the farthest distance to x_0
        for $h = 1$ to K **do**
            Perform leveling step.
            Set $r_h = (f_{hx} - d_h)/c_h$
        end for
        Set $r^*_{iter} = \max(r_h)$
        Set critical level or levels = $\arg\max_{h=1,\ldots,K}(r_h)$
        if several critical levels exist **then**
            Select one of them randomly.
        end if
        if $r^*_{iter} = r^*_{iter-1}$ **then**
            Remove one demand region randomly from the critical level.
            Add demand regions until reaching the covering percentage of the critical level.
        end if
        if $r^*_{iter} > r^*_{iter-1}$ **then**
            Set $iter = iter - 1$
            Remove one demand region randomly from the critical level.
            Add demand regions until reaching the covering percentage of the critical level.
        end if
        solve SOCP for the critical level to get new location x_0
    end for
end while
End

1. Removing 1 demand region from the critical level and adding others from a randomly selected level.

2. Removing 2 demand regions from the critical level and adding others from a randomly selected level.

3. Removing 1 demand region from the critical level and adding others from an

Table 4.3: Comparison between the DCLDEH versions

Number of demand regions	Number of levels	version 1		version 2		version 3		version 4	
		Min objective value	Min running time (s)	Min objective value	Min running time (s)	Min objective value	Min running time (s)	Min objective value	Min running time (s)
100	3	378.66	4.42	379.35	4.77	386.26	4.32	385.06	4.12
	4	368.91	4.22	369.35	4.19	386.18	2.74	376.05	4.13
196	3	355.88	0.57	355.88	0.84	355.88	0.72	355.88	0.6
	4	343.62	5.85	342.07	6.21	347.42	4.72	342.03	5.76
400	3	387.27	1.65	387.27	1.7	387.27	1.67	387.27	1.76
	4	360.68	13.34	361.23	14.32	369.25	10.84	364.91	13.28

upper level.

4. Removing 2 demand regions from the critical level and adding others from an upper level.

A preliminary computational experiment is done to select one of these versions for the rest of thesis. Six randomly generated instances are considered for this comparison and solved with four versions of DCLDEH for 20 different initial locations (replications). The minimum objective function values and the computation times over 20 replications are obtained and displayed in the Table 4.3. Computational time for all versions are close to each other but the objective function values for the first version in some instances are better than others. Hence, the first version is selected as the final version of DCLDEH.

4.1.6 Area Abstraction Heuristic (AAH)

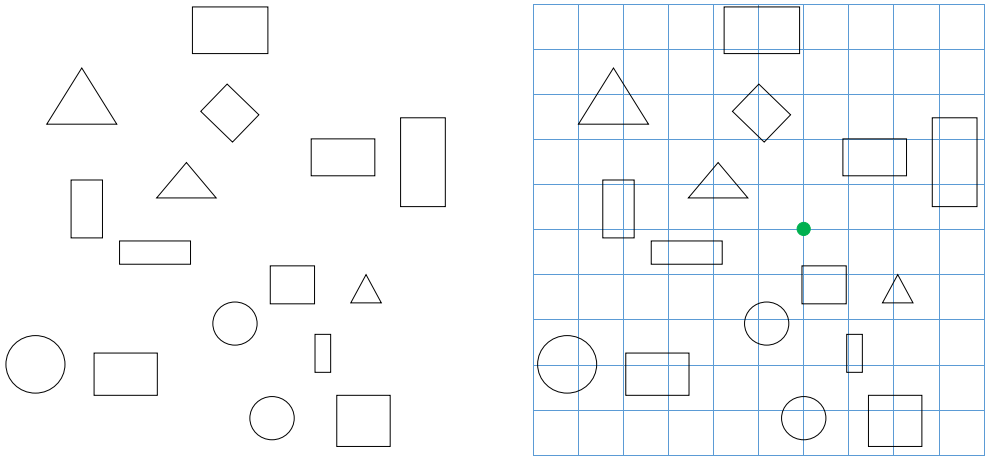
The last heuristic for the single facility case is Are Abstraction Heuristic (AAH). It is shown in the previous heuristics that the objective function value for a random initial location can be calculated by utilizing the leveling step in the polynomial time. The leveling step has time complexity of $O(log(L+m)+K(n+m))$ where K, n, m, L are number of levels, polygons, disks and corners of the polygons, respectively. Discretization of the continuous problem to the several candidate locations (not too many) and performing the leveling step on them can provide good estimation about the optimal

solution of the problem in the reasonable time. The AAH heuristic discretize the multi-level minimax problem to some candidate locations and performs leveling step for each of these points to select best of them. Next, AAH considers a smaller search region for the problem around of the obtained point and considers new candidate locations inside the new search region. Again, it finds the best location between all candidate locations and follows these steps until converging to the acceptable heuristic solution.

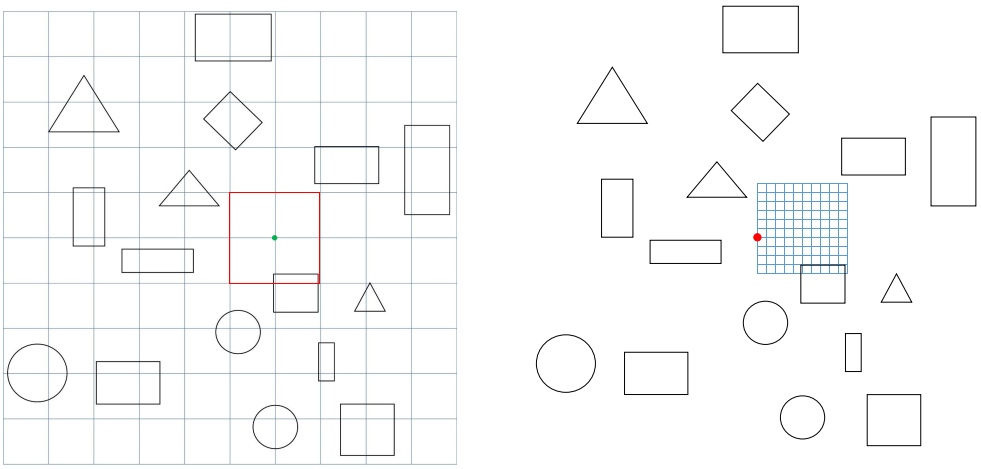
Finding candidate locations and updating the search region in each iteration are the main steps of the AAH. Number of candidate locations (α) in each iteration is fixed while the search region for selecting candidate locations get smaller iteration by iteration. Consequently, the heuristic solution quality increases in each iteration. The search region for the first iteration is enclosing square of all demand regions by the side length of γ. This square is partitioned into grids where the number of points in the grid is equal to the number of candidate locations (α). Consequently the side length of each grid for the first iteration is $\frac{\gamma}{\sqrt{\alpha}-1}$. The search region for the second iteration is a square with side length of $\frac{2\gamma}{\sqrt{\alpha}-1}$ where the best location of the first iteration is located in the middle of this square. Again, the new search region is partitioned into grids with a side length of $\frac{2\gamma}{\sqrt{\alpha}-1 \times \sqrt{\alpha}-1}$. This heuristic iterates in this manner until a stopping condition is reached (See Figure 4.5).

If the solution of the AAH in an iteration is on the border of the search region, the search region is extended by adding a parallel row or column to that border in order to obtain a solution inside the search region and gets new candidate locations. The AAH evaluates the objective function values of these new candidate locations. If the added points are promising, AAH extends the feasible region in that direction by adding another row or column.

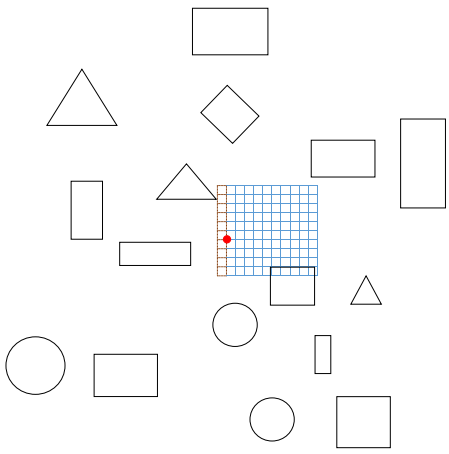
The pseudocode of the AAH heuristic is given in algorithm AAH.



(a) (b)



(c) (d)



(e)

Figure 4.5: The steps of the AAH heuristic

Algorithm AAH

Begin
Form the enclosing square of all demand regions as the initial search region.
Consider the number of candidate locations as α.
Consider the length of the enclosing square as γ_0.
Partition the enclosing square into grids with the side lenght of $\frac{\gamma_0}{\sqrt{\alpha}-1}$
Get the location of each point in the grids (x_z)
for $z = 1$ to α **do**
    for $h = 1$ to K **do**
        Perform the leveling step.
        Set $r_{zh} = (f_{hx_z} - d_h)/c_h$
    end for
    Set $r^*_{0z} = \max(r_h)$
end for
Set $r^*_0 = \min(r^*_{0z})$ and $x^*_0 = x_z$
while a stopping condition is reached **do**
    for $iter = 1$ to $Niter$ **do**
        Form a square search region around x_0 with side size of $\gamma_{iter} = \frac{2\gamma_{iter-1}}{\sqrt{\alpha}-1}$
        Partition the search region into grids with the side lenght of $\frac{\gamma_{iter}}{\sqrt{\alpha}-1}$
        Get the location of each point in the grids (x_z)
        for $z = 1$ to α **do**
            for $h = 1$ to K **do**
                Perform the leveling step.
                Set $r_{zh} = (f_{hx_z} - d_h)/c_h$
            end for
            Set $r^*_{iterz} = \max(r_h)$
        end for
        Set $r^*_{iter} = \min(r^*_{iterz})$ and $x^*_0 = x_z$
        while x_z be on the border of the search region **do**

            Extend the search region by adding a parallel row or column to that border.
            partition the new added line to $\sqrt{\alpha}$ parts.
            Find r^*_{iterz} for all points on that line.
            Set $r^*_{iter} = \min(r^*_{iterz})$ and $x^*_0 = x_z$
        end while
    end for
end while
End

Figure 4.5 displays the steps of the AAH heuristic. Figure (a) shows the demand regions which are 3 disks and 13 polygons. In the Figure (b), a search region is defined for the problem which is a enclosing square of all regions. Additionally, the search region is partitioned into grids with 121 points on it as candidate locations. The leveling step is performed on each of these candidate locations and the best of them with the lowest objective function value is selected as the promising location. A new search region in the Figure (c) is formed around of the obtained location with a specific side length. Again, the new search region is partitioned into grids with 121 points on it. The Figure (d) shows these grids and the point with lowest objective function value. Since the new obtained location is on the border of the search region, a column is added to the left of the search region and indicated in the Figure (e). The new column is also partitioned into girds and the objective function value for all points on the added column is found by leveling step. We compare the objective function values of the added points and the current location. If the added points have the lowest value, we continue the adding procedure of the columns and rows.

This heuristic checks several candidate locations in each iteration to find an improving solution. Increasing number of candidate locations in each iteration not only increases the accuracy of the heuristic but also the required running time. Consequently, there is a trade off between running time and accuracy of the heuristic in this algorithm and setting good parameters for AAH seems to be necessary. Two stopping conditions are defined for AAH. One of them is maximum number of iterations and the second one is convergence criterion similar to the previous heuristics.

4.2 Heuristics for the multi-facility case

Six heuristics for the single facility version of the problem were introduced in the previous section. Here, these heuristics are extended to the multi-facility case and one new heuristic is proposed. The commonly used heuristic for the multi-facility location problems in the literature is the Location-Allocation (LA) algorithm. Thus, we expand LA algorithm for our problem. In addition to assigning the demands to the facilities in the classical version of the problem, our problem contains assigning the demand regions to the covering levels as well. Consequently, we introduce a

Location-Allocation-Leveling (LAL) algorithm which is used in the proposed heuristics.

Location-Allocation-Leveling (LAL) algorithm

LAL algorithm considers initial locations for each facility and allocates the demand regions to the facilities and their corresponding covering levels. Demand regions are allocated to the first covering level of the closest facility until reaching the covering percentage of the first level. Then, demand regions are allocated to the second level of the closest facility and the procedure continues in this manner until allocating all demand regions. The pseudocode of the LAL algorithm is given below.

Algorithm Location-allocation-leveling algorithm

Begin
Consider T initial locations $x_{01}, x_{02}, \ldots, x_{0T}$ inside the convex hull of the demand regions.
for regions $i = 1$ to n and $j = 1$ to m **do**
    for facilities $t = 1$ to T **do**
        Find the farthest distance between $i(j)$ and x_{0t} and consider it as d_{it} or d_{jt}.
    end for
    Set $d_i = \min(d_{it})$ and $d_j = \min(d_{jt})$.
end for
Sort all regions based on d_i and d_j.
Set $N = \varnothing$, $A =$ set of all demand regions.
for $h = 1$ to K **do**
    Set $Wl_h = 0$.
    repeat
        Select the region with lowest d_i or d_j from A/N and consider it as (R).
        Assign R to the closest facility.
        $N = N \cup \{R\}$
        $Wl_h = Wl_h$+weight of R
    until sum of the weights of the regions in level h $\geq P_h$
end for
End

The steps of the LAL algorithm are shown in the Figure 4.6. Figure (a) illustrates the demand regions and 3 random initial locations for the 3-facility problem. The farthest distances between each of demand regions and facilities are obtained. The

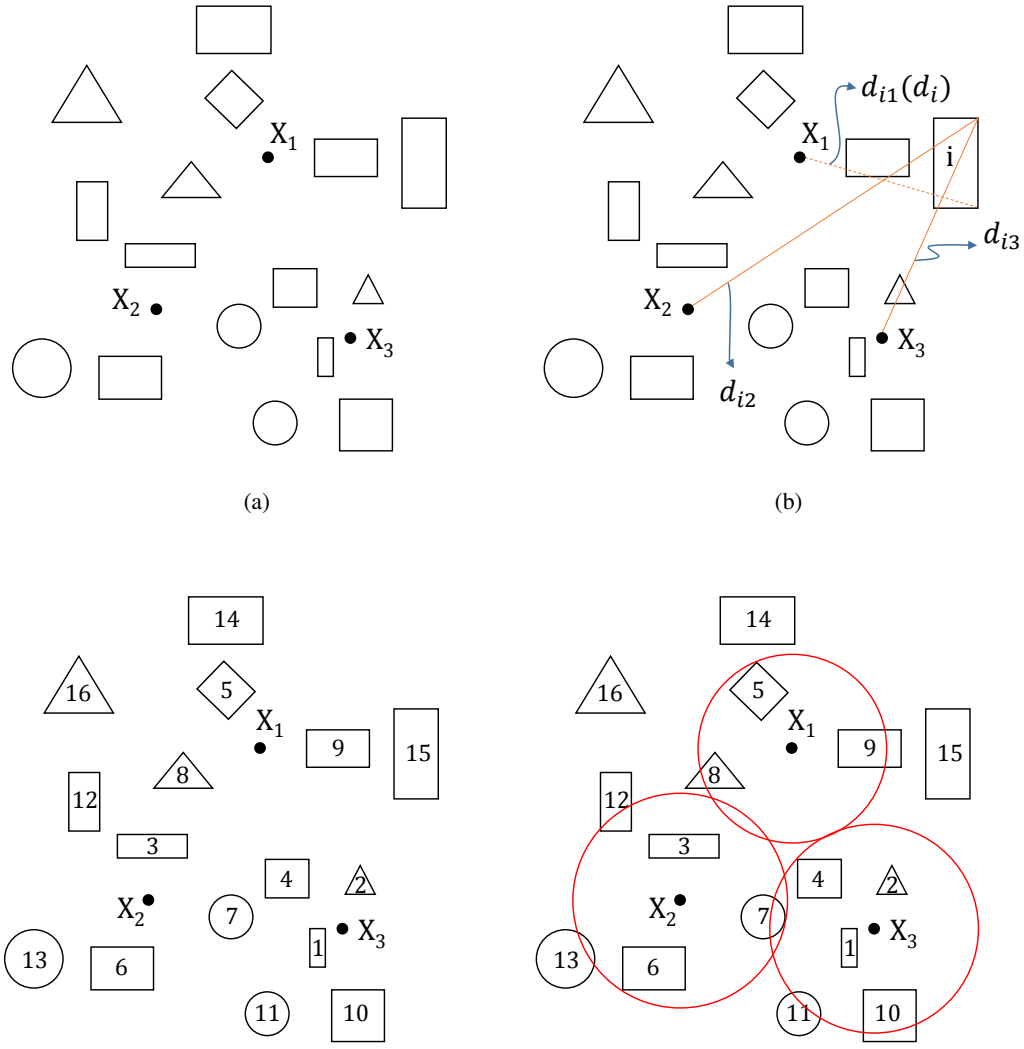


Figure 4.6: The steps of the LAL algorithm

Figure (b) displays this procedure for the demand region i. Since the closest facility for the demand region i is the first one, d_{i1} is considered as the distance of the demand i or d_i. Based on the obtained distances, the demand regions are sorted and presented in the Figure (c). Now, the demand regions are allocated to the facilities and their corresponding covering levels. Figure (d) shows the allocation procedure for the first covering level of the facilities.

4.2.1 Location-Allocation-Leveling Heuristic (LALH)

According to the described LAL algorithm, for in hand locations, allocations of the demand regions to the facilities and covering levels can be determined. Hence, the binary variables in the MISOCP formulation can be fixed and the model can be re-formulated as an SOCP problem. Location-Allocation-Leveling Heuristic (LALH) utilizes such a procedure to find the good solutions for the problem. LALH begins with some initial locations of the facilities and performs LAL algorithm for them. Then, it finds corresponding covering radius for each facility similar to the heuristics of the single facility case and reports the maximum value of them as objective function of initial locations. Based on the determined values of the binary variables in LAL step, LALH solves an SOCP problem to find a new locations for the next iteration. LALH iterates this procedure until reaching a stopping condition. The pseudocode for LALH heuristic is given in Algorithm LALH.

Algorithm LALH

1: **Begin**
2: Consider T initial locations $x_{01}, x_{02}, \ldots, x_{0T}$ inside the convex hull of the demand regions.
3: **while** a stopping condition is reached **do**
4: Perform LAL step for the initial locations (x_{0t}).
5: **for** $t = 1$ to T **do**
6: **for** $h = 1$ to K **do**
7: Set l_{th} = farthest distance between x_{0t} and the farthest region located in level h of the facility t.
8: Set $r_{th} = (l_{th} - d_h)/C_h$.
9: **end for**
10: $r_t = \max(r_{th})$
11: **end for**
12: $R^* = \max(r_t)$
13: Based on the assignments of the LAL step, solve an SOCP problem to get new x_{0t}.
14: **end while**
15: **End**

The Figure 4.7 displays the procedure of the LALH heuristic. Figures (a) to (d)

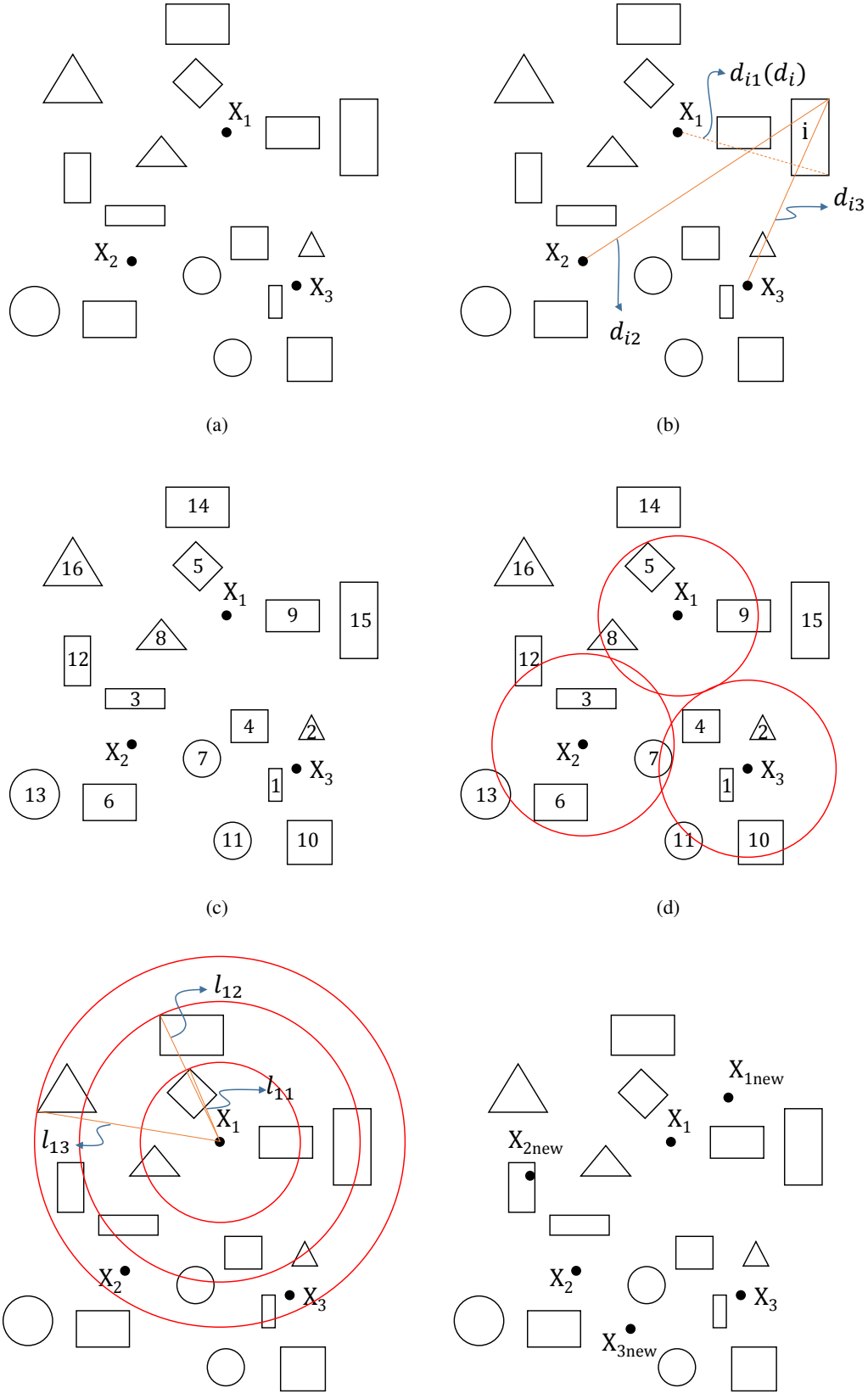


Figure 4.7: The steps of the LALH heuristic

illustrate the LAL algorithm which was discussed in the Figure 4.6. Then, the radius of the covering levels of each facility are determined. The Figure (e) shows this procedure for the first facility. The farthest distance between X_1 and the farthest region located in levels 1, 2, and 3 of the first facility (l_{11}, l_{12}, l_{13}) are obtained. Based on the determined assignments, an SOCP problem is solved for each facility to find its new location. These new locations are indicated in the Figure (f).

Time complexity of the SOCP formulation was discussed in the Section 3.1.6. LALH solves SOCP in each iteration to update facility locations. Since the complexity of SOCP is related to the number of constraints, LALH is expected to have more computation time in comparison with similar LLH heuristic in the single facility case. Stopping condition for all heuristics of the multi-facility case is the same. We consider two criteria where one of them is the maximum number of iterations and the second one is convergence conditions. Convergence condition is defined as getting 3 successive close enough solutions.

Covering Percentage Updating (CPU) algorithm

In the following sections, the single facility heuristics are extended to the multi-facility case. Our aim is separating the problem into several single facility location problems and solve them with the previously proposed heuristics. However, the multi-facility case considers the covering percentage of level h as the aggregated weights of the all regions which are allocated in the level h of all facilities. This causes some difficulties in the separation of the problem into single facility location problems. For dealing with this problem the Covering Percentage Updating (CPU) step is introduced. After performing the LAL algorithm, the CPU step gets the allocated demand regions to each level of the facilities. Then, CPU aggregates the weight of all regions which are located in a specific level of a facility to find the new covering percentage for the specific level of that facility. Since the regions and facilities are separated, the covering percentage of the last level for each facility is not necessarily equal to 1. Hence, for each facility, all levels and weight of allocated regions to that facility are multiplied by a value such that the covering percentage of the last level becomes 1. Now, the heuristics of the single facility case can be applied for the multi-facility case.

Algorithm Covering Percentage Updating (CPU) algorithm

Begin
Consider T initial locations $x_{01}, x_{02}, \ldots, x_{0T}$ inside the convex hull of the demand regions.
Perform the LAL algorithm on x_{0t}.
for facilities $t = 1$ to T **do**
    for levels $h = 1$ to K **do**
        Set $P_{th} = 0$.
        for polygonal regions $i = 1$ to n **do**
            if region i is covered by facility t in level h **then**
                $P_{th} = P_{th}+$ weight of region i
            end if
        end for
        for disk regions $j = 1$ to m **do**
            if region j is covered by facility t in level h **then**
                $P_{th} = P_{th}+$ weight of region j
            end if
        end for
    end for
    for levels $h = 1$ to K **do**
        $P_{th} = P_{th} \times (1/P_{tK})$
        for all regions (R) allocated to the facility t **do**
            weight of R = weight of $R \times (1/P_{tK})$
        end for
    end for
end for
End

4.2.2 Location-Allocation-Leveling, Critical Level Heuristic (LAL-CLH)

Location-Allocation-Leveling, Critical Level Heuristic (LAL-CLH) begins from some initial locations and performs the LAL and CPU steps in order to separate a multi-facility problem into several single facility problems and applies the proposed CLH heuristic to get new initial locations for each facility. LAL-CLH iterates this procedure until reaching the convergence criteria. The pseudocode of LAL-CLH is given below.

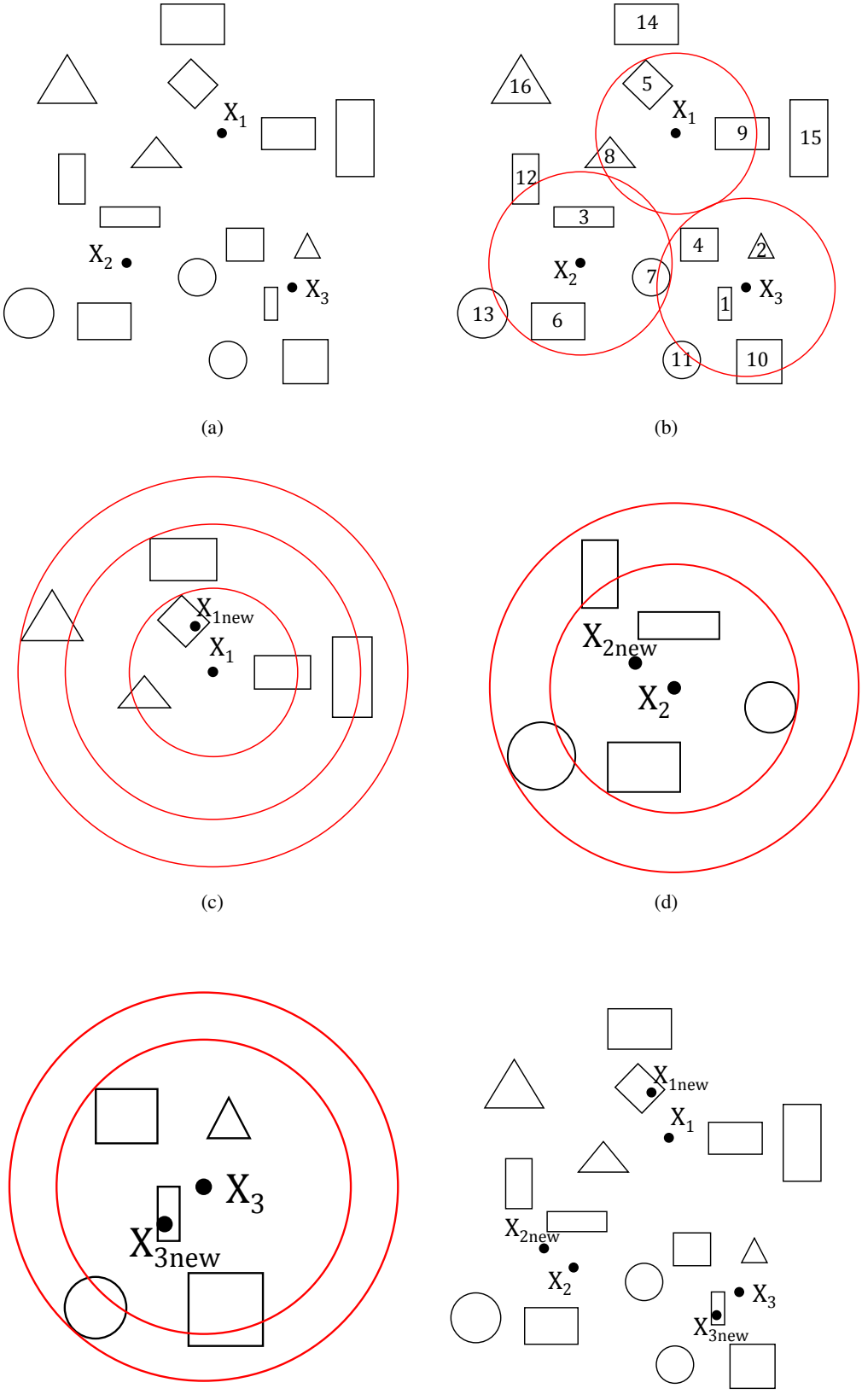
Algorithm LAL-CLH

1: **Begin**
2: Consider T initial locations $x_{01}, x_{02}, \ldots, x_{0T}$ inside the convex hull of the demand regions.
3: **while** a stopping condition is reached **do**
4: Perform the LAL algorithm on x_{0t}.
5: Perform the CPU algorithm based on the obtained allocations in the LAL step.
6: **for** facilities $t = 1$ to T **do**
7: Perform the CLH heuristic on the allocated demand regions of the facility t.
8: Set the location of the heuristic solution of the facility t as x_{0t} .
9: Set the heuristic solution value of the facility t as r_t.
10: **end for**
11: Set $R^* = \max(r_t)$.
12: Reset the weights of all demand regions to the initial values.
13: **end while**
14: **End**

The LAL-CLH algorithm is shown in Figure 4.8 as well. In Figure (a), 3 initial locations for the facilities are indicated. The demand regions are allocated to the facilities and their corresponding covering levels by LAL algorithm in the Figure (b). Then, the multi-facility problem is separated to three single facility problem. The Figures (c), (d), and (e) display the single facility problems for the first, second, and third facilities, respectively. Covering percentage of the facilities and weights of the demand regions are updated and the CLH heuristic is applied for each of these single facility problems. The new obtained locations of each facility are shown in the Figure (f).

4.2.3 Location-Allocation-Leveling, Location-Leveling Heuristic (LAL-LLH)

Similar to the previous heuristic, LAL-LLH performs the LAL and CPU steps on the starting points firstly. However, it applies the LLH heuristic on the obtained separated single facility problems to get new starting point for the next iteration. The corresponding pseudocode for this heuristic is provided in Algorithm LAL-LLH.



(a) (b) (c) (d) (e) (f)

Figure 4.8: The steps of the LAL-CLH heuristic

Algorithm LAL-LLH

1: **Begin**
2: Consider T initial locations $x_{01}, x_{02}, \ldots, x_{0T}$ inside the convex hull of the demand regions.
3: **while** a stopping condition is reached **do**
4: Perform the LAL algorithm on x_{0t}.
5: Perform the CPU algorithm based on the obtained allocations in the LAL step.
6: **for** facilities $t = 1$ to T **do**
7: Perform the LLH heuristic on the allocated demand regions of the facility t.
8: Set the location of the heuristic solution of the facility t as x_{0t} .
9: Set the heuristic solution value of the facility t as r_t.
10: **end for**
11: Set $R^* = \max(r_t)$.
12: Reset the weights of all demand regions to the initial values.
13: **end while**
14: **End**

4.2.4 Location-Allocation-Leveling, Critical Level Demand Exchange Heuristic (LAL-CLDEH)

The heuristic steps of the LAL-CLDEH is mentioned in the following pseudocode. As it was described in the single facility case, this heuristic utilizes simulated annealing in updating associated location of the facilities in each iteration.

Algorithm LAL-CLDEH

1: **Begin**
2: Consider T initial locations $x_{01}, x_{02}, \ldots, x_{0T}$ inside the convex hull of the demand regions.
3: **while** a stopping condition is reached **do**
4: Perform the LAL algorithm on x_{0t}.
5: Perform the CPU algorithm based on the obtained allocations in the LAL step.
6: **for** facilities $t = 1$ to T **do**
7: Perform the CLDEH heuristic on the allocated demand regions of the facility t.
8: Set the location of the heuristic solution of the facility t as x_{0t} .
9: Set the heuristic solution value of the facility t as r_t.
10: **end for**
11: Set $R^* = \max(r_t)$.
12: Reset the weights of all demand regions to the initial values.
13: **end while**
14: **End**

4.2.5 Location-Allocation-Leveling, Critical Level Upper Demand Exchange Heuristic (LAL-CLUDEH)

The main difference between the last heuristic and the current one (LAL-CLUDEH), is in the updating locations of each facility where LAL-CLUDEH just accepts demands from the higher level in demand exchange procedure. The pseudocode of this heuristic is given in Algorithm LAL-CLUDEH.

Algorithm LAL-CLUDEH

1: **Begin**
2: Consider T initial locations $x_{01}, x_{02}, \ldots, x_{0T}$ inside the convex hull of the demand regions.
3: **while** a stopping condition is reached **do**
4: Perform the LAL algorithm on x_{0t}.
5: Perform the CPU algorithm based on the obtained allocations in the LAL step.
6: **for** facilities $t = 1$ to T **do**
7: Perform the CLUDEH heuristic on the allocated demand regions of the facility t.
8: Set the location of the heuristic solution of the facility t as x_{0t} .
9: Set the heuristic solution value of the facility t as r_t.
10: **end for**
11: Set $R^* = \max(r_t)$.
12: Reset the weights of all demand regions to the initial values.
13: **end while**
14: **End**

4.2.6 Location-Allocation-Leveling, Descent Critical Level Demand Exchange Heuristic (LAL-DCLDEH)

LAL-DCLDEH benefits a descent algorithm in updating of the locations for each facility which means that does not accept a non improving solution inside the location updating step. Hence, it works different than the two previous simulated annealing based heuristics which move to the non improving solutions with a specific probability. The pseudocode of LAL-DCLDEH heuristic is provided in the following.

Algorithm LAL-DCLDEH

1: **Begin**
2: Consider T initial locations $x_{01}, x_{02}, \ldots, x_{0T}$ inside the convex hull of the demand regions.
3: **while** a stopping condition is reached **do**
4: Perform the LAL algorithm on x_{0t}.
5: Perform the CPU algorithm based on the obtained allocations in the LAL step.
6: **for** facilities $t = 1$ to T **do**
7: Perform the DCLDEH heuristic on the allocated demand regions of the facility t.
8: Set the location of the heuristic solution of the facility t as x_{0t} .
9: Set the heuristic solution value of the facility t as r_t.
10: **end for**
11: Set $R^* = \max(r_t)$.
12: Reset the weights of all demand regions to the initial values.
13: **end while**
14: **End**

4.2.7 Location-Allocation-Leveling, Area Abstraction Heuristic (LAL-AAH)

The structure of LAL-AAH in location updating is completely different than all of the previous mentioned heuristics for multi-facility case. In the prior heuristics, the SOCP formulation was solved for getting new starting points. However, this heuristic converts the continuous multi-facility minimax location problem to the several discrete single facility minimax location problems which are solvable in polynomial time. The associated steps of this heuristic is presented in the following pseudocode.

Algorithm LAL-AAH

1: **Begin**

2: Consider T initial locations $x_{01}, x_{02}, \ldots, x_{0T}$ inside the convex hull of the demand regions.

3: **while** a stopping condition is reached **do**

4: Perform the LAL algorithm on x_{0t}.

5: Perform the CPU algorithm based on the obtained allocations in the LAL step.

6: **for** facilities $t = 1$ to T **do**

7: Perform the AAH heuristic on the allocated demand regions of the facility t.

8: Set the location of the heuristic solution of the facility t as x_{0t} .

9: Set the heuristic solution value of the facility t as r_t.

10: **end for**

11: Set $R^* = \max(r_t)$.

12: Reset the weights of all demand regions to the initial values.

13: **end while**

14: **End**

CHAPTER 5

COMPUTATIONAL STUDIES

In this thesis, the multi-level minimax problem for both single and multi-facility cases with regional demand are considered. As it was mentioned in the background section, different versions of the continuous minimax problem have been studied in the literature. Regional demand has also been considered in some facility location problems but there are very few researches which utilize such an idea in the minimax facility location problem. Additionally, this research assumes multi-levels in the covering of all demand regions in comparison with the classical single level minimax problem. As a result, there is no benchmark study in the literature for the introduced problem in this thesis. To evaluate the proposed solution methods, several random instances are generated and the results of the exact solution methods and heuristic ones are compared in this section. Firstly, the generated instances, parameter settings for algorithms and computational results of the single facility case are described in Section 5.1. Then, in Section 5.2 the multi facility case is reported.

5.1 Computational study for the single facility case

5.1.1 Instance generation

Demand entities are considered as regions in this study and these regions are in two different forms which are polygons and disks. Also without loss of generality, all regions are assumed to be convex as taking the convex hull of the regions does not change the optimal solution.

Number of corners for each polygon is randomly distributed between 3 and 11 and moreover these regions do not have any intersection with each other. All demand regions are distributed inside a 900×900 square. Additionally, the maximum length of the sides of polygons and maximum length of circles' radius are $(900\sqrt{2})/\sqrt{\textit{Number of regions}}$.

Computational time of the single facility case is considerably less than multi-facility one. Thus, large instances are required for the single facility case to illustrate the failure of the exact solution method (MISOCP formulation).

9 sets of instances are considered for the single facility case with 36, 64, 100, 144, 196, 225, 400, 625, and 900 demand regions. Furthermore, each of these sets are tested for 2, 3, 4, and 5 levels. Consequently, 36 instances are generated for the single facility version. Summary of the problem instances are listed in the table 5.1.

5.1.2 Parameter settings of the heuristics

The proposed heuristics in Section 4 depend on some parameters which should be set. These algorithms have some common and some special parameters. Stopping conditions are the common parameters for all heuristics. Each of the heuristics are run on several instances to find the effective values for the associated parameters. All heuristics except AAH are depended on the initial location. Therefore, they are tested for 20 different initial locations and the minimum objective function values of these replications are used as the comparison criteria for parameter setting. The AAH heuristic does not use any randomness and is also independent of the initial location. Hence, the results of AAH is provided for just one replication.

CLH and LLH heuristics are very similar to each other with the same parameters. We consider two stopping conditions in these heuristics. One of these conditions is maximum number of iterations. Also, it was discussed that CLH and LLH could get stuck in the first iterations. The tested instances shows that these heuristics get stuck at most in 15 iterations and after that presents the same solution for the all next iterations. The running time for 10 iterations is almost 1 second. Setting maximum number of iterations to a number less than the required number of repetitions before convergence

Table 5.1: Randomly Generated Instances

Problem Instance	Number of Regions	Number of Levels
1	36	2
2		3
3		4
4		5
5	64	2
6		3
7		4
8		5
9	100	2
10		3
11		4
12		5
13	144	2
14		3
15		4
16		5
17	196	2
18		3
19		4
20		5
21	225	2
22		3
23		4
24		5
25	400	2
26		3
27		4
28		5
29	625	2
30		3
31		4
32		5
33	900	2
34		3
35		4
36		5

affects the solution quality. Considering the small computation time of each iteration in these heuristics, setting the maximum number of iterations to a number less than 10 does not seem reasonable. On the other hand, increasing the maximum number of iterations neither affects the quality of the solution nor the computation time due to getting stuck in the first iterations. Consequently, the maximum number of iterations is set as 10 considering the results of Table 5.2.

The second stopping condition for the CLH and LLH is convergence criterion which is set as getting three successive solutions with objective function difference less than 0.001. These algorithms get stuck after several iterations and find same solutions with equal objective function values for the next iterations. Hence, changing the

Table 5.2: CLH parameter setting

Number of demand regions	Number of levels	Max number of iterations = 5		Max number of iterations = 10		Max number of iterations = 15	
		MIN		MIN		MIN	
		Obj. value	Running time (s)	Obj. value	Running time (s)	Obj. value	Running time (s)
100	3	377.12	0.21	376.98	0.28	376.98	0.31
	4	383.63	0.23	383.63	0.24	383.63	0.25
196	3	355.88	0.38	355.88	0.4	355.88	0.38
	4	360.82	0.45	360.82	0.43	360.82	0.47
400	3	388.34	0.55	387.27	0.71	387.27	0.72
	4	381.34	0.81	381.34	0.77	381.34	0.79

convergence parameter does not affect CLH and LLH. As a result, the convergence parameter for them is taken as the value used in the other heuristics.

In the CLDEH and CLUDEH heuristics, in addition to the stopping condition parameters, there are some other parameters related to the simulated annealing which should be set. One of them is the number of iterations inside the SA procedure and the second one is temperature reduction function. There are two common approach for the cooling procedure in the literature [46]. The first one suggests the large number of iterations at few temperatures and the second one offers a small number of iterations at many temperatures. In the first approach cycling inside the SA procedure replicates for several times but the temperature reduction function is aggressive and decrease the temperature quickly. A common reduction function for the first approach is $\alpha(t) = \beta \times t$ where $0 < \beta < 1$ but 0.8-0.99 is suggested for β. Furthermore, the number of replications inside the SA procedure is considered as 10. In this thesis, the first approach is utilized for SA and its associated parameters are set. Initial temperature is considered as 10 and 6 instances are generated randomly to test different parameters to select the most proper one. β is tested for two values of 0.8 and 0.9 and number of iteration inside the SA procedure is tested for 10 and 15 iterations. Eventually, 4 cooling schedule are tested for the heuristics in Tables 5.3 and 5.4 which are (0.8, 10), (0.8, 15), (0.9, 10) and (0.9, 15).

Setting the number of replications in the SA procedure as 10 provides better solutions

Table 5.3: Parameter setting for CLDEH

Number of demand regions	Number of levels	β = 0.8 and iteration = 10		β = 0.8 and iteration = 15	
		MIN		MIN	
		Objective value	Running time (s)	Objective value	Running time (s)
100	3	376.98	1.55	376.02	1.76
	4	364.52	2	364.52	1.02
196	3	355.88	1.63	355.88	1.7
	4	339.47	6.53	339.47	6.8
400	3	387.27	2.78	387.27	2.93
	4	358.68	13.86	359.16	14.81

Table 5.4: Parameter setting for CLDEH

Number of demand regions	Number of levels	β = 0.9 and iteration = 10		β = 0.9 and iteration = 15	
		MIN		MIN	
		Objective value	Running time (s)	Objective value	Running time (s)
100	3	376.98	2.7	376.98	2.63
	4	364.52	2.51	367.66	2.34
196	3	355.88	3.7	355.88	3.49
	4	339.47	6.53	339.47	6.35
400	3	387.27	8.4	387.27	8.41
	4	358.19	13.79	359.78	13.94

Table 5.5: Parameter setting for CLDEH

Number of demand regions	Number of levels	Max number of iterations = 100		Max number of iterations = 150		Max number of iterations = 200	
		MIN		MIN		MIN	
		Obj. value	Running time (s)	Obj. value	Running time (s)	Obj. value	Running time (s)
100	3	377.81	2.84	377.48	4.21	377.3	5.81
	4	370.47	2.56	368.49	3.74	368.97	5.26
196	3	355.88	6.74	355.88	11.19	355.88	14.84
	4	341.72	3.76	341.92	5.6	341.5	7.64
400	3	387.27	14.17	387.27	23.74	387.27	31.11
	4	360.74	8.67	360.86	13.72	360.47	17.16

in both tables of 5.3 and 5.4. In the comparison between β values, they behave very similar to each other. $\beta = 0.9$ has better solution in one of the instances (400 demand regions and levels) but has almost 50 percent higher average running time for all instances. Consequently, for the CLDEH and the CLUDEH the temperature reduction coefficient (β) and number of replications inside SA procedure is considered as 0.8 and 10, respectively.

Stopping criteria for the CLDEH and the CLUDEH are again the maximum number of iterations and convergence condition. Convergence condition in SA algorithm is defined as terminating replications before temperature reaches zero. This is done by temperature reduction function that is discussed above. The second stopping condition is the maximum number of iterations. This will be activated when the temperature does not approach to zero after specific number of iterations.

The CLDEH is tested for 100, 150, and 200 maximum number of iterations to obtain the proper value for the number of iterations parameter setting. Although increasing number of iterations improves the accuracy of the heuristic, it raises the running time of the algorithm as well. Based on the indicated results in Table 5.5, 150 is selected as the maximum number of iterations.

In the DCLDEH heuristic, we just have common or general parameters to set which are the stopping conditions. For the maximum number of iterations, three values (100, 150, and 200) are tested in Table 5.6.

Table 5.6: Parameter setting for DCLDEH

Number of demand regions	Number of levels	Max number of iterations = 100		Max number of iterations = 150		Max number of iterations = 200	
		MIN		MIN		MIN	
		Obj. value	Running time (s)	Obj. value	Running time (s)	Obj. value	Running time (s)
100	3	376.98	2.61	376.98	0.47	376.02	4.67
	4	367.66	2.38	364.52	2.86	364.52	4.25
225	3	377.13	0.44	377.13	0.47	377.13	0.45
	4	351.11	4.85	351.11	0.65	351.11	0.9
400	3	387.27	0.96	387.27	1.52	387.27	1.51
	4	362.13	7.63	360.97	13.66	360.72	12.98
Time Average		-	3.145	-	3.272	-	4.127

Objective function values of the randomly generated instances for 100 iterations are worse than those for 150 and 200 iterations. In the comparison between 150 and 200 iterations, they have almost same objective values. However, the average of the running time for 150 iterations is 26% lower than that for 200 iterations. Consequently, 150 is selected as maximum number of iterations. Additionally, convergence condition stops the heuristic when 3 successive solutions with objective function value difference less than a specific number are obtained. For this aim, three values (0.01, 0.001, and 0.0001) are evaluated for the heuristic in Table 5.7. This table indicates that setting the convergence condition as 0.001 outperforms 0.01. It also has the same objective function values with 0.0001 in lower average time. Therefore, 0.001 is utilized as convergence condition.

Nature of the last heuristic is different from the first five ones, therefore, it has another parameter which should be fixed. AAH heuristic partitions the search region into grid cells. Corners of each cell represent a candidate location for the problem. As it was described in the Section 4.1.6, there is a trade off between number of cells in grid and running time of the heuristic. We evaluate 3 numbers 64, 100, 144 as the number of candidate locations in each iteration to select the better one in Table 5.8. Based on the results of this table, 100 candidate locations are evaluated in each iteration of the AAH heuristic.

Related parameters to the stopping conditions of the AAH heuristic are set by consid-

Table 5.7: Parameter setting for DCLDEH

Number of demand regions	Number of levels	Convergence condition = 0.01		Convergence condition = 0.001		Convergence condition = 0.0001	
		MIN		MIN		MIN	
		Obj. value	Running time (s)	Obj. value	Running time (s)	Obj. value	Running time (s)
100	3	376.98	5.15	376.98	4.22	376.98	5.25
	4	368.65	4.92	364.52	3.72	364.52	4.93
196	3	355.88	0.44	355.88	0.41	355.88	0.51
	4	341.13	0.73	336.99	5.66	336.99	6.81
400	3	387.27	0.98	387.27	1.06	387.27	1.16
	4	361.11	14.99	359.91	13.9	359.68	15.18
Time Average		-	4.535	-	4.828	-	5.64

Table 5.8: Parameter setting for AAH

Number of regions	Numb er of levels	Number of candidate locations = 64		Number of candidate locations = 100		Number of candidate locations = 144	
		Objective value	Running time (s)	Objective value	Running time (s)	Objective value	Running time (s)
100	3	375.53	0.08	375.53	0.09	375.53	0.1
	4	364.52	0.06	364.52	0.06	364.52	0.07
196	3	355.88	0.08	355.88	0.1	355.88	0.11
	4	341.87	0.08	339.89	0.09	339.89	0.11
400	3	387.27	0.16	387.27	0.21	387.27	0.23
	4	359.68	0.17	358.19	0.2	358.19	0.24

Table 5.9: Parameter setting for AAH

Number of regions	Number of levels	(0.01, 15)		(0.01, 20)		(0.01, 25)	
		Objective value	Running time (s)	Objective value	Running time (s)	Objective value	Running time (s)
100	3	375.54	0.08	375.54	0.08	375.54	0.08
	4	364.52	0.06	364.52	0.06	364.52	0.06
196	3	355.88	0.08	355.88	0.08	355.88	0.08
	4	344.93	0.08	344.93	0.08	344.93	0.08
400	3	387.28	0.16	387.28	0.16	387.28	0.16
	4	359.57	0.17	359.57	0.19	359.57	0.16

Table 5.10: Parameter setting for AAH

Number of regions	Number of levels	(0.001, 15)		(0.001, 20)		(0.001, 25)	
		Objective value	Running time (s)	Objective value	Running time (s)	Objective value	Running time (s)
100	3	375.53	0.09	375.53	0.09	375.53	0.09
	4	364.52	0.06	364.52	0.06	364.52	0.06
196	3	355.88	0.09	355.88	0.1	355.88	0.1
	4	344.93	0.09	344.93	0.09	344.93	0.1
400	3	387.27	0.2	387.27	0.2	387.27	0.2
	4	359.56	0.2	359.56	0.2	359.56	0.2

ering the result of Tables 5.9, 5.10 and 5.11. Three values for convergence condition and three values for the maximum number of iterations are considered. Combination of these cases provide 9 sets as follows. (0.01, 15), (0.01, 20), (0.01, 25), (0.001, 15), (0.001, 20), (0.001, 25), (0.0001, 15), (0.0001, 20) and finally (0.0001, 25). The first element of each set represents convergence value and the second element provides maximum number of iterations.

AAH heuristic is a very fast algorithm with high accuracy. It converges to an acceptable solution very quickly. Therefore, the objective function values for all combinations in Tables 5.9, 5.10 and 5.11 are close to each other. Based on the running times of these combinations, the fourth set is selected which is (0.001, 15).

Finally, the obtained results for parameter setting of all heuristics are summarized in Table 5.12.

Table 5.11: Parameter setting for AAH

Number of regions	Number of levels	(0.0001, 15)		(0.0001, 20)		(0.0001, 25)	
		Objective value	Running time (s)	Objective value	Running time (s)	Objective value	Running time (s)
100	3	375.53	0.09	375.53	0.1	375.53	0.09
	4	364.52	0.07	364.52	0.06	364.52	0.07
196	3	355.88	0.1	355.88	0.1	355.88	0.1
	4	344.93	0.1	344.93	0.1	344.93	0.1
400	3	387.27	0.22	387.27	0.21	387.27	0.22
	4	359.56	0.22	359.56	0.22	359.56	0.22

Table 5.12: Parameter setting for all heuristics

Heuristic Name	Number of iterations	Convergence condition	Other parameters
CLH	10	0.001	-
LLH	10	0.001	-
CLDEH	150	$\beta = 0.8$	Number of repetitions in SA procedure = 10
CLUDEH	150	$\beta = 0.8$	Number of repetitions in SA procedure = 10
DCLDEH	150	0.001	-
AAH	15	0.001	Number of candidate locations = 100

5.1.3 Computational results

In this section, the computational results for both exact and heuristic algorithms are presented. Since there is no benchmark problem for our study, the exact solutions of MISOCP formulation for our proposed model are found firstly and heuristics are compared with results of the exact solutions. All algorithms have been coded in .NET framework by C# 2013 language. Also these codes have been run on a PC with 3.22 GHz CPU and 16GB of RAM.

5.1.3.1 Exact solution

There are few solvers such as CPLEX, GUROBI, and MOSEK which can deal with an MISOCP formulation. Considering the performed comparisons between these solvers in the literature [25], CPLEX is selected as the solver in our study.

Time complexity of the MISOCP formulation has been discussed in section 3.1.6. It contains big-M values which make the formulation weak. Hence, it is unable to solve large instances, to optimality. During its solution by branch-and-bound (B&B) technique, the B&B tree becomes too large and out of memory error appears. For example, the instances number 24, 27, 28, 31 and 32 which represent more than 225 demand regions and more than 4 levels could not be solved optimally in the time limit of 48 hours.

In order to check all instances and compare them with heuristics, the time limit of 3600 seconds is set for solving the MISOCP formulation and results of those instances which could not be solved within the time limit have been reported with corresponding optimality gap. In Table 5.13, the results of the MISOCP formulation for the problem instances and their associated computation times have been reported.

Table 5.13 provides several useful data about the problem. At first glance, it displays the effect of instance size on the computation time. As it was mentioned in Section 3.1.6, solving large instances by MISOCP is problematic. Although CPLEX uses lots of cuts, it could not solve the instance in a reasonable time. Also, optimality gap for large instances is remarkable. Hence, the obtained results of the MISOCP

Table 5.13: Optimal or the best found objective function values by the MISOCP formulation and the associated computation times (in seconds)

problem instance	Number of regions	Number of levels	Objective function value	Running time (s)	GAP (%)
1	36	2	436.57	0.312	0
2		3	366.628	3.073	0
3		4	319.08	9.797	0
4		5	319.08	13.993	0
5	64	2	459.637	0.421	0
6		3	384.303	3.946	0
7		4	341.699	22.963	0
8		5	341.699	42.962	0
9	100	2	435.285	0.717	0
10		3	375.529	5.335	0
11		4	364.517	45.927	0
12		5	364.517	1084.94	0
13	144	2	461.255	1.654	0
14		3	381.3	18.439	0
15		4	360.716	95.956	0
16		5	360.716	3600	12.634
17	196	2	438.464	1.763	0
18		3	355.876	13.432	0
19		4	338.088	3600	13.66
20		5	336.992	3600	21.246
21	225	2	464.617	7.893	0
22		3	377.13	23.712	0
23		4	351.11	2259.799	0
24		5	359.639	3600	26.165
25	400	2	477.107	13.416	0
26		3	387.27	29.781	0
27		4	364.194	3600	22.111
28		5	373.522	3600	48.609
29	625	2	477.498	20.405	0
30		3	387.59	96.362	0
31		4	357.873	3600	29.913
32		5	364.706	3600	51.306
33	900	2	473.753	26.957	0
34		3	384.557	186.141	0
35		4	379.972	3600	40.004
36		5	375.624	3600	56.007

formulation for them is expected to be worse than the heuristic solutions. Another useful information from the Table 5.13 is downward trend of the objective function values by increase in the number of levels. This trend stems from the nature of the problem which reduces the effect of remote demand regions level by level.

5.1.3.2 Heuristics' solutions

After getting the optimal or best found solutions by the MISOCP formulation, the proposed heuristics are applied to solve the generated instances of the single facility multi-level minimax problem. Furthermore, the obtained results of the heuristics are compared with each other to evaluate their performance. Heuristics are compared in two aspects: the objective function value and corresponding computation time. Since the SCOP based heuristics are depended on the initial facility location, each instance is solved with 20 distinct initial locations. The last starting point of the heuristics for each instance is the location of the optimal solution in solving the classical minimax location problem.

The Table 5.13 indicates that large instances can not be solved by MISOCP formulation, to optimality. For these instances, the best solution between MISOCP formulation and heuristics is considered as comparison criteria (See Table 5.14). Each of the instances are solved with the first five heuristics by 20 different initial locations and the percent deviation from the best solution of these instances are found. Average and minimum of percent deviations for these replications are reported in the Tables 5.15 and 5.16, respectively. Since AAH is independent of the initial location, it is run once and the associated result is obtained. Therefore, the results of AAH in both average and minimum percent deviation tables are the same. The best found solution by heuristics is important in evaluating the performance of them. Accordingly, the minimum results for these 20 replications are listed in the Table 5.16. The instances which are specified by a star represent the instances which are solved optimally.

The Table 5.16 illustrates some of the described properties of the heuristics which were explained in Section 4. As it was mentioned, Heuristics CLH and LLH get stuck quickly. Therefore, they have the biggest amount of the percent deviation from the best solution. The CLH and LLH have the same maximum deviation between

Table 5.14: The best found objective value between MISOCP formulation and heuristics

Problem instances	Number of regions	Number of levels	Objective function value of MISOCP	The best found objective value between heuristics and MISOCP
1*	36	2	436.57	436.57
2*		3	366.628	366.628
3*		4	319.08	319.08
4*		5	319.08	319.08
5*	64	2	459.637	459.637
6*		3	384.303	384.303
7*		4	341.699	341.699
8*		5	341.699	341.699
9*	100	2	435.285	435.285
10*		3	375.529	375.529
11*		4	364.517	364.517
12*		5	364.517	364.517
13*	144	2	461.255	461.255
14*		3	381.3	381.3
15*		4	360.716	360.716
16		5	360.716	360.716
17*	196	2	438.464	438.464
18*		3	355.876	355.876
19		4	338.088	336.99
20		5	336.992	336.992
21*	225	2	464.617	464.617
22*		3	377.13	377.13
23*		4	351.11	351.11
24		5	359.639	351.11
25*	400	2	477.107	477.107
26*		3	387.27	387.27
27		4	364.194	358.19
28		5	373.522	358.19
29*	625	2	477.498	477.498
30*		3	387.59	387.59
31		4	357.873	345.07
32		5	364.706	345.1
33*	900	2	473.753	473.753
34*		3	384.557	384.557
35		4	379.972	357.82
36		5	375.624	357.82

Table 5.15: Average of the % deviation of heuristics from best found solution

Instances	Number of demand regions	Number of levels	Heuristics					
			CLH	LLH	CLDEH	CLUDEH	DCLDEH	AAH
1*	36	2	0	0	0	0	0	0
2*		3	0.82	8.34	0.54	0.56	0.57	0.44
3*		4	21.75	20.13	6.61	9.81	5.27	0
4*		5	21.75	20.96	7.55	10.69	4.51	0
5*	64	2	0	0	0	0	0	0.01
6*		3	0.53	1.68	0.53	0.52	0.53	0
7*		4	14.76	12.86	0.93	2.72	1.25	0
8*		5	14.76	12.61	1.11	3.19	1.62	0
9*	100	2	0	0	0	0	0	0.07
10*		3	7.89	4.69	0.71	3.16	0.38	0
11*		4	10.61	10.5	1.35	2.99	1.68	0
12*		5	10.61	10.54	1.1	2.72	2.67	0
13*	144	2	0	0	0	0	0	0.04
14*		3	0.81	2.54	0.38	0.42	0.49	0
15*		4	12.22	11.76	2.54	3.32	1.77	0.01
16		5	12.22	12.08	1.97	4.13	1.6	0.01
17*	196	2	0	0	0	0	0	0
18*		3	0	1.15	0	0	0	0
19		4	10.17	10.61	1.48	1.66	1.5	0.81
20		5	10.17	10.42	1.41	1.52	1.4	0.81
21*	225	2	0	0	0	0	0	0
22*		3	0	0.01	0	0	0	0
23*		4	11.32	11.31	2.18	2.97	2.01	0
24		5	11.32	11.39	2.41	3.06	1.71	0
25*	400	2	0	0	0	0	0	0
26*		3	0	0	0	0	0	0
27		4	11.04	11.19	0.54	1.98	1.76	0.38
28		5	11.04	11.51	0.65	2.24	0.82	0.38
29*	625	2	0	0	0	0	0	0
30*		3	0	0	0	0	0	0
31		4	11	11.29	0.96	1.6	1.22	0.01
32		5	10.99	11.53	0.86	1.6	1.47	0
33*	900	2	0	0	0	0	0	0
34*		3	0	0	0	0	0	0
35		4	10.77	10.75	0.49	1.57	0.48	0.35
36		5	10.77	10.77	0.47	1.56	0.5	0.35
Average deviation			6.592	6.684	1.021	1.778	0.978	0.102
Maximum deviation			21.75	20.96	7.55	10.69	5.27	0.81

Table 5.16: Minimum of the % deviation of heuristics from best found solution

Instances	Number of demand regions	Number of levels	Heuristics						Initial locations
			CLH	LLH	CLDEH	CLUDEH	DCLDEH	AAH	
1*	36	2	0	0	0	0	0	0	0
2*		3	0.62	0	0.54	0.54	0.54	0.44	5.8
3*		4	0.85	0	0.85	0.85	0.85	0	20.48
4*		5	0.85	0	0.85	0.85	0.85	0	20.48
5*	64	2	0	0	0	0	0	0.01	0
6*		3	0.53	0	0.53	0.48	0.48	0	4.36
7*		4	3.32	3.32	0.12	0.12	0.12	0	7.09
8*		5	3.32	3.32	0.12	0.12	0.12	0	7.09
9*	100	2	0	0	0	0	0	0.07	0
10*		3	0.39	1.39	0.37	0.39	0.36	0	9.97
11*		4	3.91	3.91	0.86	0.86	0	0	4.52
12*		5	3.91	3.91	0	0	0	0	4.52
13*	144	2	0	0	0	0	0	0.04	0
14*		3	0.06	0	0.06	0.06	0.06	0	1.59
15*		4	3.64	3.64	1.32	0	0	0.01	5.88
16		5	3.64	3.64	0	1.52	0	0.01	5.88
17*	196	2	0	0	0	0	0	0	0
18*		3	0	0	0	0	0	0	0
19		4	1.69	1.69	0.74	0	0.74	0.81	2.73
20		5	1.69	1.69	0	1.19	0	0.81	2.73
21*	225	2	0	0	0	0	0	0	0
22*		3	0	0	0	0	0	0	0
23*		4	3.22	3.05	0	0	0	0	6.88
24		5	3.22	3.22	0	1.46	0	0	6.88
25*	400	2	0	0	0	0	0	0	0
26*		3	0	0	0	0	0	0	0
27		4	3.61	3.61	0	0	0.1	0.38	4.3
28		5	3.61	3.61	0	0.1	0	0.38	4.3
29*	625	2	0	0	0	0	0	0	0
30*		3	0	0	0	0	0	0	0
31		4	1.89	1.89	0	0.84	0.19	0.01	2.06
32		5	1.88	1.88	0.07	0.61	0.48	0	2.05
33*	900	2	0	0	0	0	0	0	0
34*		3	0	0	0	0	0	0	0
35		4	0.5	0.5	0	0	0	0.35	0.67
36		5	0.5	0.5	0	0.09	0.24	0.35	0.67
Average deviation			1.301	1.244	0.179	0.28	0.143	0.102	3.637
Maximum deviation			3.91	3.91	1.32	1.52	0.85	0.81	20.48
Number of instances which find the best solution			14	19	24	20	23	23	14

instances but the average deviation of LLH is a little bit better. Thus, LLH could be more reliable than CLH. Simulated annealing helps the heuristics CLDEH and CLUDEH to get out of stuck. They could search most points of the feasible region to find near optimal solutions. Forcing the algorithm to select input demand regions from one upper level restricts the heuristic to check feasible region effectively. Consequently CLUDEH has worser results than CLDEH. Alos the Table 5.16 shows that the proposed decent heuristic (DCLDEH) outperforms other SOCP based heuristics. Considering all heuristics, the last one (AAH) performs better than all others. It can find near optimal solution in almost all instances. AAH reduces the feasible region iteration by iteration and simultaneously checks to not miss optimal solution in this reduction procedure. As a result, it could check feasible region effectively to find near optimal solutions.

A good heuristic should provide acceptable solution in a reasonable time. Thus, the associated computation time of the Table 5.16 for all the heuristics is presented in Table 5.17. CLH and LLH get stuck and leave algorithm in the initial iterations. Accordingly, they have lowest running time among SOCP based heuristics. AAH heuristic searches feasible region effectiely and converges after few iterations. Additionally, it uses another approach to find fitness function in each iteration, which is faster than SOCP formulation. Consequently the computation time of the AAH is low. It can be concluded that AAH obtains better results among all heuristics in the very low running time which confirms its quality.

Each of the SOCP based heuristics has specific parameters and stopping conditions. Some of them leave the heuristic in the first iterations and some others could traverse feasible region in more iterations to get near optimal solution. This makes the comparison between SOCP based heuristics unrealistic. To evaluate all SOCP based heuristics in the same condition, the time limit of 180 seconds is considered for all heuristics. Each instance is run for 180 seconds and as soon as the heuristic leaves the algorithm before the 180 seconds, it starts again with the new initial location until approaching the determined time limit. Therefore, CLH and LLH can check more initial locations in comparison with the other SOCP based heuristics which has more computation time. The number of initial location that each of these heuristics can check in the time limit of 180 seconds is presented in the Table 5.18. Also, average

Table 5.17: Computation time (in seconds) of the heuristics associated to the solutions of the Table 5.16

Instances	Number of demand regions	Number of levels	Heuristics					
			CLH	LLH	CLDEH	CLUDEH	DCLDEH	AAH
1	36	2	0.1	0.17	1.59	1.57	0.77	0.25
2		3	0.12	0.25	1.59	1.57	1.72	0.24
3		4	0.16	0.34	1.97	1.66	1.52	0.25
4		5	0.16	0.48	2.81	2.82	1.57	0.3
5	64	2	0.13	0.24	2.25	2.1	0.39	0.26
6		3	0.15	0.51	2.24	2.11	1.91	0.34
7		4	0.13	0.51	3.18	2.75	2.17	0.26
8		5	0.14	0.9	4.42	4.44	2.15	0.33
9	100	2	0.19	0.43	3.74	3.61	0.76	0.4
10		3	0.23	0.75	3.74	3.59	4.22	0.48
11		4	0.19	0.58	4.14	3.83	3.57	0.47
12		5	0.2	0.81	6.05	6.02	3.24	0.48
13	144	2	0.25	0.58	5.05	4.8	0.51	0.47
14		3	0.26	1.37	5.07	4.77	7.07	0.54
15		4	0.26	1.11	6.96	5.24	5.12	0.45
16		5	0.26	1.47	9.26	9.18	5.12	0.48
17	196	2	0.28	0.65	5.55	5.13	1.24	0.56
18		3	0.26	1.48	5.55	5.24	0.73	0.51
19		4	0.59	1.19	11.04	10.98	5.4	0.66
20		5	0.58	1.6	10.92	10.83	5.39	0.57
21	225	2	0.38	0.84	7.41	6.97	0.88	0.56
22		3	0.34	2.92	7.37	6.96	0.67	0.54
23		4	0.37	1.66	13.82	13.77	7.06	0.53
24		5	0.37	2.19	13.92	13.83	7.16	0.5
25	400	2	0.59	1.45	13.58	12.09	1.56	0.75
26		3	0.48	3.71	13.29	12	1.51	0.73
27		4	0.65	3.23	23.32	23.16	12.98	0.7
28		5	0.65	3.52	23.72	23.57	13.64	0.76
29	625	2	0.86	2.34	19.18	13.97	4.96	0.83
30		3	0.78	5.16	19.05	15.02	6.14	0.86
31		4	0.99	4.68	36.31	36.38	18.52	0.85
32		5	0.99	5.6	36.99	36.96	15.95	0.85
33	900	2	1.34	3.69	26.97	19.69	54.22	1.07
34		3	1.15	8.88	26.93	21.83	46.05	1.25
35		4	1.2	5.98	52.24	52.25	27.56	3.23
36		5	1.21	7.52	53.86	53.91	27.22	1.05

and minimum of percent deviations for these replications are provided in the Tables 5.19 and 5.20, respectively.

Considering the computation time of the proposed heuristics and exact algorithm (MISOCP formulation) indicates efficiency of our heuristics.

In the following, several figures are presented to provide a suitable vision about the performance of the heuristics. Figure 5.1 displays results of the Table 5.16 in one chart. The behavior and quality of each heuristic is specified. It is seen that all SOCP based heuristics have similar behavior on the instances and in most of the instances of level 3, deviation is almost zero. Table 5.16 shows the percent deviation of the objective function value of the heuristics from the objective function value of the best found solutions. The found location of these heuristics is also important. Figures 5.2, 5.3 and 5.4 indicate the obtained location of the facility for each heuristic and compare with the solution of the MISOCP formulation. These figures represent instances number 10, 15 and 24 respectively.

Additionally, some figures are provided for the comparison between computation time of the heuristics. Figure 5.5 displays the computation time of the heuristics on all instances. Figures 5.6 to 5.11 illustrate computation time of the each heuristic on all instances. It is obvious that increasing the size of instances has direct relation on the running time of the heuristics.

Another significant aspect in the heuristics which use randomness is the number of times that the heuristic finds good solutions in several repetitions. For this purpose, CLDEH, CLUDEH and DCLDEH which have random parameters have been replicated 100 times for the instance with 144 regions and 4 levels and histogram chart of results is provided in the Figure 5.12. This figure indicates that DCLDEH and CLDEH could find good results in lots of replications, therefore, they are more reliable than CLUDEH.

Table 5.18: Number of initial locations checked by the heuristics in a time limit

Instances	Number of demand regions	Number of levels	Heuristics				
			CLH	LLH	CLDEH	CLUDEH	DCLDEH
1	36	2	2245	1025	54	64	207
2		3	1929	652	93	106	117
3		4	1504	501	117	114	119
4		5	1584	343	117	115	116
5	64	2	1565	720	42	41	464
6		3	1422	321	58	65	102
7		4	1070	366	82	86	82
8		5	1102	216	82	87	82
9	100	2	1094	387	30	30	160
10		3	876	218	44	49	46
11		4	1010	272	49	51	55
12		5	1011	168	50	53	55
13	144	2	785	306	20	21	323
14		3	751	122	26	37	27
15		4	774	175	37	40	39
16		5	773	132	37	40	37
17	196	2	652	272	18	18	209
18		3	709	112	17	18	321
19		4	623	148	33	36	34
20		5	631	114	33	35	35
21	225	2	482	217	14	13	196
22		3	580	62	14	13	227
23		4	399	66	25	27	25
24		5	426	88	25	26	25
25	400	2	323	127	8	8	112
26		3	391	51	9	8	125
27		4	243	62	14	15	14
28		5	282	53	14	15	16
29	625	2	222	78	5	5	37
30		3	243	32	5	5	28
31		4	166	40	10	12	11
32		5	150	29	10	12	11
33	900	2	132	49	4	4	4
34		3	164	21	4	4	4
35		4	92	19	7	8	8
36		5	145	25	7	9	7

Table 5.19: Average % deviation of the heuristics for running them in a time limit

Instances	Number of demand regions	Number of levels	Heuristics				
			CLH	LLH	CLDEH	CLUDEH	DCLDEH
1*	36	2	0	0	0	0	0
2*		3	0.8	7.17	0.55	0.55	0.56
3*		4	17.98	17.12	6.84	9.66	5.69
4*		5	17.83	16.34	5.18	8.52	5.05
5*	64	2	0	0	0	0	0
6*		3	0.53	1.65	0.53	0.52	0.53
7*		4	13.17	12.99	1.62	2.39	1.81
8*		5	13.04	12.45	1.84	2.16	1.76
9*	100	2	0	0	0	0	0
10*		3	6.93	3.9	0.6	2.6	0.68
11*		4	9.48	8.92	1.18	2.52	2
12*		5	9.48	9.03	0.95	2.45	1.47
13*	144	2	0	0	0	0	0
14*		3	0.9	2.62	0.24	0.27	0.41
15*		4	10.83	9.52	2.5	3.03	2.35
16		5	10.81	10.45	2.01	3.7	1.75
17*	196	2	0	0	0	0	0
18*		3	0	0.89	0	0	0
19		4	9.3	9.11	1.45	1.49	1.55
20		5	9.63	8.82	1.39	1.45	2.02
21*	225	2	0	0	0	0	0
22*		3	0	0.01	0	0	0
23*		4	10.31	10.13	1.31	2.98	1.93
24		5	10.25	10.31	2.28	2.85	2.11
25*	400	2	0	0	0	0	0
26*		3	0	0	0	0	0
27		4	9.17	8.94	0.72	2.12	0.64
28		5	7.83	8.45	0.91	1.43	2.36
29*	625	2	0	0	0	0	0
30*		3	0	0	0	0	0
31		4	10.4	8.68	1.05	1.02	1.12
32		5	9.61	10.33	0.95	1.41	0.98
33*	900	2	0	0	0	0	0
34*		3	0	0	0	0	0
35		4	8.58	7.8	0.53	1.41	0.37
36		5	9.31	8.52	0.64	2.33	0.51
Average deviation			5.725	5.677	0.98	1.579	1.043
Maximum deviation			17.98	17.12	6.84	9.66	5.69

Table 5.20: Minimum % Deviation of the heuristics for running them in a time limit

Instances	Number of demand regions	Number of levels	Heuristics				
			CLH	LLH	CLDEH	CLUDEH	DCLDEH
1*	36	2	0	0	0	0	0
2*		3	0.54	0	0.54	0.54	0.54
3*		4	0.85	0	0.85	0.85	0.85
4*		5	0.85	0	0.85	0.85	0.85
5*	64	2	0	0	0	0	0
6*		3	0.53	0	0.48	0.48	0.48
7*		4	0.12	0.62	0.12	0.12	0.12
8*		5	0.12	0	0.12	0.12	0.12
9*	100	2	0	0	0	0	0
10*		3	0.39	1.39	0.36	0.37	0.37
11*		4	0	0	0	0	0
12*		5	0	0	0	0	0
13*	144	2	0	0	0	0	0
14*		3	0.06	0	0.06	0.06	0.06
15*		4	2.46	0	0	0	0
16		5	1.32	0	0	1.52	0
17*	196	2	0	0	0	0	0
18*		3	0	0	0	0	0
19		4	0.08	1.23	0.65	0.08	0.71
20		5	0.71	0.33	0.08	0	0.74
21*	225	2	0	0	0	0	0
22*		3	0	0	0	0	0
23*		4	0	1.18	0	1.46	0
24		5	1.59	2.48	1.18	1.18	0
25*	400	2	0	0	0	0	0
26*		3	0	0	0	0	0
27		4	0.7	0.78	0	0.14	0.1
28		5	1.64	0.1	0.46	0.38	0.44
29*	625	2	0	0	0	0	0
30*		3	0	0	0	0	0
31		4	1.79	1.75	0.83	0.08	0.31
32		5	1.35	3.2	0.54	0.6	0.45
33*	900	2	0	0	0	0	0
34*		3	0	0	0	0	0
35		4	1.57	0.75	0.3	0.55	0.3
36		5	0.51	4.18	0.37	1.06	0.38
Average deviation			0.477	0.5	0.216	0.29	0.189
Maximum deviation			2.46	4.18	1.18	1.52	0.85

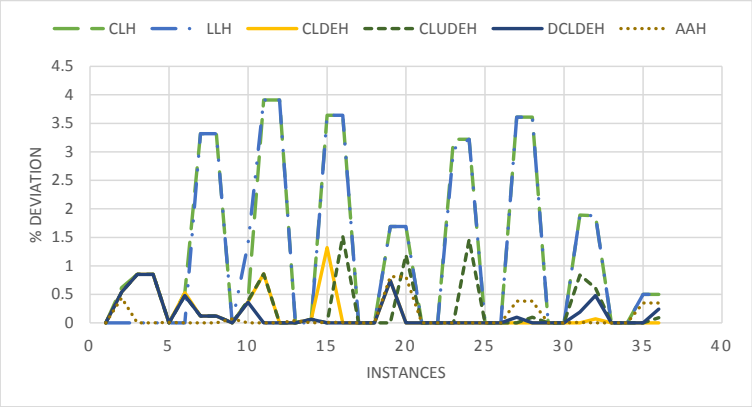


Figure 5.1: Deviations of heuristics for all instances

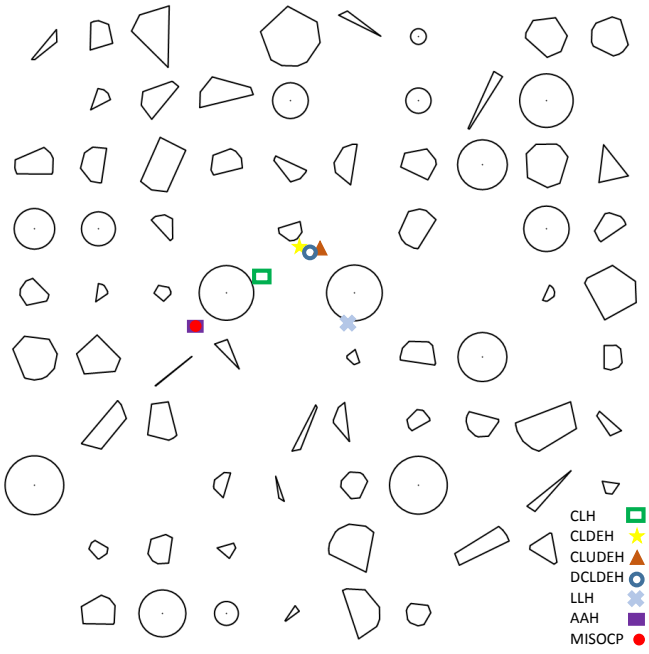


Figure 5.2: Facility locations found with heuristics for instance number 10

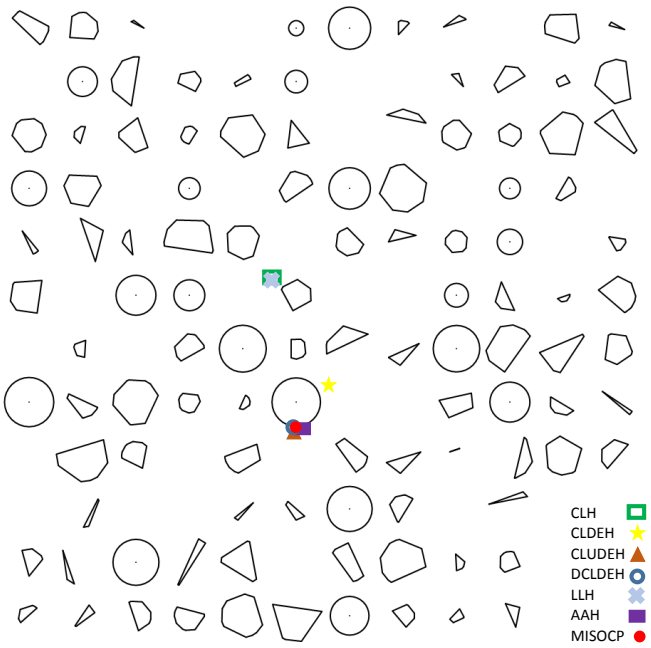


Figure 5.3: Facility locations found with heuristics for instance number 15

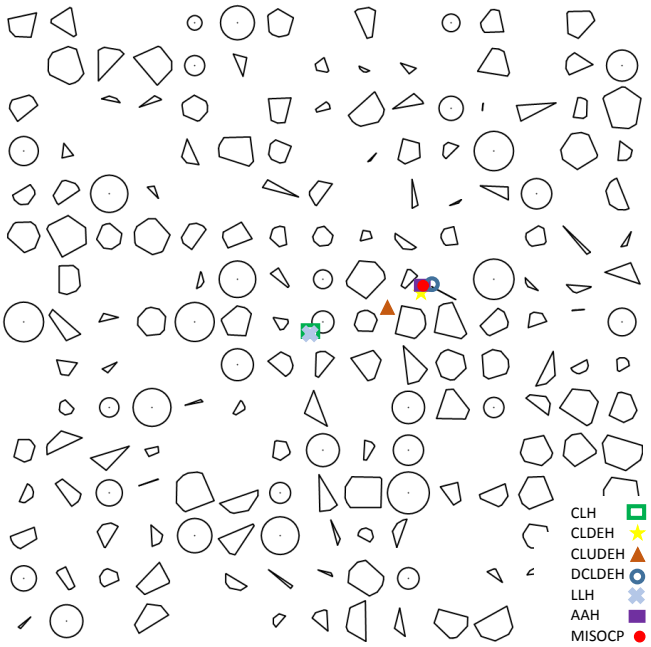


Figure 5.4: Facility locations found with heuristics for instance number 24

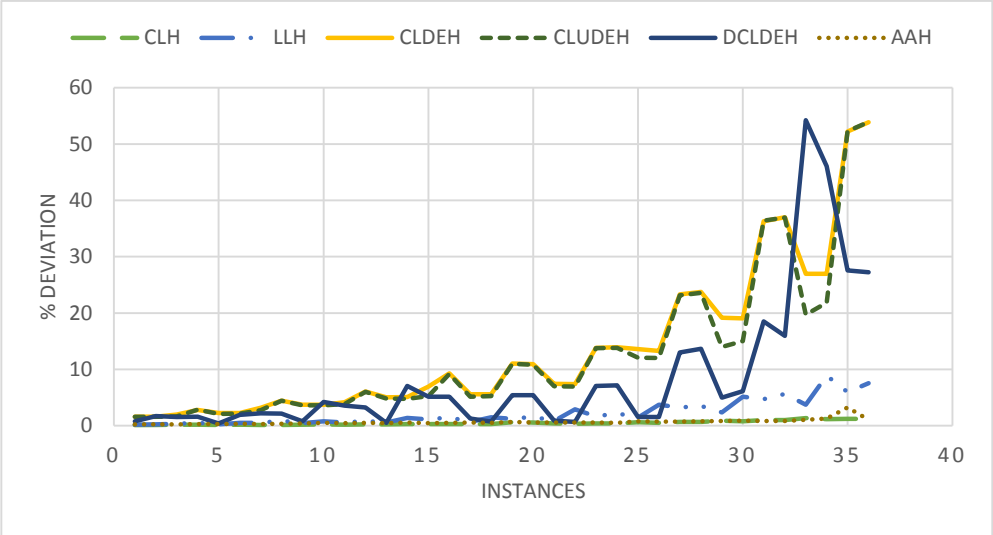


Figure 5.5: Computation time of heuristics for all instances

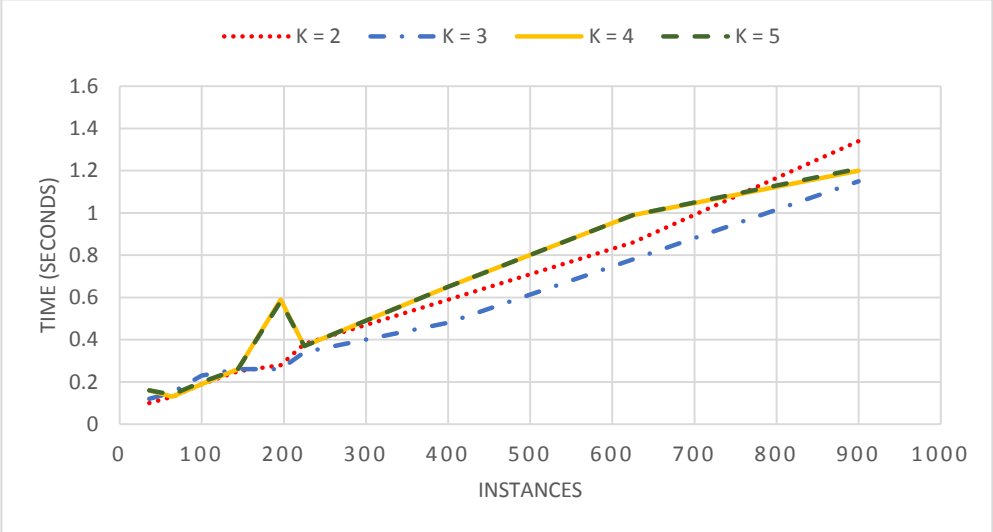


Figure 5.6: Computation time of CLH with respect to the number of regions and levels

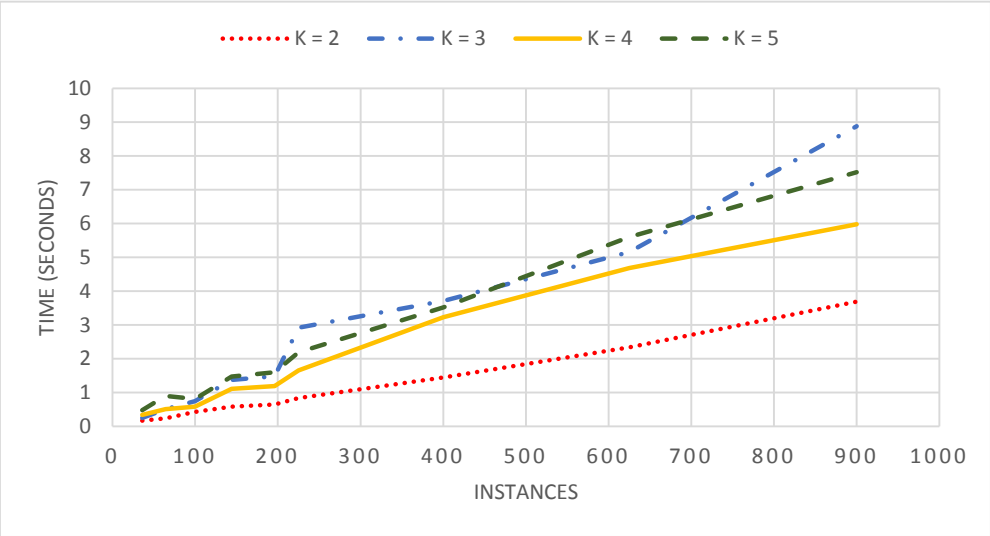


Figure 5.7: Computation time of LLH with respect to the number of regions and levels

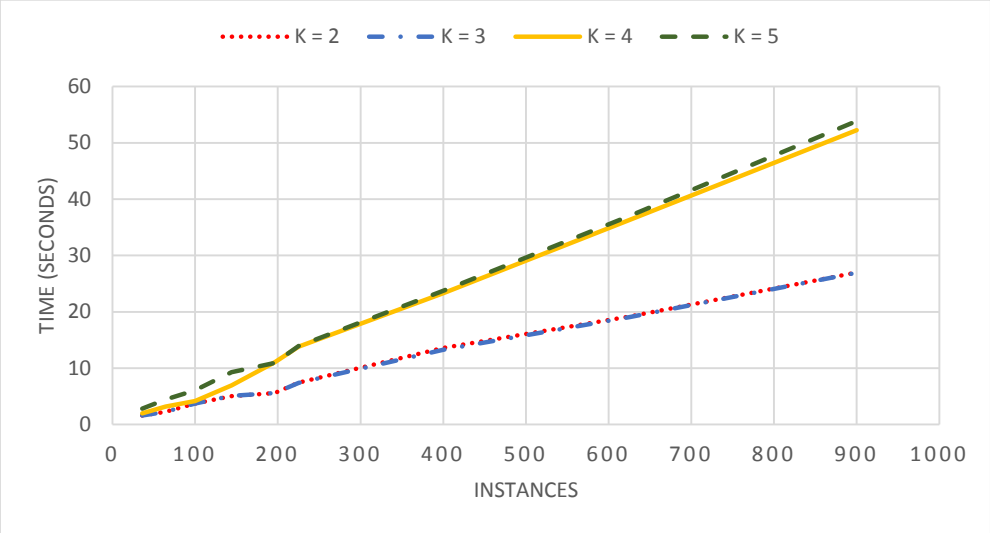


Figure 5.8: Computation time of CLDEH with respect to the number of regions and levels

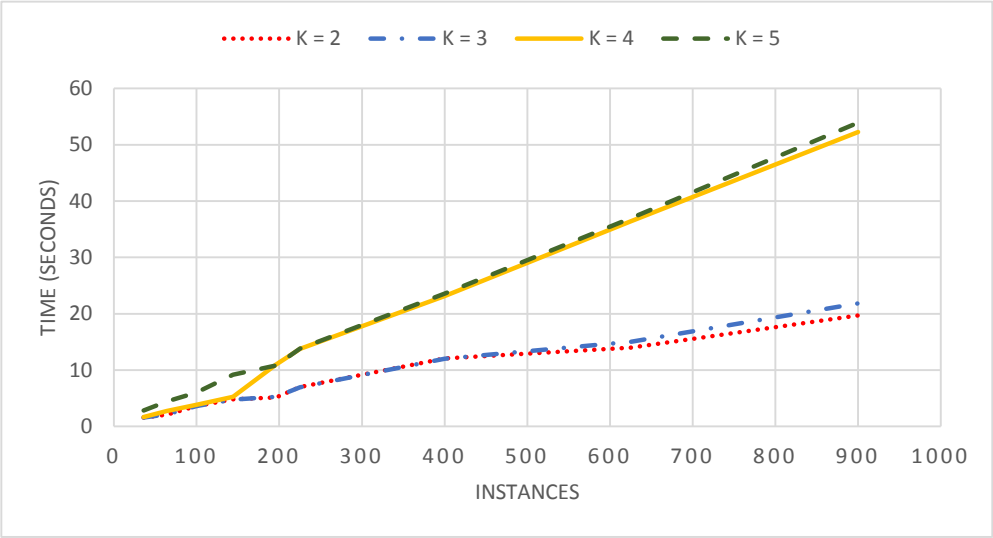


Figure 5.9: Computation time of CLUDEH with respect to the number of regions and levels

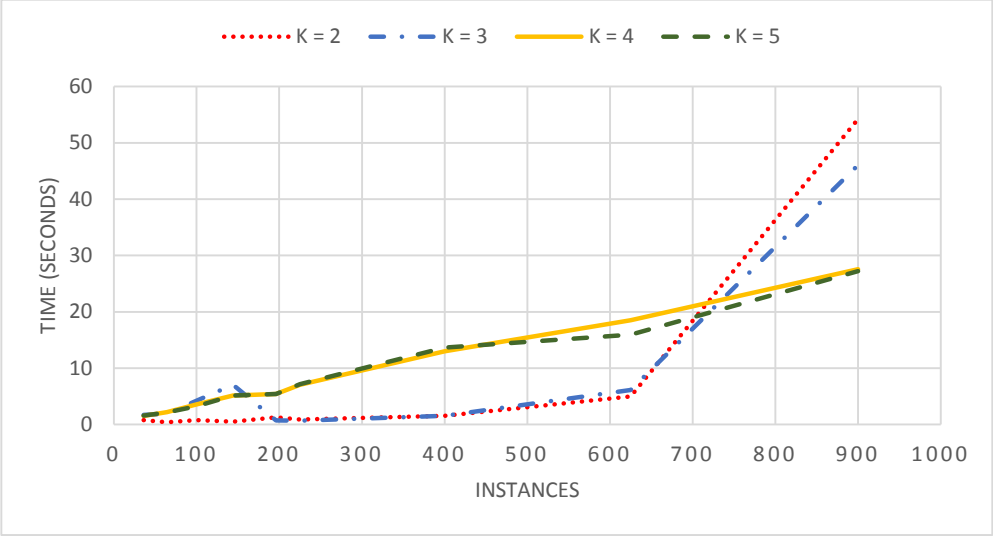


Figure 5.10: Computation time of DCLDEH with respect to the number of regions and levels

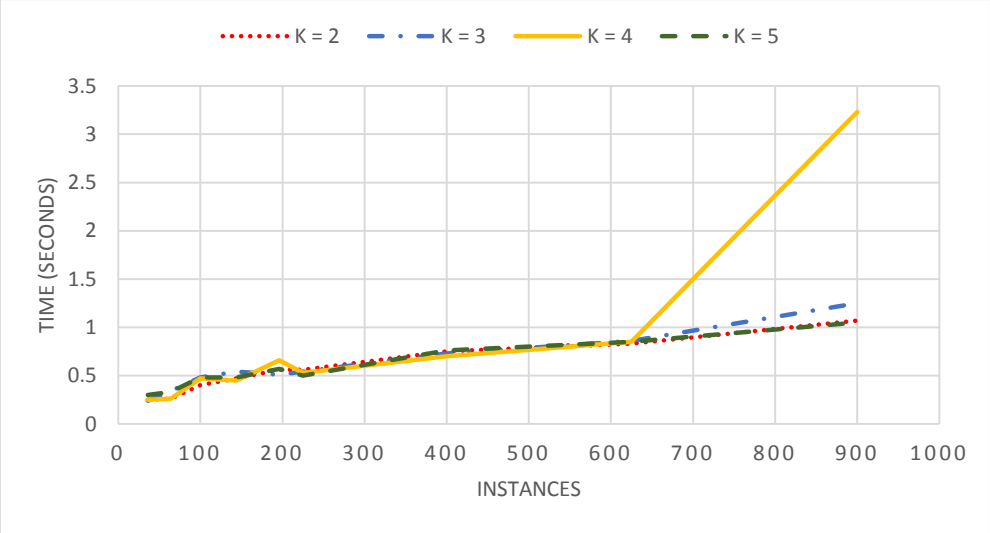
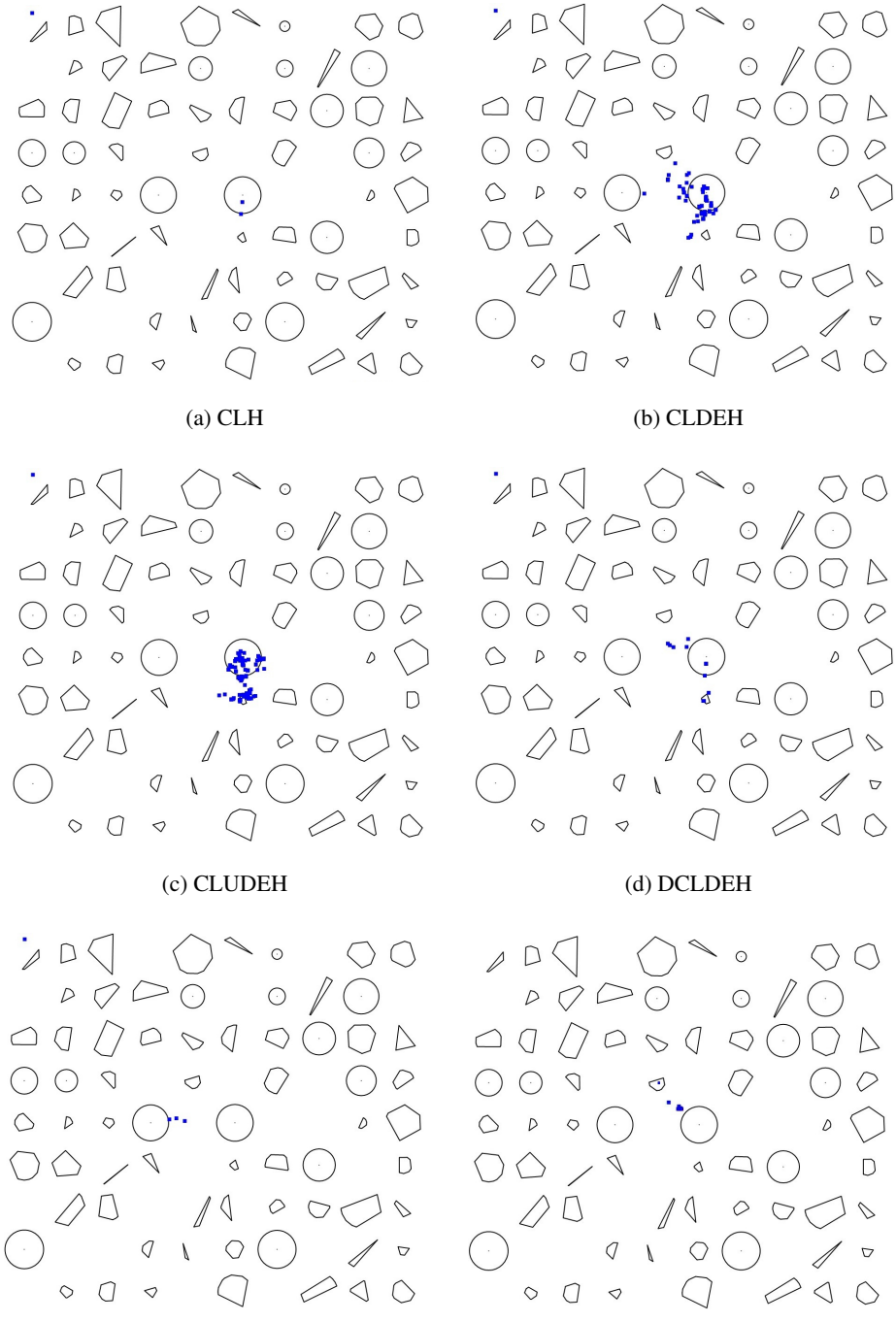


Figure 5.11: Computation time of AAH with respect to the number of regions and levels

Figure 5.12: Performance of heuristics in 100 replications of instance number 15

Finally, in the Figure 5.13, found location in each iteration for all heuristics is shown. This provides good view about the dispersion of obtained solutions for each heuristic.



(a) CLH (b) CLDEH

(c) CLUDEH (d) DCLDEH

(e) LLH (f) AAH

Figure 5.13: Facility location found in each iteration of heuristics for instance number 10

5.2 Computational study of the multi-facility case

5.2.1 Instance generation

Similar to the single facility version, some random instances are generated for multi-facility case to evaluate performance of the exact and heuristic solutions on the problem. These randomly generated instances have the same properties with the instances of the single facility case which was explained in the Section 5.1.1. In the multi-facility version, another variable is added to the problem which represents the number of facilities. This increases the size of problem even for the few number of demand regions. Hence, the instances with 9 and 16 regions are considered here as well. However, the instances with large number of demand regions are removed.

7 sets of instances are proposed for the multi-facility case with 9, 16, 36, 64, 100, 225 and 400 demand regions. Furthermore, each of these sets is tested for 2, 3, 4 and 5 facilities and 3 distinct levels of 2, 3 and 4. Consequently, 84 instances are generated and evaluated for the multi-facility case. The summary of the problem instances are listed in Table 5.21.

Table 5.21: Random Generated Instances

Instance	# of regions	# of facilities	# of levels
1	9	2	2
2			3
3			4
4		3	2
5			3
6			4
7		4	2
8			3
9			4
10		5	2
11			3
12			4
13	16	2	2
14			3
15			4
16		3	2
17			3
18			4
19		4	2
20			3
21			4
22		5	2
23			3
24			4
25	36	2	2
26			3
27			4
28		3	2
29			3
30			4
31		4	2
32			3
33			4
34		5	2
35			3
36			4
37	64	2	2
38			3
39			4
40		3	2
41			3
42			4
43	64	4	2
44			3
45			4
46		5	2
47			3
48			4
49	100	2	2
50			3
51			4
52		3	2
53			3
54			4
55		4	2
56			3
57			4
58		5	2
59			3
60			4
61	225	2	2
62			3
63			4
64		3	2
65			3
66			4
67		4	2
68			3
69			4
70		5	2
71			3
72			4
73	400	2	2
74			3
75			4
76		3	2
77			3
78			4
79		4	2
80			3
81			4
82		5	2
83			3
84			4

5.2.2 Parameter settings of the heuristics

The structure of the heuristics for the multi-facility case was described in the Section 4.2. The first heuristic gives values for the binary variables to make problem non-integer. It solves SOCP formulation to get new starting points and iterates this procedure until reaching the stopping conditions. Hence, it has some parameters related to the stopping conditions which should be set. The other heuristics are the extension of the single facility case. They have two phases. In the first phase, they convert the multi-facility problem to the several single facility problems and utilize the heuristics of the single facility case to update starting points. In the second phase, they find the value of the corresponding heuristic solution of that iteration. They repeat this algorithm while stopping conditions are not reached. First phase represents solving single facility heuristics and the related parameters are set same as the Section 5.1.2. Consequently, only stopping conditions for them should be set here.

Two stopping conditions are defined for all heuristics of the multi-facility case. The first one is maximum number of iterations and the second one is convergence criterion. As it was mentioned in the Section 4.2, convergence criterion for all heuristics is getting 3 successive enough close solutions. All heuristics of the multi-facility case begin form some initial locations and with a particular procedure update locations for the next iterations. They repeat this algorithm and after several iterations converge to a specific solution. Accordingly, they get same solutions after several iterations. Setting maximum number of iterations less than required replications for converging, decrease the quality of the solution. For the first heuristic (LALH) 3 maximum number of iterations is tested on 6 randomly generated instances. These values are 10, 15 and 20. Based on the results of the Table 5.22, it is clear that around 15 iterations is enough for converging to the heuristic solution. Considering the maximum number of iterations as 10, decreases the running time for some instances but affects the quality of the solution as well. Since the computation time difference between 10 and 15 number of iterations is negligible, 15 is accepted as the maximum number of iterations. For convergence condition of the LALH, 3 values are tested which are 0.01, 0.001 and 0.0001. Table 5.23 indicates the results of the convergence condition evaluation for LALH. As it was mentioned above, heuristics of the multi-facility

Table 5.22: LALH parameter setting

Number of demand regions	Number of facilities	Number of levels	Max number of iterations = 10		Max number of iterations = 15		Max number of iterations = 20	
			MIN		MIN		MIN	
			Obj. value	Running time (s)	Obj. value	Running time (s)	Obj. value	Running time (s)
36	3	2	280.19	1.93	280.19	1.81	280.19	1.88
	4	3	185.82	4.16	185.82	4.22	185.82	4.09
64	3	2	325.98	2.11	325.75	2.39	325.75	2.36
	4	3	189.83	9.59	189.83	9.53	189.83	9.57
100	3	2	296.12	5.01	296.12	5.11	296.12	5.22
	4	3	192.38	13.65	191.75	15.46	191.75	15.37

Table 5.23: LALH parameter setting

Number of demand regions	Number of facilities	Number of levels	Convergence condition = 0.01		Convergence condition = 0.001		Convergence condition = 0.0001	
			MIN		MIN		MIN	
			Obj. value	Running time (s)	Obj. value	Running time (s)	Obj. value	Running time (s)
36	3	2	280.19	1.96	280.19	1.94	280.19	1.94
	4	3	185.82	4.39	185.82	4.21	185.82	4.16
64	3	2	325.75	2.28	325.75	2.34	325.75	2.28
	4	3	184.82	9.42	184.82	9.53	184.82	9.57
100	3	2	296.12	5.2	296.12	5.09	296.12	5.19
	4	3	189.23	15.33	189.23	15.41	189.23	15.37

case converge to a specific solution after several iterations quickly and then, they report same solutions for the next iterations. The difference between heuristic solution value of an iteration just before convergence and the heuristic solution after convergence is more than 1. Also, the difference between heuristic solutions of the iterations after convergence is zero. Consequently, all values in the interval of (0, 1) show similar behavior on the heuristic as the convergence value. Considering the convergence value more than one affects the quality of the heuristic. As a result, similar to the single facility case, 0.001 is set as convergence value of LLH.

Next heuristics are the extended version of heuristics of the single facility case for the multi-facility case. Since the repetition procedure of the all extended heuristics for new obtained starting locations are the same, they show similar behavior on the stopping conditions. Consequently, the parameter settings for only LAL-CLDEH is

Table 5.24: LAL-CLDEH parameter setting

Number of demand regions	Number of facilities	Number of levels	Max number of iterations = 10		Max number of iterations = 15		Max number of iterations = 20	
			MIN		MIN		MIN	
			Obj. value	Running time (s)	Obj. value	Running time (s)	Obj. value	Running time (s)
36	3	2	280.19	20.51	280.19	20.59	280.19	22.13
	4	3	189.2	18.69	187.97	22.37	187.97	22.49
64	3	2	327.26	25.83	325.75	30.78	325.75	30.7
	4	3	189.36	31.72	189.36	31.61	189.36	31.99
100	3	2	296.12	51.98	296.12	51.9	296.12	51.93
	4	3	184.26	45.32	184.26	45.34	184.26	45.41

provided in this section and parameters of the other heuristics are considered same as the LAL-CLDEH.

3 values are evaluated as the maximum number of iterations for LAL-CLDEH which are 10, 15 and 20. The obtained results of these values are listed in the Table 5.24. Considering maiximum number of iterations as 15 provides less than or equal heuristic values in comparison with 10 and 20 maximum number of iterations. The computation time of all three cases are close to each other. Therefore, 15 is selected as the maximum number of iterations for LAL-CLDEH. Similar to the convergence value of the LALH, considering any convergence value of the LAL-CLDEH between 0 and 1 provides the same results for heuristic and getting large values affects the quality of the heuristic. Consequently, 0.001 is selected as the convergence value of LAL-CLDEH as well (See Table 5.25).

5.2.3 Computational results

In this section the proposed model for the multi-facility multi-level minimax location problem is solved by both MISOCP formulation (exact method) and introduced heuristics. Then, they are compared with each other to evaluate their performance on the randomly generated instances.

Table 5.25: LAL-CLDEH parameter setting

Number of demand regions	Number of facilities	Number of levels	Convergence condition = 0.01		Convergence condition = 0.001		Convergence condition = 0.0001	
			MIN		MIN		MIN	
			Obj. value	Running time (s)	Obj. value	Running time (s)	Obj. value	Running time (s)
36	3	2	280.19	22.14	280.19	22.25	280.19	22.13
	4	3	185.91	22.52	185.91	22.49	185.91	22.54
64	3	2	325.65	30.7	325.73	30.7	325.75	30.72
	4	3	189.36	31.94	189.36	31.96	189.36	32.03
100	3	2	296.12	51.9	296.12	51.81	296.12	51.84
	4	3	184.26	45.41	184.26	45.37	184.26	45.42

5.2.3.1 Exact solution

The MISOCP formulation for the multi-facility multi-level continuous minimax location problem was presented in the Section 3.2.1. It contains binary variables and the branch-and-bound technique is utilized by the solver to deal with integer variables. Increasing number of binary variables will growth branch-and-bound tree and out of memory error will appear. Multi-facility case has t times more integer variables in comparison with the single facility version where t is the number of facilities. Accordingly, solving small instances in this case can be troublesome as well. Instances number 45 and 57 was run for 48 hours but can not report the optimal solution in this time limit. Again, 3600 seconds time limit is considered for the problem to be able for solving all 84 instances by MISOCP formulation. The associated results of the MISOCP for the randomly generated instances in the multi-facility case are listed in the Tables 5.26 and 5.27. It can be seen in the Tables 5.26 and 5.27 that objective function has downward trend by increasing number of facilities.

Table 5.26: Optimal or the best found objective function values by MISOCP formulation and associated running times (in seconds)

Problem instance	Number of regions	Number of facilities	Number of levels	Objective function value	Running time (s)	Gap (%)	Best solution
1*	9	2	2	321.11	0.56	0	321.11
2*			3	275.4	0.97	0	275.4
3*			4	214.25	0.84	0	214.25
4*		3	2	211.83	1.01	0	211.83
5*			3	194.4	1.34	0	194.4
6*			4	163.64	3.18	0	163.64
7*		4	2	182.94	1.62	0	182.94
8*			3	161.18	2.07	0	161.18
9*			4	120.37	3.34	0	120.37
10*		5	2	160.94	2.36	0	160.94
11*			3	135.49	3.59	0	135.49
12*			4	100.77	3.76	0	100.77
13*	16	2	2	345.62	2.23	0	345.62
14*			3	307.84	3.03	0	307.84
15*			4	228.92	5.66	0	228.92
16*		3	2	294.4	3.17	0	294.4
17*			3	238.82	9.97	0	238.82
18*			4	167.38	15.26	0	167.38
19*		4	2	188.05	10.84	0	188.05
20*			3	166.57	18.28	0	166.57
21*			4	144.51	141.34	0	144.51
22*		5	2	169.43	19.06	0	169.43
23*			3	149.22	65.54	0	149.22
24*			4	127.6	645.58	0	127.6
25*	36	2	2	340.59	5.47	0	340.59
26*			3	277.53	16	0	277.53
27*			4	233.94	114.28	0	233.94
28*		3	2	280.19	15.97	0	280.19
29*			3	227.28	52.44	0	227.28
30*			4	184.87	1378.67	0	184.87
31*		4	2	204.32	76.88	0	204.32
32*			3	172.1	267.69	0	172.1
33			4	169.81	3600	75.4	154.76
34*		5	2	192.44	655.63	0	192.44
35			3	155.98	3600	19.3	155.98
36			4	148.41	3600	82.4	140.89
37*	64	2	2	354.87	18.38	0	354.87
38*			3	287.96	85.21	0	287.96
39*			4	217.98	319.84	0	217.98
40*		3	2	325.75	51.77	0	325.75
41*			3	264.3	307.62	0	264.3
42			4	210.44	3600	80.5	205.39

Table 5.27: Optimal or the best found objective function values by MISOCP formulation and associated running times (in seconds)

Problem instance	Number of regions	Number of facilities	Number of levels	Objective function value	Running time (s)	Gap (%)	Best solution
43*	64	4	2	211.14	406.25	0	211.14
44*			3	184.82	2617.84	0	184.82
45			4	205.88	3600	92.2	158.19
46		5	2	199.86	3600	1.8	198.49
47			3	178.9	3600	84	171.17
48			4	203.21	3600	92.8	147.72
49*	100	2	2	356.05	169.02	0	356.05
50*			3	289.13	272.66	0	289.13
51			4	248.87	3600	26.3	240.57
52*		3	2	296.12	424.73	0	296.12
53			3	241.67	3600	2.5	240.23
54			4	238.17	3600	94.9	193.58
55*		4	2	222	1718.6	0	222
56			3	209.81	3600	88.6	184.26
57			4	227.16	3600	94.5	165.15
58		5	2	206.71	3600	85.5	195.86
59			3	188.14	3600	87.2	167.54
60			4	219.32	3600	94.4	148.12
61*	225	2	2	362.35	1441.07	0	362.35
62*			3	294.04	1663.09	0	294.04
63			4	326.43	3600	87.3	236.66
64		3	2	326.54	3600	13.5	321.25
65			3	314.77	3600	74.8	259.95
66			4	303.87	3600	97.3	203.63
67		4	2	264.01	3600	92.7	227.55
68			3	337.43	3600	95.1	187.34
69			4	355.15	3600	97.7	173.11
70		5	2	292.08	3600	94.5	208.54
71			3	348.91	3600	95.3	171.81
72			4	357.22	3600	97.7	150.53
73	400	2	2	375.2	3600	0.8	373.35
74			3	311.97	3600	15.8	302.97
75			4	395.65	3600	97.3	253.87
76		3	2	344.01	3600	76.8	337.63
77			3	389.23	3600	94.7	251.99
78			4	386.52	3600	98.6	205.68
79		4	2	357.24	3600	95.7	231.85
80			3	368.23	3600	96.7	190.33
81			4	379.25	3600	98.5	178.47
82		5	2	400.79	3600	96.2	208.62
83			3	383.47	3600	96.9	176
84			4	397.15	3600	98.6	160.31

5.2.3.2 Heuristic's solutions

The seven proposed heuristics of the multi-facility case are tested on the randomly generated instances in this section. The obtained results of these heuristics are compared with results of the MISOCP formulation to evaluate effectiveness of our heuristics. Since all of the proposed heuristics are depended on the starting points, each instance is solved with 20 distinct initial locations and the percent deviation of the results of these heuristics from the best found solution are obtained. For those instance which the optimal solution can not be achieved in the specified time limit, minimum value between objective function of the MISOCP solution with some gap and the values of the heuristic solutions is considered as the best found solution of that instance. The best solutions of these heuristics for all instances which have the minimum percent deviation between 20 replications are reported in the Tables 5.28 and 5.29. Furthermore, average of the percent deviations for these replications are provided in the Tables 5.30 and 5.31. In the last column of these tables the percent deviation of the problem's objective function for 20 initial locations is reported. This helps to figure out how much the initial locations are improved by utilizing the heuristics on them. As it was expected, the LAL-AAH again outperforms the other heuristics. However, in some of the instances non of the heuristics can find the optimal solutions. This indicates the advantage of MISOCP formulation specially in small instances.

For providing the comprehensive comparison between heuristics of the multi-facility case, the computation time of them is also presented. Tables 5.32 and 5.33 indicate the computation time of the all seven heuristics in seconds associated to the solutions of the Tables 5.28 and 5.29. As it was mentioned above, the MISOCP could not solve instances 45 and 57 in 48 hours. However, these heuristics solve them at most in 34 and 45 seconds, respectively. Similar to the single facility case, LAL-AAH heuristic again has the lowest computation time between all others.

In the following, the presented results of the heuristics are displayed on some figures to provide a good view for the performance of each heuristic. Figures 5.14 and 5.14 illustrate the minimum percent deviation of heuristics and the Figure 5.16 shows the associated computation time of the previous tables. Finally, Figures 5.17 and 5.18 provide the obtained locations of the facilities by each heuristic and compare them

Table 5.28: Minimum % deviation of the heuristics from the best found solution

Problem Instance	LALH	LAL - CLH	LAL - LLH	LAL - CLDEH	LAL - CLUDEH	LAL - DCLDEH	LAL - AAH	Initial locations
1*	0	0	0	0	0	0	0	17.51
2*	0	0	0	0	0	0	0.02	19.95
3*	0	7.41	0	7.41	0	7.41	0	31.33
4*	0	0	0	0	8.63	0	0	50.88
5*	0	0	0	0	0	0	0	60.72
6*	0	14.08	0	0	0	0	0.56	58.05
7*	0	8.69	0	0	0	8.69	0	50.08
8*	0.26	0	0	0	0	0	0	41.88
9*	0.13	0	0	5.14	5.14	0	0	89.99
10*	1.3	0	0	0	0	0	0	45.99
11*	5.44	0	1.04	0	0	0	0	60.43
12*	0	0	0.92	0	0	0	0	114.02
13*	0	0	0	0	0	0	0	21.56
14*	0	1.06	0	0	0	0	0	10.77
15*	0	7.83	0	0	0	7.81	0.89	38.95
16*	0	0	0	0	0	0	0.09	16.6
17*	0	3.35	0	0	0	0	0	22.09
18*	0	13.77	7.14	7.96	1.97	7.96	5.47	37.33
19*	0.03	4.23	5.32	1.24	4.23	4.23	0	40.36
20*	0	17.67	5.05	6.99	9.98	5.05	0.67	35.17
21*	8.39	15.83	7.22	5.7	1.07	7.22	3.56	39.58
22*	3.14	2.89	8.09	8.48	0	6.82	3.14	55.07
23*	3.59	8.32	3.83	1.64	1.64	4.02	2.37	42.81
24*	4.56	26.67	13.49	4.56	13.49	4.56	0.4	52.41
25*	0	0	0	0	0	0	0	28.87
26*	0	0	0	0	0	0	0	28.37
27*	0.56	3.95	0	0	0.56	0.56	0	20.07
28*	0	0	0	0	0	0	0	28.76
29*	0	0	0	0	0	0	0.01	28.81
30*	8.88	13.61	14.09	5.98	4.32	6.64	6.31	28.63
31*	0	8.6	0	4.11	4.11	4.11	0.23	43.11
32*	7.97	11.28	14.76	9.22	11.24	9.22	0.01	37.83
33	4.26	14.55	10.16	0	6.08	5.99	4.37	37.92
34*	0	2.98	8.1	1.48	0	0.18	0	42.6
35	6.92	6.8	4.89	2.99	7.76	6.85	3.01	42.7
36	0	10.73	0	0	5.32	3.39	0.89	22.35
37*	0	0	0	0	0	0	0	30.16
38*	0.18	0.18	0.18	0	0.18	0.18	0.18	30.43
39*	8.09	14.21	21.44	1.42	3.17	1.37	0	31.47
40*	0	0	0	0	0	0	0	12.24
41*	0	0	0	0	0	0	0	12.26
42	0	0.66	0.66	1.95	4.87	3.91	0	14.8

Table 5.29: Minimum % deviation of the heuristics from the best found solution

Problem Instance	LALH	LAL - CLH	LAL - LLH	LAL - CLDEH	LAL - CLUDEH	LAL - DCLDEH	LAL - AAH	Initial locations
43*	0	0	0.76	0.45	0	0	0	41.07
44*	2.84	6.35	3.09	2.59	0	0.13	0.13	30.91
45	6.2	12.08	13.22	4.2	0.61	4.62	0	26.25
46	3.75	0.67	1.79	0	3.73	3.98	3.99	39.48
47	4.87	6.89	2.93	0	2.71	4.67	4.52	31.2
48	5.77	8.54	11.72	0.51	0	5.79	0.72	30.04
49*	0	0	0	0	0	0	0.02	25.86
50*	0	0	0	0	0	0	0	25.8
51	3.65	4.17	0	0.27	2.71	2.58	0.51	25.1
52*	0	0	0	0	0	0	0.02	21.53
53	0	2.83	0	0	0	0	0.01	21.56
54	5.7	10.6	6.49	2.8	0	7.88	0.95	15
55*	0	0	1.52	0.64	0	0	0	42.66
56	4.06	0.11	3.78	0	0.11	0	1.13	39.45
57	3.02	7.14	8.13	0	3.87	1.24	1.02	18.55
58	6.34	6.38	0	6.38	5.67	6.38	6.41	45.66
59	2.39	3.62	2.55	0	4.24	3.28	1.09	38.13
60	6.97	7.74	12.89	1.15	7.74	2.03	0	21.44
61*	0	0	0	0	0	0	0	28.85
62*	0	0	0	0	0	0	0	28.88
63	6.34	13.2	10.4	1.85	5.59	4.55	0	16.52
64	3.84	0	0	0	0	0	0	16.08
65	4.12	1.4	0	1.39	1.4	1.39	0.27	16.41
66	5.81	8.3	5.64	0.76	5.59	4.01	0	16.78
67	0.38	0	0.58	0.58	0.58	0.58	0.59	33.07
68	4.04	2.75	2.78	0	2.36	1.61	1.12	31.13
69	7.84	10.07	7.58	2.47	6.52	2.21	0	15.06
70	0	1.37	2.04	2.96	2.96	2.96	2.96	41.32
71	1.22	9.62	0	2.22	1.75	3.74	3.76	39.15
72	7.41	11.86	16.17	2.29	5.1	2.19	0	25.97
73	0.5	0.44	0	0.44	0.44	0.44	0.44	24.86
74	0.5	0.5	0	0.5	0.44	0.44	0.44	24.9
75	4.04	4.04	4.04	1.4	4.04	2.01	0	4.55
76	1.02	0	0	0	0	0	1.37	13.33
77	5.86	10.04	1.07	8.31	8.31	0	0.02	23.22
78	7.25	6.22	8.27	0.92	4.63	1.03	0	16.25
79	0.16	0.16	0.49	0	0	0.49	0.49	28.96
80	1.6	2.09	2.09	0.91	1.52	0.91	0	27.45
81	7.22	7.82	9.63	0.27	5.2	0.23	0	14.83
82	1.81	5.58	0	5.81	5.81	5.58	2.68	41.33
83	0.77	2.77	0.58	1.51	2.11	0.95	0	35.9
84	7.38	8.55	9.97	2.13	4.02	0	0.89	15.16
Average	2.362	4.801	3.293	1.571	2.304	2.191	0.806	32.08
Number of best found solutions	3	4	4	4	4	4	4	0

Table 5.30: Average % deviation of the heuristics from the best found solution

Problem Instance	LALH	LAL - CLH	LAL - LLH	LAL - CLDEH	LAL - CLUDEH	LAL - DCLDEH	LAL - AAH	Initial locations
1*	3.88	8.7	3.59	7.4	6.15	7.92	4.35	45.18
2*	7.99	16.29	7.38	6.22	13.21	15.58	7.43	48.07
3*	21.52	17.28	20.42	14.43	12.1	17.2	8.84	79.65
4*	15.89	25.83	19.41	16.75	19.7	27.77	15.59	82.27
5*	19.89	36.2	20.76	31.86	32.74	31.06	19.7	79.37
6*	23.99	43.91	42.05	23.97	23.11	37.06	22.77	101.79
7*	12.34	32.32	14.75	22.75	18.83	33.06	9.4	88.12
8*	15.06	28.94	15.98	28.25	27.37	29.16	12.02	94.14
9*	31.52	54.52	31.5	22.24	29.61	55.59	30.81	145.26
10*	15.52	32.96	23.51	32.17	22.78	33.53	15.1	95.64
11*	18.16	40.36	31.59	40.05	38.11	41.04	17.8	112.46
12*	34.18	48.88	57.34	32.48	30.19	51.51	30.73	175.59
13*	6	17.41	5.14	9.05	7.95	11.55	5	47.72
14*	5.71	14.74	6.26	4.33	6.13	7.26	3.21	36.19
15*	19.04	34.35	31.87	16.24	20.78	21.94	13.07	60.39
16*	5.01	9.42	5.84	2.98	1.22	8.36	5.75	43.94
17*	10.37	19.17	11.09	4.5	6.03	13.8	7.02	45.11
18*	26.17	35.17	32.6	28.63	26.78	29.62	16.24	78.89
19*	26.1	38.17	33.76	26.21	34.77	40.02	34.88	83.41
20*	28.82	39.28	34.2	28.4	33.81	39.46	29.05	75.81
21*	25.96	38.56	26.95	29.78	32.38	32.21	16.71	84.21
22*	32.51	35.44	35.99	34.56	33.28	33.58	32.95	93.81
23*	32.53	42.17	34.45	39.91	36.81	44.26	31.7	84.27
24*	31.32	41.92	55.74	34.51	36.27	39.47	20.45	93.33
25*	0.89	8.43	4.98	3.74	4.06	4.14	3.89	53.11
26*	5.23	20.37	7.15	5.1	6.07	6.11	5.43	52.54
27*	16.96	21.64	22.28	6.27	11.3	13.08	5.92	46.61
28*	11.78	16.09	13.76	11.35	13.23	14.01	14.07	53.06
29*	13.67	21.41	15.76	11.23	10.61	15.15	14.34	53.15
30*	20.84	25.25	30.09	16.36	16.35	19.08	15.92	50.68
31*	18.66	30.11	24.55	18.9	23.49	24.42	23.69	79.44
32*	26.12	35.2	28.91	23.21	28.62	27.52	24.56	73.78
33	24.83	32.7	31.39	19.51	21.5	26.06	16.75	56.62
34*	10.62	29.53	26.33	22.41	19.82	21.81	19.89	81.44
35	24.41	35.88	30.93	29.73	24.79	30.67	23.9	81.64
36	32.73	44.74	42.53	28.96	31.52	36.65	25.59	59.48
37*	4.9	7.87	7.48	5.59	6.53	6.26	6.14	54.4
38*	5.67	20.69	7.84	6.51	5.74	7.17	7.32	54.74
39*	32	32.31	35.28	11.29	16.79	14.66	8.45	53.11
40*	2.36	2.09	3.17	2.11	1.99	2.14	2.04	40.9
41*	3.41	6.85	3	1.1	4.52	2.78	2.55	40.96
42	25.32	24.74	34.81	10.77	14.92	17.32	12.1	38.09

Table 5.31: Average % deviation of the heuristics from the best found solution

Problem Instance	LALH	LAL - CLH	LAL - LLH	LAL - CLDEH	LAL - CLUDEH	LAL - DCLDEH	LAL - AAH	Initial locations
43*	25.27	29.82	26.55	20.68	18.68	26.32	27.91	85.08
44*	22.72	30.63	26.53	19.27	19.86	23.21	21.05	71.81
45	23.46	28.86	32.51	13.89	16.8	17.49	11.68	50.03
46	18.2	26.69	24.68	18.61	21	25.52	25.82	87.1
47	15.61	29.75	24.16	14.83	16.29	22.7	18.55	76.06
48	19.54	18.51	27.34	10.53	13.34	19.81	10.89	51.27
49*	0.97	9.1	4.08	2.01	1.16	2.85	2.81	51.11
50*	3.77	14.93	5.29	1.82	1.97	3.47	4.42	51.06
51	21.01	19.87	21.22	6.75	10.81	9.79	5.7	41.97
52*	7.83	10.47	9.99	8.16	8.16	11.59	10.16	52.59
53	15.17	20.84	16.15	12.48	13.9	15.65	15.1	52.66
54	16.7	20.55	21.47	8.79	12.19	13.69	7.75	39.35
55*	15.7	19.9	21.18	14.03	12.69	16.32	16.11	66.5
56	16.09	23.44	20.23	13.32	12.31	16.91	14.69	62.79
57	19.04	19.55	21.9	9.57	11.4	13.69	8.66	39.91
58	17.15	23.14	27.38	19.26	19.25	24.12	22.78	80.44
59	16.25	24.85	23.4	14.68	14.49	15.89	15.48	71.17
60	18.46	19.03	24.76	9.57	14.64	14.66	10.81	46.29
61*	1.39	7.32	5.77	5.54	5.54	5.65	5.78	52.23
62*	1.66	12.68	7.24	5.55	5.44	5.81	5.78	52.3
63	19.08	19.25	19.57	7.15	15.91	12.24	5.65	38.77
64	5.65	6.47	7.18	6.38	6.51	6.86	6.88	42.88
65	6.03	7.74	6.36	6.19	6.61	6.49	6.59	43.32
66	13.49	15.75	16.99	3.44	9.9	8.93	4.05	31.02
67	16.25	19.73	17.52	15.52	17.24	19.36	21.28	70.9
68	20.46	25.26	17.36	13.1	15.39	20.1	17.82	68.46
69	15.05	17.18	17.9	6.83	12.19	10.73	5.53	32.15
70	9.24	22.57	22.83	19.15	19.13	21.99	21.26	79.74
71	9.81	25.62	21.49	17.09	17.65	21.09	19.49	77.05
72	19.51	22.84	27.79	9.96	16.12	13.89	8.26	42.62
73	1.03	5.2	5.14	4.94	4.94	5.24	5.21	51.6
74	2.36	14.55	4.53	4.01	5.21	6.07	5.26	51.66
75	13.32	14.21	16.15	3.75	10.77	7.33	3.02	26.69
76	11.25	12.46	13.99	12.2	11.2	12.49	13.15	40.27
77	11.41	13.75	12.38	12.12	12.12	12.04	12.29	52.56
78	14.99	17.25	16.91	4.05	9.22	7.55	3.61	29.23
79	19.4	22.31	22.98	17.91	18.34	22.41	22.57	71.25
80	20.72	23.49	20.7	17.68	17.77	21.07	21.66	69.3
81	12.21	14.99	15.98	4.06	8.24	7.64	4.34	26.8
82	11.26	20.59	23.29	21.53	21.27	21.97	22.06	83.92
83	13.13	18.05	19.56	16.76	17.83	18.27	18.33	76.91
84	11.57	16	17.05	5.45	10.09	10.27	5.11	32.51
Average	15.75	23.25	20.57	14.65	16.02	18.93	13.79	63.33

Table 5.32: Running time (in seconds) to obtain solutions of Tables 5.28 and 5.29

Problem Instance	LALH	LAL - CLH	LAL - LLH	LAL - CLDEH	LAL - CLUDEH	LAL - DCLDEH	LAL - AAH
1	0.14	1.42	0.75	6.25	5.71	1.35	0.04
2	0.17	2.77	0.93	5.91	6.31	2.7	0.04
3	0.27	2.87	1.11	6.11	5.96	2.04	0.04
4	0.16	2.97	0.91	6.69	7.04	2.28	0.04
5	0.23	2.51	1.11	6.43	6.51	1.73	0.04
6	0.5	2.2	1.28	7	6.73	2.46	0.04
7	0.19	2.6	1.06	7.14	8.62	2.12	0.04
8	0.31	3.17	1.23	7.96	8.33	1.9	0.04
9	0.79	2.98	1.44	8.39	7.4	2.23	0.04
10	0.23	2.86	1.15	7.35	8.77	2.46	0.04
11	0.4	1.73	1.32	7.92	7.93	2.06	0.04
12	1.29	1.92	1.55	8.73	7.61	2.81	0.05
13	0.39	3.03	1.5	12.97	10.58	6.01	0.07
14	0.6	4.19	1.92	12.32	11.32	4.94	0.07
15	1.17	3.9	2.55	11.44	11.44	6.56	0.07
16	0.72	5.16	1.69	13.5	11.74	6.64	0.07
17	1.16	4.28	2.13	14.18	14.6	4.89	0.07
18	1.66	5.21	2.74	12.58	15.35	6.1	0.07
19	1.19	5.47	1.84	14.32	12.55	6.03	0.07
20	2.02	5.95	2.31	15.14	15.2	4.88	0.08
21	2.59	5.82	2.92	12.68	13.88	5.51	0.07
22	2.25	5.92	1.99	13.73	14.49	5.73	0.07
23	2.83	6.23	2.4	15.3	15.89	4.73	0.07
24	3.74	6.37	2.95	14.8	13.14	5.24	0.08
25	0.84	1.91	2.85	19.47	12.36	7.19	0.14
26	1.51	4.29	4.22	18.16	18.63	5.27	0.14
27	1.79	6.68	5.72	16.1	18.95	8.29	0.14
28	1.93	3.49	2.92	22.13	15.34	7.31	0.15
29	3.07	5.36	4.19	20.23	21.08	9.78	0.14
30	4.52	6.39	5.92	18.66	21.78	9	0.14
31	2.76	3.15	3.08	23.4	19.09	9.41	0.13
32	4.46	8.53	4.34	22.49	21.76	10.98	0.14
33	5.54	9.84	5.91	20.07	22.43	10.5	0.14
34	4.55	5.91	3.3	25.58	22.12	9.47	0.13
35	7.02	8.04	4.47	26.45	24.75	10.42	0.14
36	10.01	10.85	7.5	21.74	26.52	12.46	0.14
37	1.32	2.01	4.1	30.18	16.42	6.23	0.24
38	3.54	3.17	6.85	28.25	27.54	6.79	0.23
39	2.55	8.43	9.05	20.04	30.34	11.09	0.23
40	2.68	4.3	4.48	30.7	20.22	10.01	0.23
41	7.78	5.06	6.7	29.31	29.65	10.53	0.23
42	6.35	9.8	8.78	25.31	30.82	13.28	0.23

Table 5.33: Running time (in seconds) to obtain solutions of Tables 5.28 and 5.29

Problem Instance	LALH	LAL - CLH	LAL - LLH	LAL - CLDEH	LAL - CLUDEH	LAL - DCLDEH	LAL - AAH
43	4.72	3.79	4.47	33.78	23.86	12.07	0.24
44	9.45	8.01	6.61	31.99	30.37	12.81	0.23
45	12.36	11.53	8.94	28.59	33.21	16.64	0.24
46	8.73	5.48	4.91	35.75	26.62	12.51	0.22
47	15.26	11.22	6.77	35.24	32.64	14.62	0.23
48	17.45	12.73	8.92	32.29	34.45	16.76	0.23
49	2.12	2.23	6.83	44.57	23.82	10.25	0.35
50	3.6	6.82	11.84	42.97	38.98	7.04	0.35
51	4.36	17.21	16.51	31.56	43.13	24.9	0.35
52	5.87	2.91	6.77	51.93	27.21	9.2	0.35
53	9.63	7.83	11.63	47.5	44.51	17.13	0.35
54	11.03	13.21	15.66	32.43	52.09	17.06	0.35
55	10.21	4.29	6.77	48.06	29.17	10.87	0.34
56	16.6	8.62	11.3	45.41	41.54	18.32	0.36
57	17.89	14.76	15.33	34.19	44.13	19.99	0.35
58	19.89	5.62	7.07	48.8	33.71	15.45	0.34
59	27.65	9.05	11.24	49.05	45.77	21.47	0.35
60	25.43	15.08	15.5	40.16	46.45	22.16	0.34
61	5.64	2.65	14.34	97.51	52.69	10.55	0.82
62	10.06	7.5	22.03	90.58	90.11	12.11	0.82
63	15.25	16.88	37.78	53.57	91.44	17.33	0.83
64	12.37	3.55	19.87	110.92	57.2	12.83	0.8
65	42.65	6.07	26.32	107.42	105.97	20	0.81
66	29.31	16.42	33.81	70.76	111.95	18.87	0.81
67	46.7	3.45	15.95	103.24	53.54	15.37	0.8
68	43.8	7.71	27.89	98.26	84.09	22.31	0.81
69	68.21	17.04	35.78	68.74	105.55	31.07	0.8
70	73.14	4.97	15.25	104.57	56.47	18.11	0.78
71	110.59	10.79	24.78	94.49	84.77	28.39	0.8
72	107.2	24.81	40.58	66.45	106.34	35.68	0.79
73	24.85	4.86	25.8	169.69	83.86	21.94	1.52
74	325.32	7.38	65.07	158.34	168.46	15.19	1.53
75	32.72	22.21	55.69	102.21	171.05	19.51	1.54
76	33.41	6.9	29.25	200.29	100.45	19.94	1.49
77	70.62	7.7	43.71	190.09	189.61	21.92	1.49
78	59.09	22.02	54.84	102.98	197.76	24.44	1.51
79	87.23	5.07	28.93	188.07	98.89	21.87	1.47
80	153.97	6.72	50.1	153.45	142.68	25.5	1.48
81	105.01	29.13	56.08	114.92	183.31	41.36	1.47
82	158.46	5.6	27.61	176.65	101.98	21.48	1.45
83	317.67	14.41	48.52	160.71	153.72	30.42	1.46
84	253.45	28.86	58.32	134.41	175.3	56.56	1.45

with the solution of the MISOCP formulation.

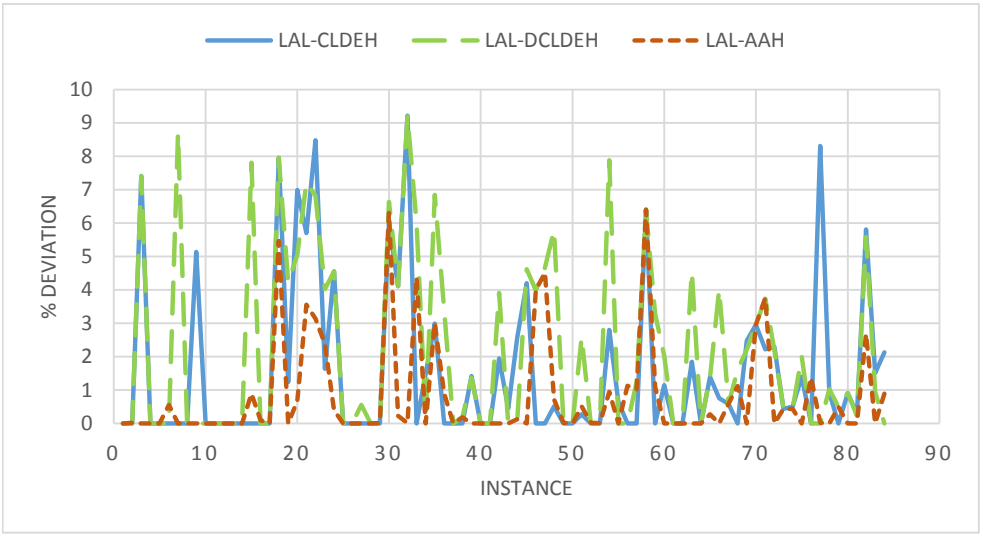


Figure 5.14: Minimum % deviation of LAL-CLDEH, LAL-DCLDEH and LAL-AAH for all instances

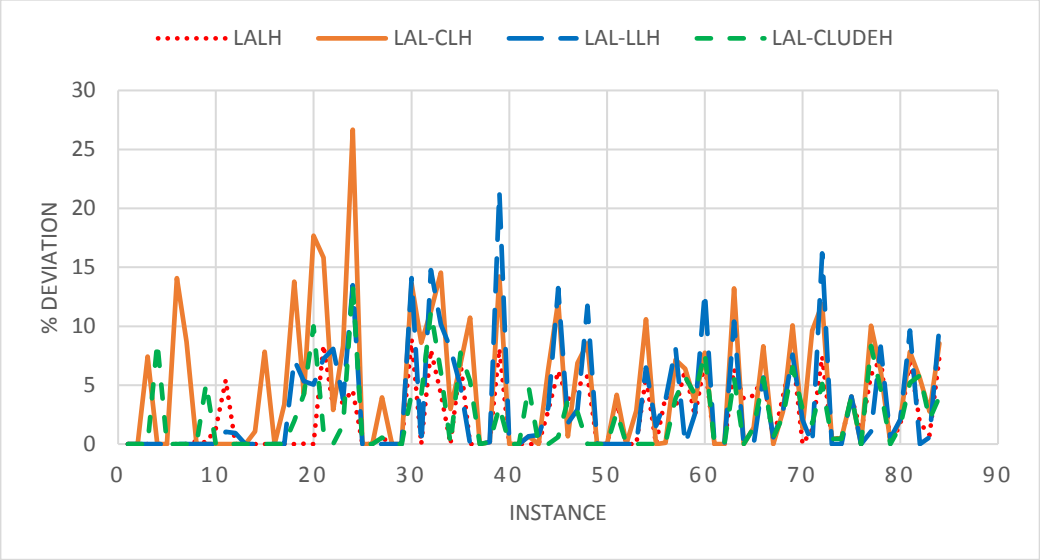


Figure 5.15: Minimum % deviation of LALH, LAL-CLH, LAL-LLH and LAL-CLUDEH for all instances

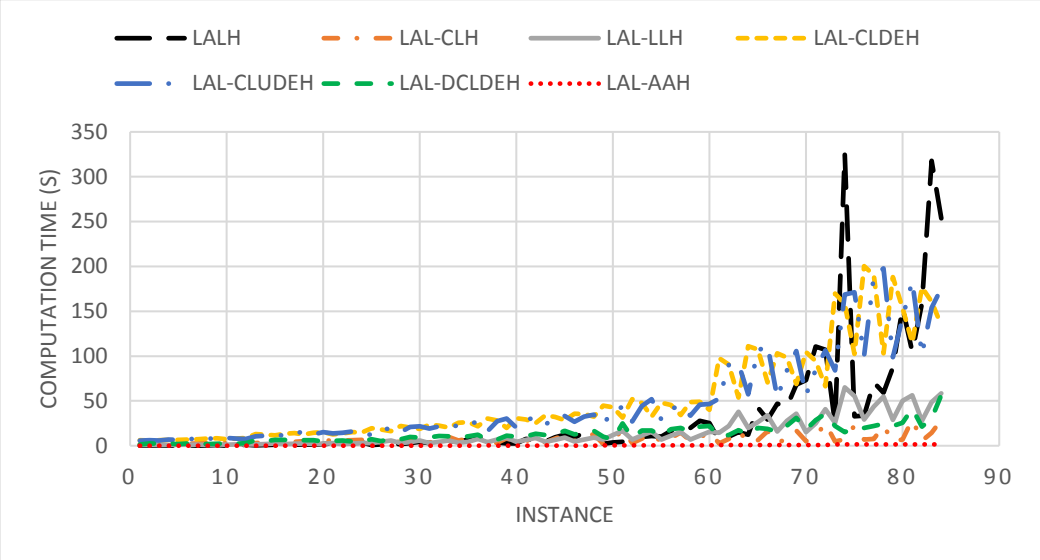


Figure 5.16: Comutation time of LALH (in seconds) for all instances

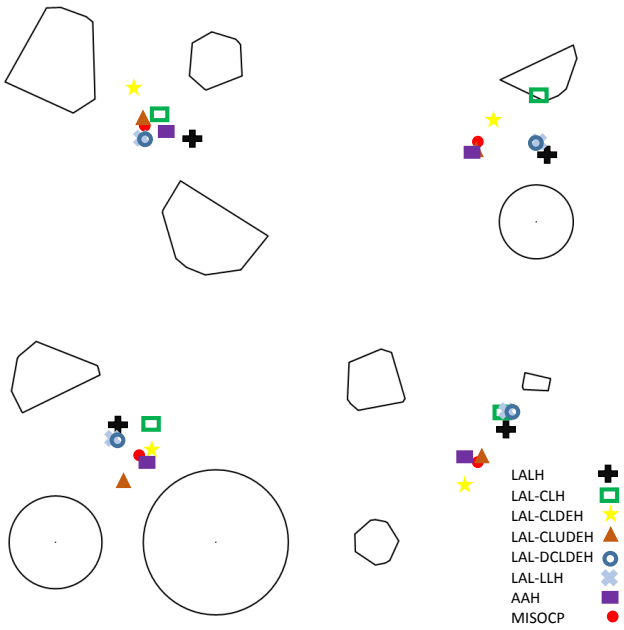


Figure 5.17: Facility locations found with the heuristics for the instance number 21

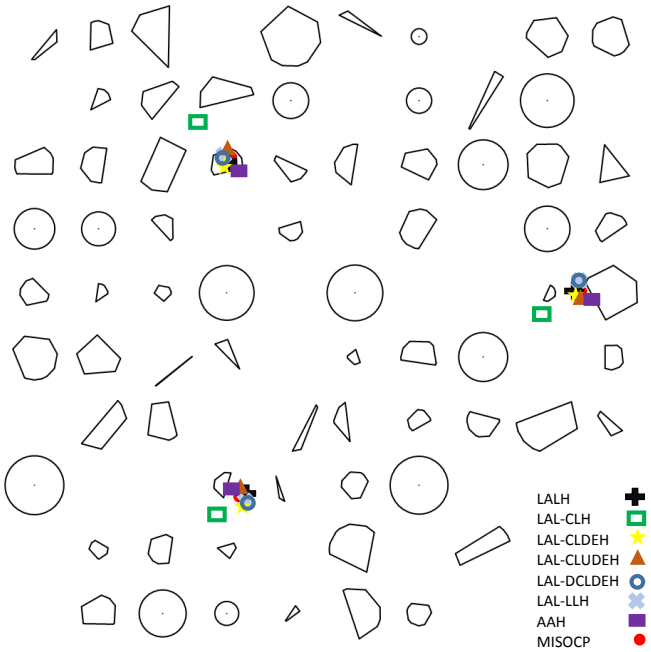


Figure 5.18: Facility locations found with the heuristics for the instance number 53

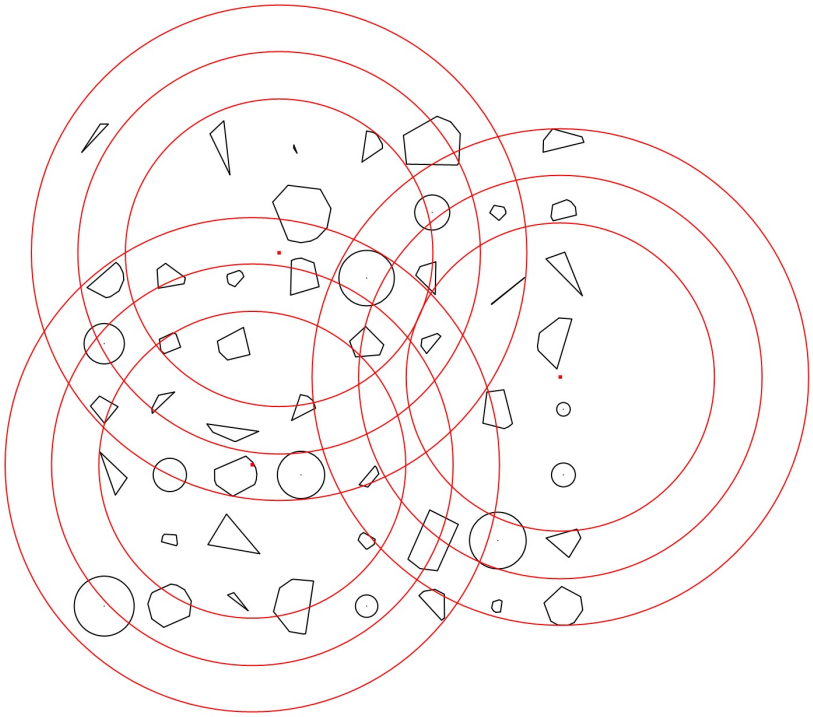


Figure 5.19: Covering levels for the instance number 41

CHAPTER 6

CASE STUDY OF ISTANBUL

Computational results of the proposed multi level continuous minimax location problem were presented for some randomly generated instances in the previous chapter. For evaluating the performance of our model more precisely, we apply this model for the real case of Istanbul. Pre-positioning of relief items in humanitarian logistics was explained in Section 1. Here, we aim at finding the optimal facility location(s) to pre-position some relief items in Istanbul. Based on the nature of the problem, it is required to cover all population. Hence, all districts pulls the facilities to themselves. The problem is finding the locations such that relief items could be provided for all inhabitants with some constraints. Accordingly, the problem can be formulated by minimax model. Istanbul city contains 38 districts with total population of 14.8 million people. We considered each district as a demand region and its population as corresponding weight of that region. These regions and their associated weights are listed in the Table 6.1. Districts have irregular shapes with uncountable number of vertices. For reducing the complexity of the problem, all 38 districts (demand regions) are estimated by polygons with maximum number of 70 corners and the associated coordinates for each corner is found (See Figure 6.1). ArcGIS (ArcMap) version 10.0 software is utilized to digitize of the Istanbul map. Since taking the convex hulls of the demand regions do not change the optimal solution, the convex hulls of the demand regions are considered in calculations. Other parameters of the problem are set same as in Section 5. The single facility case is studied with four scenarios where the number of levels are 2, 3, 4, or 5. In the multi-facility case, 12 scenarios are considered which are the combinations of 2, 3, 4, 5 facilities and 2, 3, and 4 levels. Again, the time limit for running of the MISOCP formulation for both

single and multi-facility cases is considered as 3600 seconds.

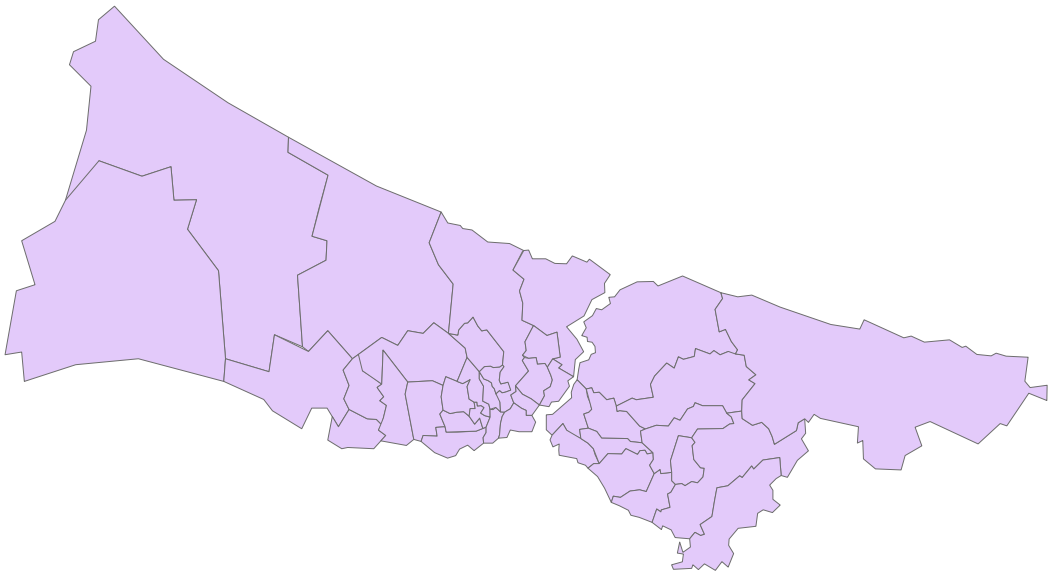


Figure 6.1: Districts map of Istanbul

Based on above mentioned configurations, firstly, the model is applied for the single facility case and solved with both MISOCP formulation and proposed heuristics. For the SOCP based heuristics, each scenario is solved with 20 different initial locations and percent deviations from the best solution is found. The minimum and average of percent deviations from the best solution for each scenario is listed in Tables 6.2 and 6.3, respectively. Additionally, the Table 6.4 compares the computation time of the MISOCP formulation and heuristics' solution related to Table 6.2.

Results of the single facility multi-level minimax location problem for Istanbul case again confirm the quality of the Area Abstraction Heuristic (AAH) which has lower average and minimum percent deviation than the other heuristics. Also, Table 6.2 indicates that considering 3 covering levels for the problem reduces objective function by almost 13 Km in comparison with having 2 covering levels.

In Figure 6.2, obtained locations for the classic or one-level (green circle), bi-level (red star) and three-level (blue square) single facility minimax location problem are shown. This figure indicates the effect of increasing the number of covering levels on the location of the facility. It should be mentioned that the objective function for one-level minimax problem in this case is 82846.65 meters which is significantly greater

Table 6.1: Districts of Istanbul

District	Name	Population	Weight
1	Catalca	68935	0.005
2	Silivri	170523	0.012
3	Buyukcekmece	237185	0.016
4	Arnavutkoy	247507	0.017
5	Basaksehir	369810	0.025
6	Esenyurt	795010	0.054
7	Beylikduzu	297420	0.02
8	Avcilar	430770	0.029
9	Kucukcekmece	766609	0.052
10	Bakirkoy	222437	0.015
11	Bahcelievler	598097	0.04
12	Bagcilar	751510	0.051
13	Gungoren	298509	0.02
14	Esenler	457231	0.031
15	Bayrampasa	273148	0.018
16	Zeytinburnu	287897	0.019
17	Fatih	417285	0.028
18	Beyoglu	238762	0.016
19	Gaziosmanpasa	499766	0.034
20	Sultangazi	525090	0.036
21	Eyup	377650	0.026
22	Kagithane	439685	0.03
23	Besiktas	189356	0.013
24	Sisli	272803	0.018
25	Sariyer	342753	0.023
26	Beykoz	250410	0.017
27	Uskudar	535537	0.036
28	Kadikoy	452302	0.031
29	Atasehir	422513	0.029
30	Umraniye	694158	0.047
31	Cekmekoy	239611	0.016
32	Maltepe	490151	0.033
33	Kartal	459298	0.031
34	Sancaktepe	377047	0.025
35	Sultanbeyli	324709	0.022
36	Pendik	691681	0.047
37	Tuzla	242232	0.016
38	Sile	34241	0.002
Sum		14789638	1

Table 6.2: Objective function of the MISOCP formulation and minimum of the % deviation of heuristics from best found solution for single facility case

Number of regions	Number of levels	MISOCP	Heuristics					
			CLH	LLH	CLDEH	CLUDEH	DCLDEH	AAH
38	2	40349.99	1.35	0	1.24	1.35	1.24	0
	3	37152.59	8.15	0	2.01	7.81	2.01	0
	4	34983.38	4.82	0	2.25	3.95	3.95	0.71
	5	34377.86	2.55	4.46	0	0	0	0
Average			4.218	1.115	1.375	3.278	1.8	0.178

Table 6.3: Objective function of the MISOCP formulation and average of the % deviation of heuristics from best found solution for single facility case

Number of regions	Number of levels	MISOCP	Heuristics					
			CLH	LLH	CLDEH	CLUDEH	DCLDEH	AAH
38	2	40349.99	11.13	1.3	1.26	1.45	1.26	0
	3	37152.59	19.33	4.29	3.54	8.04	3.09	0
	4	34983.38	22.44	19.73	4.13	4.17	4.17	0.71
	5	34377.86	22.55	12.16	0.13	0	0	0
Average			18.83	9.37	2.265	3.415	2.13	0.178

Table 6.4: Comparison between the computation time (in second) of the MISOCP formulation and heuristics' solution related to Table 6.2.

Number of regions	Number of levels	MISOCP	Heuristics					
			CLH	LLH	CLDEH	CLUDEH	DCLDEH	AAH
38	2	0.14	0.2	0.29	2.54	2.82	3.64	0.03
	3	0.172	0.22	0.45	3.67	3.53	4.11	0.03
	4	0.249	0.22	0.68	3.48	3.8	4.16	0.03
	5	7.07	3.04	1.04	3.95	4.17	4.29	0.05

than the optimal objective function values of the 2 and 3 level cases.

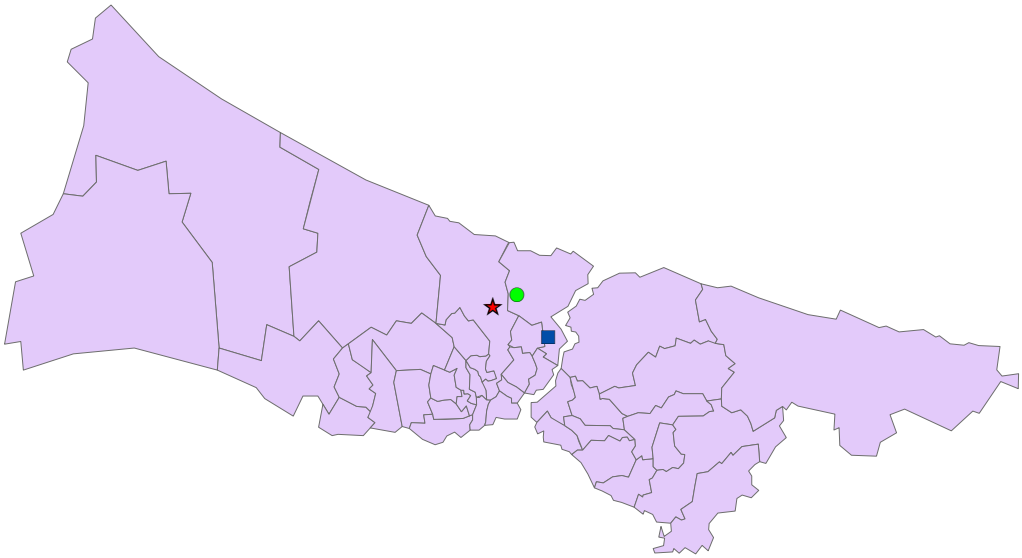


Figure 6.2: Single facility locations found for single-level (green circle), bi-level (red star) and three-level (blue square) cases of Istanbul

In the following, MISOCP formation and proposed heuristics of the multi-facility version are applied for the Istanbul case. Similar to the single facility case, all heuristics are run with 20 distinct initial locations and the solution with lowest heuristic solution between these replications is selected as the best solution of that heuristic. Minimum percent deviation of these replications and the average of them are listed in Tables 6.5 and 6.6, respectively. Also, the obtained result of the MISOCP formulation is presented in these tables to provide a comparison between performance of the MISOCP and heuristics. The percent deviations are calculated based on the best found solutions. All instances with less than or equal 4 number of facilities are solved optimally in the time limit of 3600 seconds. Hence, the best found solution for them is the solution obtained by the MISOCP formulation. For the instance of 5 facility and 2 levels, again the MISOCP solution with a gap outperforms the heuristics and has the better solution. For the last two cases, the best solution is obtained by the LAL-CLDEH and the LAL-AAH heuristics, respectively.

Similar to the results of Section 5.2.3.2, Table 6.5 shows that the LAL-AAH heuristic has better results than other heuristics. The LAL-CLDEH can find the best solu-

Table 6.5: Objective function value of the MISOCP formulation and minimum of % deviation of heuristics from best found solution for multi-facility case

Number of regions	Number of facilities	Number of levels	MISOCP	Gap (%)	Heuristics						
					LALH	LAL-CLH	LAL-LLH	LAL-CLDEH	LAL-CLUDEH	LAL-DCLDEH	LAL-AAH
38	2	2	27946.76	0	2.3	8.73	1.15	0	2.47	6.79	0.02
		3	25187.79	0	3.17	3.66	0	2.99	3.59	3.66	0.68
		4	23589.51	0	0.92	10.76	8.09	8.76	0	4.27	0
	3	2	23083.59	0	1.73	6.64	0.93	1.9	0.26	0.26	0.13
		3	21605.49	0	9.23	4.35	10.49	0	6.08	3.28	0
		4	20452.08	0	3.67	10.68	10.68	2.17	4.05	9.11	8.46
	4	2	19835.89	0	0.39	13.6	3.72	2.51	13.5	6.69	1.72
		3	19077.94	0	11.42	18.81	17.72	5.43	14.84	6.72	0.01
		4	18524.47	0	5.85	15.09	12.89	0	3.99	3.99	6.64
	5	2	17231.56	23.6	9.31	10	11.7	3.04	1.77	6.33	0.96
		3	17906.01	40.36	6.54	8.08	5.83	0	3.84	1.24	0.38
		4	17660.41	50.89	1.08	11.64	0.15	3.29	2.98	4.54	0
Average					4.63	10.17	6.95	2.51	4.78	4.74	1.58

Table 6.6: Objective function value of the MISOCP formulation and average of % deviation of heuristics from best found solution for multi-facility case

Number of regions	Number of facilities	Number of levels	MISOCP	Gap (%)	Heuristics						
					LALH	LAL-CLH	LAL-LLH	LAL-CLDEH	LAL-CLUDEH	LAL-DCLDEH	LAL-AAH
38	2	2	27946.76	0	11.4	20.17	15.94	11.45	10.03	17.87	5.75
		3	25187.79	0	7.81	24.83	27.01	16.18	17.59	18.2	8.84
		4	23589.51	0	23.63	33.59	38.04	23.11	17.77	22.49	9.53
	3	2	23083.59	0	33.37	33.84	38.75	21.38	23.1	24.07	14.65
		3	21605.49	0	22.77	32.87	39.94	21.25	26.32	19.67	17.62
		4	20452.08	0	29.33	40.44	52.42	19.57	24.36	23.15	20.04
	4	2	19835.89	0	42.35	36.69	41.31	29.18	30.53	38.19	28.36
		3	19077.94	0	48.47	28.51	37.75	21.21	28.1	21.63	22.51
		4	18524.47	0	28.61	31.95	48.81	22.78	28.64	21.85	26.58
	5	2	17231.56	23.6	30.3	47.78	56.97	40.21	43.05	46.09	12.37
		3	17906.01	40.36	35.73	32.67	45.1	33.01	29.03	29.33	33
		4	17660.41	50.89	34.03	33.51	48.95	37.24	36.58	26.07	28.9
Average					28.98	33.07	40.92	24.71	26.26	25.72	19.01

Table 6.7: Computation time (in second) comparison between the MISOCP formulation and heuristics' solution related to the Table 6.5.

Number of regions	Number of facilities	Number of levels	MISOCP	Heuristics						
				LALH	LAL-CLH	LAL-LLH	LAL-CLDEH	LAL-CLUDEH	LAL-DCLDEH	LAL-AAH
38	2	2	16.66	20.13	3.45	4.59	38.72	36.34	21.85	0.96
		3	55.95	22.47	5.07	7.22	38.26	37.57	25.34	0.97
		4	41.18	27.61	11.22	11.21	42.75	39.43	33.68	0.97
	3	2	98.72	19.42	3.95	4.67	42.33	39.49	19.29	0.95
		3	120.26	23.79	4.33	7.56	42.05	40.68	22.91	0.97
		4	551.05	31.05	8.25	11.04	43.89	39.89	28.48	0.98
	4	2	885.81	22.74	5.06	4.76	43.79	39.67	23.79	0.98
		3	1774.26	28.11	6.18	8.29	43.79	42.93	27.22	0.97
		4	2742.99	30.06	7.62	11.48	45.6	43.54	28.1	0.98
	5	2	3600	24.62	5.69	5.08	44.53	38.61	23.16	1.08
		3	3600	25.89	6.17	8.86	45.24	41.69	24.82	1.02
		4	3600	29.36	6.97	11.45	46.06	45.32	26.22	1.03

tion one time more than the LAL-AAH but the average of percent deviation for all instances in LAL-CLDEH is worse. Furthermore, Table 6.6 confirms the better performance of the LAL-AAH in comparison with the LAL-CLDEH as well. Table 5.2.3.2 also indicates the effect of increasing covering levels in reducing the optimal objective function value of the problem. For example, in the instance with 2 facilities and 2 covering levels, increasing the number of covering levels by 2 has the same influence on the objective function with increasing number of facilities by one.

Another aspect in comparison of the heuristics with each other is their computation time. Table 6.7 displays the required computational time for solving the MISOCP formulation and heuristics.

Considering the computation time of the heuristics, it can be claimed that the LAL-AAH heuristic not only has the better solutions, but also has the lower computation time. It can solve instances of the Istanbul case almost in 1 second. However, the computation time for solving the MISOCP formulation for the instances with 5 facilities is more than 3600 seconds. Furthermore, the LAL-AAH also outperforms other heuristics in terms of the computation time.

Finally, the found facility locations in 3-facility problem have been compared for the single level (green circles), bi-level (Red stars) and three-level (blue squares) versions in order to see the effects of the proposed model on the locations of the facilities for

the Istanbul case.

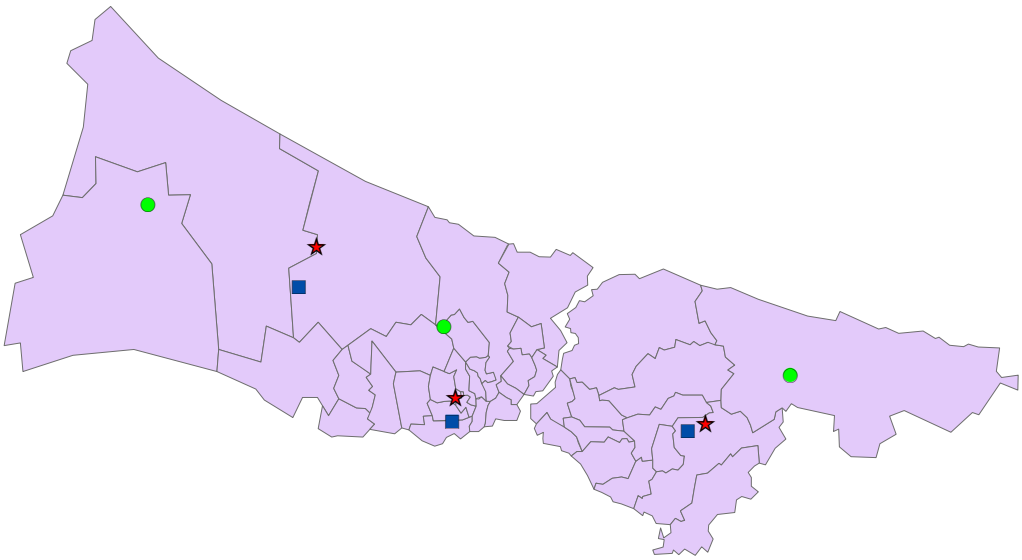


Figure 6.3: Multi facility locations found for single-level(Green), bi-level(Red) and three-level(Blue) cases of Istanbul

CHAPTER 7

CONCLUSION AND FUTURE STUDY DIRECTIONS

Facility location problem aims at optimal placement of the facilities in order to satisfy requirements of the corresponding demands. The common objective function in location problems is minimizing the costs (especially transportation costs) which usually can be measured by distance or required traveling time from the facilities to the demands. Accordingly, some location models seek to minimize the maximum distance (Minimax problem), some others consider minimizing sum of the weighted distance (Minisum problem), several of them focus on maximizing the coverage area with a specific covering distance (Covering problem) and etc. Furthermore, there are 3 specific distance measures which provide another classification for location problems; Euclidean distance, Rectilinear distance and Chebyshev distance.

In this thesis, we consider the pre-positioning problem in humanitarian logistic as a case study which is a public and social oriented problem. Thus, we are applying Minimax model with Euclidean distance. As we explained in the Section 2, the facility location problem is studied as discrete and continuous problem. In the continuous facility location problem, the facilities can be located all over the plane. Furthermore, the demands are distributed continuously in the plane. The common approach in modeling of such a problem is abstracting the demands into some points on the plane. For obtaining more realistic solutions for the problem, the demands are considered as regions instead of points. An important question for this assumption is how to measure the distance between the facilities and the demand regions. 3 distance measures are introduced by Drezner and Wesolowski [20] which are the distance of the closest point of the region to the facility, the distance of the farthest point and average

distance. According to the nature of our problem, the Euclidean distance between the facility and the farthest point of the demand region is utilized to take the worst-case scenario into account. Additionally, the multi-level coverage is considered for the Minimax problem in this thesis. In the classic version of the Minimax location problem, even unimportant remote demands pull the facilities toward themselves and increase the distance between other demands and facilities. The multi-level Minimax location problem considers several coverage levels and the facilities cover specific percentage of the demand regions in each level. It ensures covering of the important demand regions in the inner levels and others in the outer levels. Therefore, the maximum distance between demand regions and facilities in the inner levels is lower than the classic version. Consequently, we are applying Euclidean distance for studying the continuous minimax location problem by regional demand with several covering levels.

It was mentioned in Section 3.1.3 that the coefficient and domain of the first level are considered as 1 and 0, respectively. It should be reminded that the covering percentage of the last level (P_K) is always equal 1, since the last level provides the total coverage for the problem. Also, the coefficients of levels (c_h), domains of levels (d_h) and their percentages (P_h) are monotonically increasing by moving to the outer levels. This ensures greater covering range for outer levels in comparison with inner levels. The values of these parameters are considered randomly in this thesis. However, the decision maker can obtain these parameters by testing the model with different configurations or based on the previous experiences.

Firstly, we propose the single facility case of the problem and give an MISOCP formulation of the problem which is an exact method. Since the formulation can not solve the big size instances in the reasonable time, some heuristics are introduced for the problem. Then, a multi-facility extension is considered. In this case, the MISOCP formulation can not solve even medium size problems. Hence, new heuristics are proposed for the multi-facility case. Heuristics provide acceptable solutions in the reasonable time but the results of the MISOCP formulation in smaller size instances are better than heuristics. Therefore, providing of the MISOCP formulation of the problem is beneficial.

Lack of the similar studies make us to generate several random instances to evaluate our proposed heuristics. In the single facility case, the AAH heuristic has the best results between all heuristics. It finds the exact or best found solution in 23 instances of 36 total instances. Also, the average of the percent deviations from the best solution for all 36 instances is 0.102 which is lower that all others. Additionally, it has the lowest computational time after the CLH heuristic. The second best heuristic is DCLDEH which has almost 13 times greater average computational time in comparison with the AAH. In the multi-facility case, in addition to the extensions of the single facility case heuristics, one new heuristic is introduced. Again, the extension of the AAH for the multi-facility case (LAL-AAH) outperforms other heuristics. It has the average percent deviation of 0.806 between all 84 instances. Furthermore, it has the lowest running time between 7 proposed heuristics of the multi-facility case.

Finally, the model is tested for the real case of Istanbul to evaluate its performance more precisely. There are 38 districts in the Istanbul which are considered as the demand regions. Also, the population of each district is considered as its corresponding weight. The multi-level minimax location problem for Istanbul case confirms the results of the random generated instances. Again, in the single facility case, AAH heuristic has lower average and minimum percent deviation between other heuristics. Also, considering the results of the heuristics in the multi-facility case, it can be claimed that the LAL-AAH heuristic not only has the better solutions, but also has the lower computation time. It can solve instances of the Istanbul case almost in 1 second.

The future research directions for this problem can be as follows.

- The distance of the closest point of the region to the facility or the average distance can be applied instead of the farthest distance.
- In this thesis the demand regions are allocated to exactly one facility. It can be considered that several facilities cover a demand region simultaneously.
- Instead of random initial locations for the heuristics, some innovative methods can be proposed.

REFERENCES

[1] Milton Abramowitz, Irene A Stegun, et al. Handbook of mathematical functions. *Applied mathematics series*, 55(62):39, 1966.

[2] Burcu Balcik and Benita M Beamon. Facility location in humanitarian relief. *International Journal of Logistics*, 11(2):101–121, 2008.

[3] Charles D Bennett and Abbas Mirakhor. Optimal facility location with respect to several regions. *Journal of Regional Science*, 14(1):131–136, 1974.

[4] Oded Berman, Zvi Drezner, Dmitry Krass, and George O Wesolowsky. The variable radius covering problem. *European Journal of Operational Research*, 196(2):516–525, 2009.

[5] Oded Berman, Dmitry Krass, and Zvi Drezner. The gradual covering decay location problem on a network. *European Journal of Operational Research*, 151(3):474–480, 2003.

[6] Jack Brimberg and George O Wesolowsky. Locating facilities by minimax relative to closest points of demand areas. *Computers & Operations Research*, 29(6):625–636, 2002.

[7] Emilio Carrizosa and Frank Plastria. Polynomial algorithms for parametric minquantile and maxcovering planar location problems with locational constraints. *Top*, 6(2):179–194, 1998.

[8] R Chen. Optimal location of a single facility with circular demand areas. *Computers & Mathematics with Applications*, 41(7):1049–1061, 2001.

[9] Nicos Christofides and John E Beasley. A tree search algorithm for the p-median problem. *European Journal of Operational Research*, 10(2):196–204, 1982.

[10] Richard Church and Charles ReVelle. The maximal covering location problem. In *Papers of the Regional Science Association*, volume 32, pages 101–118. Springer, 1974.

[11] Richard L Church. The planar maximal covering location problem. *Journal of Regional Science*, 24(2):185–201, 1984.

[12] Leon Cooper. Location-allocation problems. *Operations research*, 11(3):331–343, 1963.

[13] Leon Cooper. Heuristic methods for location-allocation problems. *Siam Review*, 6(1):37–53, 1964.

[14] Georges A Croes. A method for solving traveling-salesman problems. *Operations research*, 6(6):791–812, 1958.

[15] Mark S Daskin. *Network and discrete location: models, algorithms, and applications*. John Wiley & Sons, 2011.

[16] Derya Dinler, M Tural, and C Iyigun. Location problems with demand regions. *Global Logistics Management*, page 237, 2014.

[17] Derya Dinler and Mustafa Kemal Tural. A minisum location problem with regional demand considering farthest euclidean distances. *Optimization Methods and Software*, 31(3):446–470, 2016.

[18] Zvi Drezner. The weighted minimax location problem with set-up costs and extensions. *Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle. Locational Analysis*, 25(1):55–64, 1991.

[19] Zvi Drezner and George O Wesolowsky. A new method for the multifacility minimax location problem. *Journal of the Operational Research Society*, pages 1095–1101, 1978.

[20] Zvi Drezner and George O Wesolowsky. Location models with groups of demand points. *INFOR: Information Systems and Operational Research*, 38(4):359–372, 2000.

[21] Serhan Duran, Özlem Ergun, Pınar Keskinocak, and Julie L Swann. Humanitarian logistics: advanced purchasing and pre-positioning of relief items. In *Handbook of global logistics*, pages 447–462. Springer, 2013.

[22] Abdul Aziz El-Tamimi and Khalid Al-Zahrani. An improved quadratic program for unweighted euclidean 1-center location problem. *Journal of King Saud University-Engineering Sciences*, 25(2):161–165, 2013.

[23] Jack Elzinga and Donald W Hearn. Geometrical solutions for some minimax location problems. *Transportation Science*, 6(4):379–394, 1972.

[24] Reza Zanjirani Farahani, Nasrin Asgari, Nooshin Heidari, Mahtab Hosseininia, and Mark Goh. Covering problems in facility location: A review. *Computers & Industrial Engineering*, 62(1):368–407, 2012.

[25] Henrik A Friberg. Cblib 2014: a benchmark library for conic mixed-integer and continuous optimization. *Mathematical Programming Computation*, 8(2):191–214, 2016.

[26] S Louis Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations research*, 12(3):450–459, 1964.

[27] Horst W Hamacher and Stefan Nickel. Restricted planar location problems and applications. *Naval Research Logistics (NRL)*, 42(6):967–992, 1995.

[28] Pierre Hansen, Dominique Peeters, Denis Richard, and Jacques-François Thisse. The minisum and minimax location problems revisited. *Operations Research*, 33(6):1251–1265, 1985.

[29] Zhou He, Bo Fan, TCE Cheng, Shou-Yang Wang, and Chin-Hon Tan. A mean-shift algorithm for large-scale planar maximal covering location problems. *European Journal of Operational Research*, 250(1):65–76, 2016.

[30] Donald W Hearn and James Vijay. Efficient algorithms for the (weighted) minimum circle problem. *Operations Research*, 30(4):777–795, 1982.

[31] Jianlin Jiang and Xiaoming Yuan. A barzilai-borwein-based heuristic algorithm for locating multiple facilities with regional demand. *Computational Optimization and Applications*, 51(3):1275–1295, 2012.

[32] Orhan Karasakal and Esra K Karasakal. A maximal covering location model in the presence of partial coverage. *Computers & Operations Research*, 31(9):1515–1526, 2004.

[33] DT Lee and Ying-Fung Wu. Geometric complexity of some location problems. *Algorithmica*, 1(1):193–211, 1986.

[34] Miguel Sousa Lobo, Lieven Vandenberghe, Stephen Boyd, and Hervé Lebret. Applications of second-order cone programming. *Linear algebra and its applications*, 284(1-3):193–228, 1998.

[35] Nimrod Megiddo and Arie Tamir. New results on the complexity of p-centre problems. *SIAM Journal on Computing*, 12(4):751–758, 1983.

[36] Abraham Mehrez, Zilla Sinuany-Stern, and Allan Stulman. A single facility location problem with a weighted maximin-minimax rectilinear distance. *Computers & operations research*, 12(1):51–60, 1985.

[37] Nenad Mladenović, Jack Brimberg, Pierre Hansen, and José A Moreno-Pérez. The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, 179(3):927–939, 2007.

[38] Alan T Murray, Morton E O'Kelly, and Richard L Church. Regional service coverage modeling. *Computers & Operations Research*, 35(2):339–355, 2008.

[39] Alan T Murray and Daoqin Tong. Coverage optimization in continuous space facility siting. *International Journal of Geographical Information Science*, 21(7):757–776, 2007.

[40] Alan T Murray, Daoqin Tong, and Kamyoung Kim. Enhancing classic coverage location models. *International Regional Science Review*, 33(2):115–133, 2010.

[41] KPK Nair and R Chandrasekaran. Optimal location of a single service center of certain types. *Naval Research Logistics Quarterly*, 18(4):503–510, 1971.

[42] B John Oommen. An efficient geometric solution to the minimum spanning circle problem. *Operations Research*, 35(1):80–86, 1987.

[43] Jean-Claude Picard and H Donald Ratliff. A cut approach to the rectilinear distance facility location problem. *Operations Research*, 26(3):422–433, 1978.

[44] Frank Plastria. Continuous covering location problems. *Facility location: applications and theory*, 1:37–79, 2002.

[45] Justo Puerto and Antonio M Rodríguez-Chía. On the structure of the solution set for the single facility location problem with average distances. *Mathematical programming*, 128(1):373–401, 2011.

[46] Colin R Reeves. *Modern heuristic techniques for combinatorial problems*. John Wiley & Sons, Inc., 1993.

[47] Atsuo Suzuki and Zvi Drezner. The p-center location problem in an area. *Location science*, 4(1-2):69–82, 1996.

[48] Atsuo Suzuki and Atsuyuki Okabe. Using voronoi diagrams. *Facility location: a survey of applications and methods. Springer, New York*, pages 103–118, 1995.

[49] Michael B Teitz and Polly Bart. Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations research*, 16(5):955–961, 1968.

[50] Hu Wei, Alan T Murray, and Ningchuan Xiao. Solving the continuous space p-centre problem: planning application issues. *IMA Journal of Management Mathematics*, 17(4):413–425, 2006.

[51] Endre Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal, First Series*, 43:355–386, 1937.

[52] Emo Welzl. Smallest enclosing disks (balls and ellipsoids). *New results and new trends in computer science*, pages 359–370, 1991.

[53] Sheng Xu, Robert M Freund, and Jie Sun. Solution methodologies for the smallest enclosing circle problem. *Computational Optimization and Applications*, 25(1):283–292, 2003.