HEURISTICS FOR A CONTINUOUS MULTI-FACILITY LOCATION PROBLEM WITH DEMAND REGIONS

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

DERYA DİNLER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
OPERATIONAL RESEARCH

SEPTEMBER 2013

Approval of the thesis:

HEURISTICS FOR A CONTINUOUS MULTI-FACILITY LOCATION PROBLEM WITH DEMAND REGIONS

submitted by **DERYA DİNLER** in partial fulfillment of the requirements for the degree of **Master of Science in Operational Research Department, Middle East Technical University** by,

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

Prof. Dr. Murat Köksalan
Head of Department, **Operational Research** ____________

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

Assist. Prof. Dr. Cem İyigün
Co-supervisor, **Industrial Engineering Department, METU** ____________

Examining Committee Members:

Assoc. Prof. Dr. Sinan Gürel
Industrial Engineering Department, METU ____________

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

Assist. Prof. Dr. Cem İyigün
Industrial Engineering Department, METU ____________

Assist. Prof. Dr. Serhan Duran
Industrial Engineering Department, METU ____________

Assist. Prof. Dr. Sibel Alumur Alev
Industrial Engineering Department, TOBB ETÜ ____________

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: DERYA DİNLER

Signature :

ABSTRACT

HEURISTICS FOR A CONTINUOUS MULTI-FACILITY LOCATION PROBLEM WITH DEMAND REGIONS

DİNLER, DERYA
M.S., Department of Operational Research
Supervisor : Assist. Prof. Dr. Mustafa Kemal Tural
Co-Supervisor : Assist. Prof. Dr. Cem İyigün

September 2013, 94 pages

We consider a continuous multi-facility location problem where the demanding entities are regions in the plane instead of points. Each region may consist of a finite or an infinite number of points. The service point of a station can be anywhere in the region that is assigned to it. We do not allow fractional assignments, that is, each region is assigned to exactly one facility. The problem we consider can be stated as follows: given m demand regions in the plane, find the locations of q facilities and allocate regions to the facilities so as to minimize the sum of squares of the maximum Euclidean distances of the demand regions to the facility locations they are assigned to. We assume that the regions are closed polygons as any region can be approximated within any desired accuracy with a polygon.

We first propose mathematical programming formulations of single and multiple facility location problems. The single facility location problem is formulated as a second order cone program (SOCP) which can be solved in polynomial time. The multiple facility location problem is formulated as a mixed integer SOCP. This formulation is weak and does not even solve medium-size problems. We therefore propose heuristics to solve larger instances of the problem. We develop three heuristics that work when the regions are polygons. When the demand regions are rectangles with sides parallel to coordinate axes, a special heuristic is developed. We compare our heuristics in terms of both solution quality and computational time.

Keywords: Facility Location, Minimum Sum of Squares, Second Order Cone Programming, Hyperbolic Smoothing, Min-max Problem

ÖZ

TALEP ALANLARI DÜŞÜNÜLEREK SÜREKLİ DÜZLEMDE ÇOK TESİSLİ YER SEÇİMİ PROBLEMİ İÇİN SEZGİSEL YÖNTEMLER

DİNLER, DERYA
Yüksek Lisans, Yöneylem Araştırması Bölümü
Tez Yöneticisi : Yrd. Doç. Dr. Mustafa Kemal Tural
Ortak Tez Yöneticisi : Yrd. Doç. Dr. Cem İyigün

Eylül 2013 , 94 sayfa

Bu çalışmada sürekli düzlemde çok tesisli yer seçimi problemi üzerinde çalışılmıştır. Fakat problemde talep noktaları yerine talep alanları düşünülmüştür. Her talep alanında sonlu ya da sonsuz sayıda talep noktası olabilir. Bir tesisin servis sunacağı nokta kendisine atanmış olan talep alanının herhangi bir yerinde bulunabilir. Bu çalışmada kısmi atamaya izin verilmemekte ve her talep alanı yalnızca bir tesise atanabilmektedir. Üzerinde çalıştığımız problem kısaca şu şekilde ifade edilebilir: servis sunulması gereken m tane talep alanı varken, q tane tesis yerini talep alanları ve atanmış oldukları tesisler arasındaki maksimum uzaklıkların karelerini en azlayacak şekilde bulmak ve talep alanlarını bulunan tesislere atamak. Herhangi bir alan istenen herhangi bir kesinlik seviyesinde bir poligona benzetilebildiği için bu çalışmada talep alanlarının kapalı poligonlar olduğu varsayılmıştır.

Çalışmamızda ilk olarak tek ve çok tesisli yer seçimi problemleri için matematiksel programlama formülasyonları önerdik. Tek tesisli yer seçimi problemini polinom zamanda çözülebilen ikinci dereceden konik programlama (SOCP) formülasyonu ile modelledik. Çok tesisli yer seçimi problemi için karışık tamsayılı ikinci dereceden konik programlama formülasyonu kullandık. Bu formülasyon zayıf bir formülasyondur ve orta büyüklükteki problemleri bile çözememektedir. Bu yüzden çok tesisli büyük yer seçimi problemlerini çözebilmek için sezgisel yöntemler önerdik. Talep alanları poligon olduğunda kullanabilen 3 tane sezgisel yöntem geliştirdik. Ayrıca talep alanlarının kenarları koordinat eksenlerine paralel dikdörtgensel alanlar olduğu durum için özel bir sezgisel yöntem geliştirdik. Sezgisel yöntemlerimizi hem çözüm kaliteleri hem de çözüm süreleri bakımından karşılaştırdık.

Anahtar Kelimeler: Tesis Yeri Seçimi, Enküçük Kareler Toplamı ,İkinci Dereceden Konik Programlama, Hiperbolik Düzleme, Enküçük-enbüyük Problem

To my family for their endless support...

ACKNOWLEDGMENTS

First of all I would like to thank my thesis supervisor Assist. Prof. Dr. Mustafa Kemal Tural for his support, guidance, effort and patience throughout whole thesis period. It is a pleasure to work with him. His eagerness to work made this work possible.

I also thank to Assist. Prof. Dr. Cem İyigün who motivated us to start this research. I owe thanks to Assoc. Prof. Dr. Haldun Süral for his support, attention and encouragements.

I would like to express my deepest gratitude to my family for their endless support. I cannot ever pay my debt to them.

It is pleasure to acknowledge my friends for not leaving me alone and easing my work with their great ideas and supports. Specially, I would like to thank Haluk Damgacıoğlu who kindly supported me by doing most of our joint works for his patience.

In addition, I gratefully acknowledge the committee members Assoc. Prof. Dr. Sinan Gürel, Assist. Prof. Dr. Sibel Alumur Alev and Assist. Prof. Dr. Serhan Duran for their invaluable feedbacks.

Last but not the least, I am grateful to all my professors and colleagues for their contributions to my life. I am always proud of being a member of METU-IE department both as a student and an assistant.

This study was supported by the Scientific and Technological Research Council of Turkey (TUBITAK).

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

The facility location decisions are very critical for strategic planning for both private and public firms. The facility location problems are encountered in many fields : operations research, industrial engineering, mathematics, urban planning, and geography. Most facility location problems are combinatorial in nature and challenging to solve to optimality. Location of warehouses, hospitals, retail outlets, radar beams, exploratory oil wells are some of the application areas of the facility location models.

The problem of locating q serving facilities for m demanding entities and allocating the demanding entities to the facilities so as to optimize a certain objective such as minimizing transportation cost, providing better service to customers etc. is known as the facility location problem. Facility location models can differ in their objective function, the number of the facilities to locate, the solution space in which the problem is defined, and several other decision indices. The problem is called as a discrete facility location problem if there is a finite number of candidate facility locations. If the facilities can be placed anywhere in some continuous regions, i.e. there are infinitely many candidate facility locations, then the problem is called as a continuous facility location problem. When q is equal to 1 the problem is called as a single facility location problem, otherwise it is called as a multiple facility location problem. These are the most common versions of the facility location problem. Many other versions of the problem are also studied in the literature. More details about facility location problems and in particular about different classification of facility location problems can be found in books on facility location, like [16],[19], and [14].

The two main questions of the facility location problems are:

1. Where should facilities be located?
2. Which customers should be serviced from which facility or facilities?

so as to minimize the total cost [31]. Cost is commonly measured as a function of the distance in some metric, amount of commodity transported or time between demanding entities and facilities.

In this study, we consider a continuous multi-facility location problem where the demanding entities are demand regions in the plane. For the following three cases, it would be more appropriate to represent a demanding entity as a region instead of a fixed point:

1. The size of the demanding entity may not be negligible with respect to the distances in the problem.
2. The location of the demanding entity may follow a bivariate distribution on the plane.
3. The number of demanding entities may be so large that it may be more appropriate to first cluster them into regions instead of treating each one seperately.

Consider the problem of locating a fire station that will serve forests. If each forest is represented as a fixed point by a center and a fire bursts out at an area far from the center, it may take more than the estimated time for the firefighters to reach the fire area.

Recently, Turkish military was considering to establish travelling headquarters in the South East Anatolia region of Turkey. The location of each travelling headquarter will follow a bivariate distribution. Military when deciding where to place a facility to serve these headquarters should consider the location of each headquarter as a region along with a density that represents the likelihood of the presence of the headquarter.

When demand regions are in consideration, usually there are three ways of measuring the distance between the region and the facility [25]:

- Farthest (maximum) distance,
- Closest (minimum) distance,
- Expected distance.

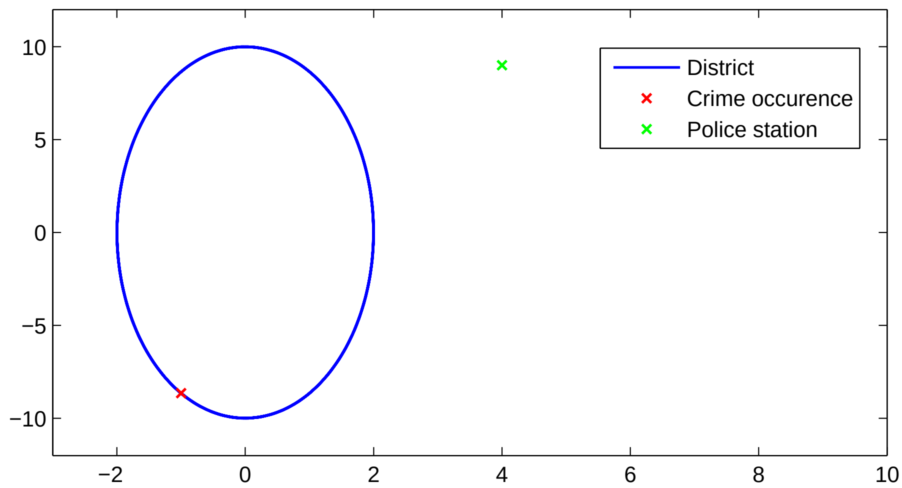

Figure 1.1: Using the maximum distance between demand region and facility location

Consider the problem of locating a police station that will serve many districts. Representation of the police station and a district can be seen in Figure 1.1. If a crime occurs on the southern west frontier of the district, the response time to this crime is directly related with the distance between the point of crime occurrence and the police station. In this work, considering the worst case scenario we choose to work with maximum distances. In our objective function we will minimize the sum of squares of the maximum Euclidean distances of the demand regions to the facility locations they are assigned to.

Using sum of squares of the distances comes from the desire of minimizing the variability. By taking squares of distances, we penalize being too far from the located facility.

It should be noted that the maximum distanced point of a region to a facility does not change if the convex hull of the demand region is taken. Therefore, we may restrict ourselves to the case where the demand regions are convex regions. As any region can be approximated in any accuracy by a polygon, we restrict ourselves to the case where each demand region is a convex polygon.

In summary, the problem we consider can be stated as follows: given m (closed) convex polygonal demand regions in the plane, find the locations of q facilities and allocate each region to exactly one facility so as to minimize the sum of squares of the maximum Euclidean distances of the demand regions to the facility locations they are assigned to.

This problem is NP-hard in general. When each demand region is a single point, the problem reduces to the minimum sum of squares clustering problem which is NP-hard in general [2]. We first show that the single facility location problem can be solved in polynomial time as it can be modeled as a second order cone program (SOCP). We then formulate the multiple facility location problem as a mixed integer SOCP. The formulation is a big-M formulation and is weak. It does not even solve medium-size problems in an hour. To be able to find good solutions for larger problem instances, we propose three heuristics. For small size instances, we compare solution quality of the heuristics with the exact solutions. We also develop a special case heuristic when the regions are of rectangular shape with sides parallel to coordinate axes. All of the three heuristics proposed for general polygonal regions can be adapted to the case in which distances are used instead of squared distances in the objective function.

First heuristic is an alternate location-allocation heuristic. We call it as the second order cone programming based alternate location allocation heuristic (SOCP-H). When the locations of the facilities are given, each region is assigned to a facility that minimizes the maximum distance between the region and the facility (allocation step). When the allocations of the regions to the facilities are known, the location algorithm solves q SOCP problems to determine the location of the facilities (location step). Starting with an arbitrary placement of the facilities, this heuristic repeats allocation and location steps until convergence is achieved.

Second heuristic is again an alternate location-allocation heuristic. We call it as max point based alternate location allocation heuristic (MP-H). It has the same allocation step with SOCP-H. It is different from SOCP-H in that only one point from each demand region is taken into consideration in the location step. When the allocations are known, updated location of a facility is computed by averaging the farthest points of the allocated regions from the previous location of the facility (location step). Again, allocation and location steps are repeated until convergence.

The mathematical modeling of the problem is nondifferentiable. Third heuristic which we call it as smoothing based heuristic (SBH) is based on a smoothing strategy which substitutes nondifferentiable functions with continuously differentiable functions. We convert the smoothed problem into an unconstrained nonlinear problem using the implicit function theorem. It is then solved with a quasi-newton algorithm that uses BFGS updating.

Our special case heuristic for rectangular regions is also an alternate location allocation heuristic. It is called as line based heuristic (LBH). It has the same allocation step with previous alternate location allocation heuristics. But its location step is quite different. When the allocations of the regions to the facilities are known, algorithm solves q single-facility location problems by converting each one into two single facility location problems on the line. The optimal solutions of these two problems on the line give the coordinates of the optimal solution of the single facility location problem on the plane. Again, allocation and location steps are repeated until convergence is achieved.

The outline of this thesis is as follows. We first discuss the variants of facility location problems in the literature in Chapter 2. In Chapter 3, we give the mathematical programming formulations of the single and multiple facility location problems in consideration. In Chapter 4, we propose heuristics for the multi-facility case. In Chapter 5, we describe how a bivariate normal distribution can be approximated by a polygonal region. Computational studies are given in Chapter 6. We conclude the study in Chapter 7.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

Facility location decisions are strategic decisions because they involve large investments and long-term plannings. They have been extensively discussed by researchers from a variety of disciplines, like geographers, marketing institutions and supply chain management specialists. This makes the constructing a comprehensive literature research very difficult. Most of the authors classify the facility location problems and their solution approaches based on the solution space (continuous space or discrete space), the number of facilities (single facility or multiple facility), the objectives of the decision maker (distance, travel time or cost), geometry of the demanding entities (demand points or demand regions). When the geometry of the demanding entities are regions, problems can be further partitioned based on the distance measure used (expected, minimum or maximum distance) [25]. More details on different versions of facility location problems and solution approaches to the problems can be seen from the facility location surveys [16], [37], [22] and [34].

The organization of this section is as follows. Discrete and continuous facility location problems and common solution approaches to these problems are discussed in Section 2.1. In Section 2.2, single and multiple facility location problems and important solution methods are given. Some of the different objectives of the decision maker in facility location problems are mentioned in Section 2.3. Finally, in Section 2.4 different types of facility location problems in terms of the geometry of demanding entites and proposed solution approaches to these problems are explained.

2.1 Classification Based on the Solution Space

The answer of the question "where will we put the facilities" determines the solution space. If the facilities can be located only at a finite list of candidate sites, then we have *discrete facility location problem*. In these problems, all canditate sites are known in advance. When the facilities can be located anywhere on the plane or anywhere on a continuous region defined in advance, we have *continuous facility location problem*. The followings are important characteristics of discrete and continuous facility location problems [37].

1. Discrete location problems are integer programming/combinatorial optimization problems since they involve binary variables.

2. Continuous location problems are generally nonlinear optimization problems.

In the following subsections, discrete and continuous facility location problems are further discussed.

2.1.1 Discrete solution space

The problem in this study is a continuous facility location problem. Therefore, solution approaches for discrete facility location problems are not mentioned. Instead of solution approaches, we give the descriptions of some discrete facility location problems.

There are eight basic problems; namely, set covering, maximal covering, p-center, p-dispersion, p-median, fixed charge, hub, and maxisum location problems [16]. Short definition of each one and some important characteristics of the problems can be seen below.

Set covering problem is to cover all of the demanding entities with minimum number of opened facilities. For each facility we know which demanding entities can be covered if it is opened and cost of opening facilities in advance. There may be costs related to serving a demanding entity from a facility. In this case, objective is to minimize the cost of siting facilities. Both versions of the problem are NP-hard.

In maximal covering problem, we have some budget or other constraints. With these constraints we want to cover most of the demanding entities. Assumption of covering all of the demanding entities in set covering problem is relaxed. This problem is also NP-hard.

In p-center problem, we want to locate p facilities to serve m demanding entities so as to minimize the maximum distance between a demanding entity and the closest facility to the demanding entity. Demanding entities may be weighted or unweighted. The complexity of the problem is $O(n^p)$. For the fixed number of facilities problem can be solved in polynomial time, but it is NP-hard in general.

p-dispersion problem only deals with the distances between facilities. In this problem, we want to locate p facilities so as to maximize the minimum distance between any pair of the facilities.

p-median problem finds the location of p facilities so as to minimize the weighted total distance between the demanding entities and the facilities. NP-hardness of this problem is similar with the p-center problem. It is NP-hard in general.

In fixed charge location problem, opening a facility has a cost and each facility generally has a capacity. This problem tries to find locations of facilities so as to minimize total facility opening and transportation costs. Because of the capacities, demanding entities may not be assigned to the closest facility.

In hub location problems, there is cost due to the interaction between facilities (hubs) in addition to the cost due to the interaction between demanding entities and facilities. This problem finds the location of facilities so as to minimize total cost of transportation between demanding entities.

Maxisum location problems are generally used for locating undesirable facilities like prisons, power plants, etc. This problem finds the location of facilities so as to maximize the total weighted distance between demanding entities and facilities.

2.1.2 Continuous solution space

In some cases, the number of candidate facility locations may be infinitely many. Facilities can be placed anywhere in the plane (possibly with some restrictions).

The most famous continuous facility location problem is the Weber Problem. In 1909 Alfred Weber asked the question of "how to locate a single facility so as to minimize the total Euclidean distance between the facility and the customers". This can be considered as the beginning of the location theory [34]. The Weber problem was later generalized, extended, reformulated and criticized many times by several authors. We investigate the original problem and solution approaches in more detail below.

The Weber problem is to find the point (x^*, y^*) so as to minimize the sum of weighted Euclidean distances between this point and m fixed points (demand points) with coordinates $(a_i, b_i), i = 1, 2, ..., m$. w_i is the weight of the i^{th} fixed point. The problem can be formulated as [29]:

$$\text{minimize}_{x,y} W(x, y) = \sum_{i=1}^{m} w_i d_i(x, y) \tag{2.1}$$

$$\text{where } d_i(x, y) = \sqrt{(x - a_i)^2 + (y - b_i)^2} \tag{2.2}$$

Here $d(x, y)$ is the Euclidean distance between points (x^*, y^*) and (a_i, b_i). It should be noted that other distance metrics like rectilinear or p-norm distances can also be used.

One common approach to solve the Weber problem is the Weiszfeld Procedure [41]. This procedure can be summarized as follows.

1. Start with an arbitrary initial solution, (x^1, y^1).

2. Update (x, y) by the following formula, obtained by setting partial derivatives of the

objective function to 0, until convergence is achieved.

$$\left(x^{(k+1)}, y^{(k+1)}\right) = \left(\frac{\sum_{i=1}^{m} \frac{w_i a_i}{d_i\left(x^{(k)}, y^{(k)}\right)}}{\sum_{i=1}^{m} \frac{w_i}{d_i\left(x^{(k)}, y^{(k)}\right)}}, \frac{\sum_{i=1}^{m} \frac{w_i b_i}{d_i\left(x^{(k)}, y^{(k)}\right)}}{\sum_{i=1}^{m} \frac{w_i}{d_i\left(x^{(k)}, y^{(k)}\right)}} \right) \tag{2.3}$$

In his work, Weiszfeld showed that his algorithm converges to the optimal solution of the problem. But he did not realize the possibility that his method might fail if the iteration falls on a fixed point. In [26], Kuhn realized that when the facility location and a fixed point coincide, the Weizfeld algorithm does not converge and does not give the optimal solution.

Some authors used the dual of the problem to find more efficient algorithms and to provide information about range of facility locations. In [20], authors give an economic interpretation of the dual. When objective is to minimize weighted sum of the distances in the primal problem, revenue maximization is the objective in dual of the problem. Therefore, dual variables can be interpreated as transportation rates if they are positive or as government support if they are negative.

Full covering, maximal covering, and empty covering problems are the basic continuous facility location problems other than the Weber problem. Short definitions of each problem can be seen below.

In full covering problem, we want to find a facility location so as to minimize the maximum distance between the demanding entities and the facility. Therefore, it is also called as a minimax location problem.

In maximal covering problem, radius of the covering circle is known. Therefore, we want to find the center of the covering circle so as to cover most of the demanding entities. In this problem, all of the demanding entities may not be covered if the radius is too small. Assumption of covering all demanding entities in full covering problem is relaxed.

Empty covering problem is used to locate an undesirable facility. We want to find the center of the largest circle which does not cover any of the fixed points. The center is the facility location.

There are multiple facility counterparts of all of the above four continuous single facility location problems.

2.2 Classification Based on the Number of Facilities

The answer of the question "how many facilities will we open" determines the classification in this section. When we want to locate only one facility, the problem is called as a single facility location problem, otherwise it is called as a multiple facility location problem. Single facility location problems are easier than the multiple facility location problems in general [13].

In the following subsections, single and multiple facility location problems are further discussed.

2.2.1 Single facility

The most famous single facility location problems are 1-center problem (discrete case) and the Weber problem (continuous case). 1-center problem is a specialization of the p-center problem which is mentioned in Section 2.1.1. The Weber problem is mentioned in detail in Section 2.1.2. Also, all problems that we have mentioned so far have both single and multiple facility versions.

Single facility location problems and solution approaches will not be mentioned further. However it should be noted that single facility location problems are the basis for the multiple facility location problems. In the location-allocation models, single facility location problems are solved as many as the number of the facilities in each iteration. For our problem, a mathematical programming formulation of the single facility case will be given in Chapter 3.

2.2.2 Multiple facilities

The most important problem in this category is the multiple facility version of the Weber problem. It has been extensively studied in the literature. Adding new facilities to the Weber problem can be done in two ways. First of them is to give interactions between each demanding entity and each facility. This is a convex optimization problem with nondifferentiable objective function. Nondifferentiable objective function is the result of the possibility that a demanding entity and a facility may coincide. In such a situation, distance between the demanding entity and the facility will be zero and derivative of the objective function will not be available [16].

To solve this kind of multiple facility Weber problem, there is an important approach. It is to update objective function to avoid nondifferentiability. If we want to locate q facilities to serve m demanding entities, the problem can be written as:

$$\text{minimize}_{(x_j,y_j)\ j=1,2,\ldots,q} \sum_{j=1}^{q}\sum_{i=1}^{m} w_{ij}d_{ij} + \sum_{j=1}^{q-1}\sum_{s=j+1}^{q} v_{js}d_{js} \tag{2.4}$$

where w_{ij} and v_{js} are weighting factors between a facility and a demanding entity, and between two facilities respectively. d_{ij} and d_{js} show distance between a facility and a demanding entity, and between two facilities respectively.

In [42], Wesolowsky and Love propose a gradient reduction method for problem 2.4 where rectilinear distances are used. The problem 2.4 take the following form for rectilinear distances.

$$\text{minimize}_{(x_j,y_j)\ j=1,2,\ldots,q}\ \sum_{j=1}^{q}\sum_{i=1}^{m} w_{ij}\left(\left|x_j-a_i\right|+\left|y_j-b_i\right|\right)+\sum_{j=1}^{q-1}\sum_{s=j+1}^{q} v_{js}\left(\left|x_j-x_s\right|+\left|y_j-y_s\right|\right) \tag{2.5}$$

Authors do not minimize the objective function of the problem. Instead of that, they minimize the function that is a nonlinear approximation of the objective fuction. This approximation function has continuous derivatives with respect to the facility locations. Authors make approximation by using hyperbolas with centers $(a_i, 0)$ and $(b_i, 0)$ for x and y coordinates of the facilities, respectively. Also, this hyperbolas have asymptotes with slopes $\pm w_{ij}$. To be more clear, let's consider a term $w_{ij}\left|x_j-a_i\right|$. Its hyberboloid approximation function is $w_{ij}\left(\left(x_j-a_i\right)^2+c^2\right)^{1/2}$. By keeping constant term of hyperbolas, c, arbitrarily small, sum of these hyperbolas approximate to the original objective function. Authors find an upper bound for error as a function of c. Also, they conclude that speed of convergence increases by inceasing c with the small expense in the accuracy.

In [18], authors propose an approximation method when the Euclidean distances are used. The multifacility Weber problem take the following form for the Euclidean distances.

$$\text{minimize}_{(x_j,y_j)\ j=1,2,\ldots,q}\ \sum_{j=1}^{q}\sum_{i=1}^{m} w_{ij}\sqrt{\left(x_j-a_i\right)^2+\left(y_j-b_i\right)^2}$$
$$+\sum_{j=1}^{q-1}\sum_{s=j+1}^{q} v_{js}\sqrt{\left(x_j-x_s\right)^2+\left(y_j-y_s\right)^2} \tag{2.6}$$

Authors introduce a small positive number ϵ into 2.6. Let's consider a term $w_{ij}d_{ij}$ where d_{ij} is the Euclidean distance in 2.6. Its hyberboloid approximation function is $w_{ij}\sqrt{d_{ij}^2+\epsilon}$. When ϵ goes to zero approximation functions goes to the original function. As long as ϵ is not zero, 2.6 is continuously differentiable. By using this approximating function, they defined an algorithm called as Hyperboloid Approximation Procedure (HAP). Here are the steps of the algorithm.

1. Start with initial solution.
2. Update facility locations until convergence is achieved by the formula that is obtained by taking the derivative of the approximated objective function and setting it to 0.

In the paper, convergence of the algorithm is not proven. But Rosen and Xue proved that HAP algorithm always converges [38].

Primal-dual methods and interior point methods are also applied to multifacility Weber problem [16].

The second way of adding new facilites to the Weber problem is to model problem as location-allocation model (LA model). In this model, allocation of demand regions to the facilities in addition to location of facilities are decided [16]. LA models firstly introduced to the literature by Leon Cooper in 1963. Objective functions of LA models are neither convex nor concave, so they are quite difficult and they have many local minima [16].

LA models can be defined as follows : given the location of a set of m demanding entities in terms of coordinates, find the locations (coordinates) of q facilities and allocations of the demanding entities to the facilities in order to minimize total shipping cost or distances [11]. These models can be mainly divided into two. Demanding entities can be served by several facilities or they can be served by only one facility. In later one, we have binary variables that show whether the demanding entity i is assigned to the facility j or not. To solve LA problems, exact equations, heuristics and metaheuristics were proposed [19].

In [10], Cooper developed equations for exact solution of LA problem. These equations are obtained by taking derivative of the objective function and setting it to 0. Cooper then minimized the objective function for all possible combinations of assignments by using these equations. Also, he proposed an iterative method in the paper. The method always converged when the starting location of facilities were chosen as indicated in the paper. Convergence is not based on a proof, it is based on many experimentations. However, computation amount of this solution approach is problematic for the large problem instances (number of destinations > 10).

In [11], the following two basic observations about LA problems are made.

1. When the location of the facilities is known, determination of the allocations is trivial.
2. When the allocations are fixed, determination of the locations can be made easily.

Author proposed four heuristics in the paper. One of them is the alternate location-allocation heuristic which is very much related to our study. This heuristic uses the above observations and it can be explained as follows.

1. Start with a set of initial facility locations and generate q subsets of facilities.
2. Solve each subset by using exact location method for the single facility location case.
3. Allocate each destination to the nearest facility to generate new subset of destinations. If new subsets are same with the previous subsets stop algorithm, otherwise return to Step 2.

This is also the basic idea that we used in our alternate location allocation heuristics which will be mentioned in Chapter 4.

As mentioned earlier, LA models are NP-hard and have many local minima. To deal with large problems and to avoid local optimality, metaheuristics like simulated annealing, tabu search, genetic algorithm and variable neihborhood search have also been proposed [19].

2.3 Classification Based on the Objective of the Decision Maker

In the location problems, there are many different objectives. For example, some may try to minimize cost, response time, damage or discomfort. In addition to minimization type objectives, there may be maximization problems. Some may try to maximize profit, quality or wellbeing [16].

Most of the above objectives can be represented as a function of distance. Therefore, distance metrics used in the literature should be mentioned. Most famous distance metrics are rectilinear distance, euclidean distance and L_∞ norm. In [19], a table that shows various kinds of distance metrics used to solve location problems can be found.

2.4 Classification Based on the Geometry of the Demanding Entity

In location theory, customers are generally assumed as points in space. When the size of the customers are small relative to the distances between facilities and customers, this assumption can be used. However, size of the customers cannot be ignored in some situations. Customers can be treated as fixed regions with density functions or distributions representing the demand over these regions.

The regional demand may arise in the following situations.

1. In the concern of uncertain demand in which the demand is a random vector over a region,
2. In the concern of mobile demand,
3. In the concern of some discrete situations that the number of demand points is very large which makes the discrete problem cumbersome.

2.4.1 Demand points

The literature on problems with demand points are very wide. In all of the articles mentioned until now, demanding entities are assumed as demand points.

2.4.2 Demand regions

In the cases explained in Section 2.4, treating the demanding entities as demand regions instead of demand points are more meaningful. If demand regions are used, the distance calculation between a demanding entity and a facility requires more attention. This calculation can be done in three ways; namely, by the expected distance, or by the closest distance, or by the farthest distance between the demand region and the facility.

2.4.2.1 Expected distance

Expected distance has been extensively used in the literature. This distance may be meaningful when the distances from the facility to each person living in the region is important. Also, for the cases in which demand points are represented as random vectors expected distances are mostly used.

Love considered the situation in which the number of demand points is too large to treat each of them as a discrete point in his work [28]. He introduced the possibility of grouping demand points into demand areas. He divided the total population area under consideration into rectangular regions with known dimensions. His objective is to find the location of a facility so as to minimize total expected Euclidean distances between rectangular regions and the facility. He proved that the objective function of the problem is convex. He developed a responce-surface technique. Firstly, for the starting responce-surface parameter, gradient reducing process (search along the gradient) is conducted until the given criterion is obtained. Then, a smaller responce-surface parameter is chosen and gradient reducing process is repeated until limit of the responce-surface parameter is zero. Computational time of this method is better when compared to author's previous studies.

In [4], authors extended the Love's study in [28]. Love assumed that each demand region is rectangular and each region has uniform population density. These assumptions are the results of the effort to ease complex integral expressions. In this paper, authors get rid of these complex expressions by replacing the demand regions, which may be not rectangular and not uniformly distributed, with the centroids of the regions. Centroids of the regions are calculated such that the first moment of the region is divided by the area of the region. By this method, computational time is reduced and any geometric shape with an easily found centroid can be used.

Cooper considered the stochastic extension of the Weber problem in [12]. The location of demanding entities are not predetermined but random variables with given probability distribution in the paper and author tries to minimize the sum of the expected values of the Euclidean distances between demanding entities and the facility. The probability distribution used in the paper is bivariate normal distribution assuming that 2 random variables are uncorrelated, i.e. correlation coefficient is zero. Cooper prove that the objective function of the problem is strictly convex and therefore it has unique minimum where the gradient of it is equal to zero. He developed an iterative algorithm for this problem. Iteration formula and a convenient set of starting point can be seen from the paper.

In [23], authors show the errors in representing demand areas by a single point, aggregated point. The error may be as much as eight percent. Authors divided the errors into three types. First type of errors are inherent in the measurement of the distance from an aggregation point instead of spatially distributed population. Depending to the cases, this type of error may be both positive and negative. Second type of error is occured when facility lands on aggregation points. This error always underestimates the real distance and it is negative. Last type of error is due to the wrong allocation decisions. If there are multiple facilities, estimated distance may lead to allocate demand regions to the wrong facilities. This error is always positive and estimated distance is overestimation of real distance. These errors are independent of the number of individuals in each demand region but it is inversely propotional to number of demand regions. Use of many small regions to represent population areas will decrease the error in the cost at the expense of computational cost.

In [3], authors tried to locate one or more facilities to serve existing rectangular regions where the rectilinear norm is used. The objective was to minimize total weighted expected distances. This problem can be decomposed into two subproblems for the coordinates. The objective function is convex and nondifferentiable. Authors proposed a gradient-free direct search method for the problem. Method for good initial point and good search direction was also proposed. For the emprical solutions, algorithm converged but there is not a formal proof of convergence. For the computational time, the algorithm's performance is promising.

Carrizosa et al. [7] proposed general notation for the problem in which both demanding entities and facilities can be regions and expected distances are tried to minimize. The notation is inspired in Kendall's notation for Quening Theory. Authors showed the similarities and the differences between generalized Weber problem and its point version. Objective function of this problem is convex, continuous, finite everywhere, and positive. With these properties, authors proved that the problem has unique optimal solution when the distance measurement gauge is a strict gauge and probability distribution of the demanding entities are absolutely continuous. Also, they showed that when the probability distribution of the demanding entities are absolutely continuous, objective function is differentiable everywhere and its gradient is available. In such cases, gradient descent algorithms can be used instead of evaluating complex expectations.

In the regional Weber problems, evaluation of the objective function, i.e. calculaton of the bidimensional integral, has high computational cost. To avoid this issue, approximation by centroids or disks centered at centroid were used in previous works. Former approximation is very dependent to norm used and shape of the demand regions. Later one has also drawbacks. The expected distance to a disk is known only for special gauges. Also, the question of how large the error is not answered by these approximations. In [8], authors replaced demand regions with simpler regions by which the error kept under control. Therefore, problem can be solvable for any accuracy. Since there is an exact algorithm to evaluate expected distance to a triangle, their approximation method uses somewhat triangle approximation. For example, for an elliptical region, they made approximation by m-sided polygon and they used triangles constructed by the successive three corners of the polygon to calculate the expected distances. The error in this approximation is inversely proportional to the number of sides of the polygon but large number of sides requires high computations. In the paper, it is shown that results are very good even when the number of sides of the polygon is not too large.

In [15], authors analyzed two convex objective functions; namely, minimize the weighted sum of the maximum distances between demand regions and the facility, and minimize the maximum of the weighted average distance between demand regions and the facility. Algorithm for the former problem is explained in Section 2.4.2.3. For the later problem, Elzinga-Hearn type algorithm is used. To apply this type of algorithm, spherical approximation procedure is used for the solution of the problem based on three groups. Algorithm for the minimax-average distance model also starts with facility located on center of gravity. Then, algorithm selects three demand groups with the farthest average distances and solves the problem of these three groups. Then, the demand group with farthest average distance is found. If the distance of this group is not greater than the required amount, optimal solution is found. Otherwise, three different problems of consisting two out of three oldest groups (there are three ways of selecting these groups) and one newly found farthest group is solved. Next iteration is defined by the group with the largest objective function value among three problems solved in previous step. Computational experiment shows that run times are proportional to the total number of demand points and number of iterations required to find solution decreases with the number of groups.

Chen proposed a new Weiszfeld-like iterative approach to locate single facility that will serve circular demand regions in [9]. He proposed equations for the net pulling forces of circular demand regions for three cases, namely facility can be inside, or outside, or on the boundary of the demand region by integration. Then, he used these equations as the numerator of the Weizsfeld algoritm. For the denominator of Weizsfeld algorithm as the step-size of the descent algoritm, he proposed weight functions for the same three cases. As a starting point for the algorithm, weighted average of the center point of the circular demand regions is recommended. To test the effectiveness of the proposed algorithm, author generated some examples in which circles has very small radius so he could compare the results with the results of standard Weiszfeld algorithm for the demand points are located at the centers of these circles. The results were very similar but original Weiszfeld converged faster.

2.4.2.2 Minimum distance

Closest distance may be meaningful for the installation areas since drop-off and take-off points will be on the boundary of the area which is closest to the facility [5]. In other words, flow from/to the facilities will enter/leave the given demand area at the closest point and then internal distribution costs within the demand area will not be considered.

In the paper [5], authors considers the single facility minimax problem. Different than the papers in the previous section, they used the closest point on the boundary of the regions instead of expected distance. They tried to minimize the maximum closest point between the demand regions and the facility. Objective function of this problem is convex. Since the standard nonlinear packages may be inefficient for large problem instances, authors developed a procedure based on the iso-contours. They demonstrated the iso-contour constraction in analytic form for both Euclidean and rectangular closest distances. Then, procedure starts with an arbitrary facility location. For this location they find customer with the maximum closest distance and by using the iso-contour of this customer they find the descent direction. Facility location is updated by using this descent direction. Then, procedure starts again with updated facility location until convergence is achieved. Also, they modeled the special case of the problem such that all demand regions are rectangular regions and rectangular distance is used as a linear program.

In [6], a single facility location problem with closed convex demand regions is studied. Objective of the paper is to locate a facility so as to minimize the sum of the closest Euclidean distances between demand regions and the facility. It is a minisum problem. The algorithm developed starts with an arbitrary initial facility. The closest point for each demand region is found and authors treated these closest points as fixed points which are replaced with the respective demand regions. Therefore, problem becomes a standard Weber problem they solve the problem with Weiszfeld algorithm as described in Section 2.1.2. Then, procedure is start again with the new facility until convergence is achieved. As in the Weiszfeld algorithm, a special attention is required when the facility location lands in a demand region. Authors also take this into consideration in their algorithm. They avoid the comptutation of descent directions and step sizes by adapting Weiszfeld algorithm. However, the proposed algorithm may converge to a non-optimal point on the boundary of any region. When this is the case, authors defined additional steps in the paper. Also, the convergence of the algorithm for special case such that any solution in the sequence and the optimal solution of the problem is not on the boundary of any demand region is proven in the paper.

2.4.2.3 Maximum distance

When the worst-case scenarios are important, this type of models are used. The location of emergency facilites such as fire stations, police stations, hospitals etc. are mostly modeled by using farthest distances.

In [39], authors considered the extension of standard p-center problem. The objective is to minimize the maximum of the maximum distances between demand originating in an area and the closest facility. In other words, aim is to cover all demand areas by p circles with smallest possible radius. For the problem, an algorithm based on error-free Voronoi diagram is proposed. Algorithm starts with p random centers. Then, Voronoi diagram is constructed based on these centers. For each Voronoi polygon, facilities are relocated by solving 1-center problem. Then, procedure starts again until convergence is achieved. The 1-center problems solved in the algorithm is equivalent to 1-center problem with demand points on the vertices of the Voronoi polygon. Since the number of vertices is low on average, procedure is fast and efficient. In some cases, convergence of the algorithm is slow. Therefore, authors developed a finishing-up algortihm. Also, lower and upper bounds are proposed for the square area.

By assuming demand points are clustered into groups, authors generalized the single facility location problem in [15]. Each cluster is one demand source. As it said before, authors analyzed two convex objective functions; namely, minimize the weighted sum of the maximum distances between demand regions and the facility, and minimize the maximum of the weighted average distance between demand regions and the facility. When the rectilinear norm is used, problems are formulated as linear problems. For Euclidean norm is used problems become nonlinear problems. For small problem instances Excell or AMPL were used. For large problem instances AMPL and Excell is not able to handle such large nonlinear problems. So they proposed special algorithms. Algorithm for minisum-max distance model starts with initial solution, let say center of the gravity. The farthest points of each demand group to this facility is found. Based on these farthest points one iteration of Weizsfeld algortihm is performed. If the distance between two successive facility location is less than given tolerance, optimal solution is found. Otherwise, farthest points from each group is calculated again. If farthest points do not change algorithm continues to perform one Weiszfeld iteration again. If farthest points change, the point with minimum objective function on the segment connecting two successive facility location is found by golden section search and algorithm continues with this facility location. The algortihm for other problem is mentioned in Section 2.4.2.1.

Jiang and Yuan considered the extended version of Weber problem in which customers are convex demand regions (they assumed rectangular or circular regions in the paper) in [24]. Their aim is to minimize sum of weighted farthest Euclidean distances between demand regions and the facility. The difficulty of solving this problem is the discontinuity of the farthest points. When facility location changes, farthest points of the regions may also change discontinuosly. If the region is circular then the calculation of farthest distance is easy and it is equal to distance between center of the circle and the facility plus radius of the circle. When the region is rectangular, let us look Figure 2.1. Authors draw a pair of symmetry axes so they divide the plane into four quadrants. If the facility is within a quadrant, the farthest point of the region is unchanged and it is the corner within the opposite quadrant. Like in the figure, when facility is within the first quadrant the farthest point is the corner within the third quadrant. When the number of rectangular regions increase, each of the rectangles have symmetry axes and they construct the subdivision of the plane called as fixed regions. Let's say that the number of rectangular regions is k, then number of fixed regions is bounded by $2k^2 + k + 1$. In some cases this number may be very large. Authors showed that all of the fixed regions may not be considered. If an unbounded fixed regions satisfies some conditions, solution of the problem can only be on the bounded sides of these regions. Therefore, visiting the only bounded fixed regions and unbounded fixed regions that do not satisfy the conditions in the paper will be enough. For a given fixed region the farthest points of each rectangular region is known and authors replaced regions with this points. Then, they solved constrained Weber problem (CWP) for each of these fixed regions. In other words, problem is transformed to at most $2k^2 + k + 1$ CWPs. To solve CWPs, authors transformed CWPs into linear variational inequalities and then solved these inequalities by a projection contraction method. If the solution is in the interior it is the optimal solution. If it is on the boundary, algorithm continues with solving neighborhood CWP. Computational time of the algorithm is very dependent to the starting fixed region. Authors also propose a greedy strategy for choosing the starting region.

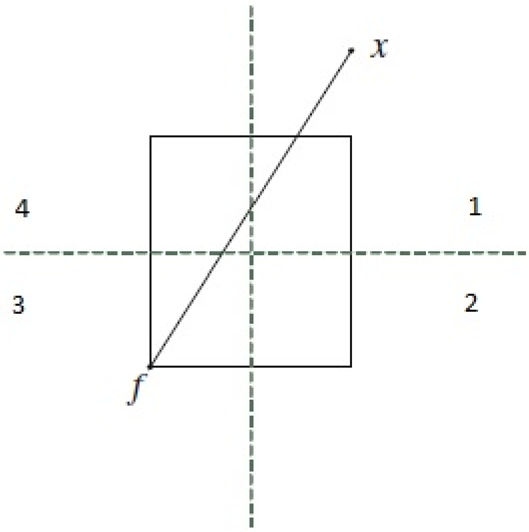

Figure 2.1: Dividing the plane into fixed regions

In [25], Jiang and Yuan considered multiple facility extension of the problem in [24]. They developed a hybrid Cooper type location-allocation heuristic. In the location step they solved single source minisum problems (SMP) with regional demand under the farthest distance. Each SMP is transformed into CWPs like in [24]. Since the number of CWPs may be too large for the cases of large number of regions, to solve each CWP with high efficiency is very important. For this purpose authors used Barzilai-Borwein gradient method. Original version of Barzilai-Borwein method is not applicable to this problem because of the nonsmoothness of the objective function. Therefore, authors updated the method with integration of the modified Weiszfeld procedure and called it as Barzilai-Borwein-Weiszfeld procedure. Authors proved the convergence of the Barzilai-Borwein-Weiszfeld procedure. In the allocation step, they used the nearest center reclassification heuristic. Numerical studies showed that the proposed algorithm is quite efficient.

CHAPTER 3

MATHEMATICAL PROGRAMMING FORMULATIONS OF FACILITY LOCATION PROBLEMS WITH DEMAND REGIONS BY USING SOCP

SOCP is a type of convex programming. Objective function of SOCP is a linear function. Constraints of SOCP are affine mappings of Second Order Cones (SOC). SOCP can also handle linear equality constraint in addition to cone constraints. The general form of an SOCP is

$$\text{minimize} f^T x$$

$$\text{subject to } \|A_i x + b_i\| \leq c_i^T x + d_i \text{ for } i = 1, 2, ..., L \tag{3.1}$$

$$Fx = g \tag{3.2}$$

where A_i and F are matrices, f, c_i and g are column vectors of appropriate size, and b_i, d_i are scalars.

SOCP is a more general convex program than LP. There are fast algorithms for SOCP that find global optimum. Like LP problems, SOCP problems can be solved in polynomial time. Many convex programs like convex quadratic programs, quadratically constrained convex quadratic programs can be formulated as SOCP problems [1]. The single facility version of our problem can be also formulated as a SOCP problem.

As it was mentioned earlier, the characteristics of the problem we consider in this study can be listed as follows.

1. We have m demand regions. All of these regions are closed convex polygons. Actually, at the beginning our regions may not be closed or convex. But any region can be approximated with a polygon in any accuracy. Details of the approximation procedure is mentioned in Section 5.
2. We want to locate q facilities.
3. Each demand region has equal weights.

4. Each demand region is to be allocated to exactly one facility.
5. Our objective is to minimize the sum of squares of the maximum Euclidean distances between the facilities and the allocated regions.

Let $S = \{s_1, ..., s_m\}$ be the set of m closed convex polygonal demand regions and $K_j = \left\{1, ..., k_j\right\}$ be the index set for corners of j^{th} demand region. Let $\left\{s_j^1, s_j^2, ..., s_j^{k_j}\right\}$ be the set of corners of the j^{th} demand region. Let $X = \left\{x_1, ..., x_q\right\}$ be the set of q facility locations and $x_i \in \mathbb{R}^2$, i.e., x_i is a two dimensional vector representing the coordinates of the i^{th} facility. By using SOCP, we propose the mathematical formulation for the problem above for the case $q = 1$ in Section 3.1 and for the case $q > 1$ in Section 3.2.

3.1 Mathematical Formulation of the Single Facility Case

In single facility location problems, there is not an issue of the allocation of the demanding entities. Therefore, there is no need for the binary variables that show the allocation of the demanding entities to the facilities. Let $X = \{x\}$ be the facility location. Then, the maximum Euclidean distance between the facility and the demand region s_j is calculated by the following formula.

$$z_j = \max_{k \in K_j} \left\| s_j^k - x \right\| \tag{3.3}$$

Let t_j be the square of the maximum distance, $t_j = z_j^2$. The mathematical formulation of the single facility location problem can be represented as follows.

$$\text{minimize} \sum_{j=1}^{m} t_j$$

$$\text{subject to } t_j = \max_{k \in K_j} \left\| s_j^k - x \right\|^2 \ \forall\ j \tag{3.4}$$

Since the problem is a minimization problem, equation 3.4 is equivalent to the following equation.

$$t_j \geq \left\| s_j^k - x \right\|^2 \forall\ j, k \in\ K_j \tag{3.5}$$

Constraint 3.5 is a SOCP representable constraint. After a small alteration, SOCP representation is as follows.

$$\text{minimize} \sum_{j=1}^{m} t_j$$

$$\text{subject to} \quad \frac{1+t_j}{2} \geq \left\| \begin{matrix} s_j^k - x \\ \frac{1-t_j}{2} \end{matrix} \right\| \quad \forall\ j, k \in\ K_j \tag{3.6}$$

It is easy to see that, the square of the constraint of the model 3.6 is equal to equation 3.5.

Since SOCP is solvable in polynomial time, we say that the single facility location problem can be solved in polynomial time with the model 3.6. Time complexity of per iteration of single facility location problem with SOCP is $O(n^2 \sum_{i=1}^{N} n_i)$, where n is the number of decision variables, n_i is the dimension of i^{th} cone constraint and N is the number of cone constraints. Number of decision variables is $m + 2$, m decision variables for distance squares and 2 decision variables for the coordinates of the facility. Dimension of all cone constraints is 3 and the number of cone constraints is equal to the total number of corners in the problem, $N = \sum_{j=1}^{m} \left|K_j\right|$. Therefore, time complexity is $O(m^2 N)$ for one iteration. The number of iterations in a SOCP is bounded above by $O(\sqrt{N})$. Thus, the overall worst case time complexity of a single facility location problem with SOCP is $O(m^2 N^{3/2})$. In practice, however, the number of iterations is usually ≤ 50, and hence the complexity in practice is usually $O(m^2 N)$. If $\left|K_j\right|$s are constant, the complexity of the single facility location problem will be $O(m^3)$. [27] can be investigated for further information about the time complexity of SOCP.

Remark: *If our objective is to minimize the sum of the maximum Euclidean distances between the facilities and the allocated regions, the problem becomes same with the farthest minisum location problem (FMLP) considered in [24]. Solving FMLP is difficult because of the discontinuity of the farthest points of demand regions to the facility. To solve this difficulty, FMLP problem is converted to the constrained Weber problems in that paper. However, the problem is SOCP representable. In other words, it can be easily solved in polynomial time. SOCP representation of the single facility location problem with sum of the distances as an objective follows.*

$$\text{minimize} \sum_{j=1}^{m} z_j$$

$$\text{subject to} \quad z_j \geq \left\| s_j^k - x \right\| \quad \forall\ j, k \in\ K_j \tag{3.7}$$

3.2 Mathematical Formulation of the Multiple Facility Case

The maximum distance between the demand region and the nearest facility in multiple facility location problem can be calculated by the following formula.

$$z_j = \min_{i=1,\ldots,q} \max_{k\in K_j} \left\| s_j^k - x_i \right\| \tag{3.8}$$

The allocation of the demanding entities to the facilities is an important issue in multiple facility location problems. By defining binary variables for allocation decisions, we can get rid of the $\min_{i=1,\ldots,q}$ function in equation 3.8. Let us define the following binary variable.

$$a_{ji} = \begin{cases} 1 & \text{, if region } j \text{ is allocated to facility } i \\ 0 & \text{, otherwise} \end{cases}$$

Then, equation 3.8 can be rewritten as following.

$$z_j = \sum_{i=1}^{q} a_{ji} \max_{k\in K_j} \left\| s_j^k - x_i \right\| \tag{3.9}$$

The following is a mathematical formulation of the multiple facility location problem when $t_j = z_j^2$.

$$\text{minimize} \sum_{j=1}^{m} t_j$$

$$\text{subject to } t_j = \sum_{i=1}^{q} a_{ji} \max_{k\in K_j} \left\| s_j^k - x_i \right\|^2 \; \forall \; j \tag{3.10}$$

$$\sum_{i=1}^{q} a_{ji} = 1 \; \forall \; j \tag{3.11}$$

$$a_{ji} \in \{0,1\} \;\; \forall \; i, j \tag{3.12}$$

Constraint 3.10 obviously shows the square of the maximum distance from a demand region to the nearest facility. Constraint 3.11 guarantees that each demand region can be allocated to only one facility.

Since the problem is minimization problem, we can rewrite the constraint 3.10 as follows.

$$t_j + \left(1 - a_{ji}\right) M \geq \left\| s_j^k - x_i \right\|^2 \;\; \forall \; j, i, k \in \; K_j \tag{3.13}$$

where M is big-M.

Selecting M is an important issue, since it effects the size of the branch-and-bound tree. To select big-M as small as possible, we calculated the square of the distance between the southern west and northeast corner of the rectangle that covers all the demand regions. The sides of this rectangle are parallel to the coordinate axes.

The constraint 3.13 is a SOCP representable constraint. With a small alteration, here is the mixed integer SOCP representation of the multiple facility location problem.

$$\text{minimize} \sum_{j=1}^{m} t_j$$

$$\text{subject to} \quad \frac{1 + t_j + \left(1 - a_{ji}\right) M}{2} \geq \left\| \begin{matrix} s_j^k - x_i \\ \frac{1 - t_j - \left(1 - a_{ji}\right) M}{2} \end{matrix} \right\| \;\; \forall \; j, i, k \in \; K_j \tag{3.14}$$

$$\sum_{i=1}^{q} a_{ji} = 1 \; \forall \; j \tag{3.15}$$

$$a_{ji} \in \{0, 1\} \;\; \forall \; i, j \tag{3.16}$$

The mixed integer SOCP above is a weak formulation. Solving large instances with this formulation is problematic. The branch-and-bound tree becomes too large and out of memory error appears.

Remark: *If our objective is to minimize the sum of distances instead of distance squares, the problem is again mixed integer SOCP representable. SOCP representation of the multiple facility location problem with sum of the distances as an objective is:*

$$\text{minimize} \sum_{j=1}^{m} z_j$$

$$\text{subject to} \;\; z_j + (1 - a_{ji}) M \geq \left\| s_j^k - x_i \right\| \;\; \forall \; j, i, k \in \; K_j \tag{3.17}$$

$$\sum_{i=1}^{q} a_{ji} = 1 \; \forall \; j \tag{3.18}$$

$$a_{ji} \in \{0, 1\} \;\; \forall \; i, j \tag{3.19}$$

CHAPTER 4

HEURISTICS FOR THE MULTIPLE FACILITY LOCATION PROBLEM WITH DEMAND REGIONS

Characteristics of the problem we deal with are mentioned in Chapter 3. We developed three heuristics and a special case heuristic for this problem. In Section 4.1, the details of the our first heuristic called as "second order cone programming based alternate location allocation heuristic" is given. Our second heuristic whose name is "max point based alternate location allocation heuristic" is mentioned in Section 4.2. In Section 4.3, our third heuristic called as "smoothing based heuristic" is described. Finally, "line based heuristic" that we developed for a special case of the problem is explained in Section 4.4.

4.1 Second Order Cone Programming Based Alternate Location Allocation Heuristic (SOCP-H)

Alternate location-allocation procedure is well-known heuristic approach for multi facility location problems as mentioned in Section 2.2.2. With the same idea, we developed an alternate location allocation heuristic. In the location step of the algorithm we solve q single facility location problems with SOCP formulation in Section 3.1. Basic framework of SOCP-H is given in **Algorithm 1**.

Step 1 is **the initialization step**. We select random q facility locations which are inside the convex hull of the corner points of the demand regions since any facility outside of the convex hull cannot be an optimal solution. Step 3 is **the allocation step**. In this step, locations of the facilities are known. Then allocation of the demand regions to the facilites are relatively easy. The algorithm tries to assign each demand region to a facility that minimizes the maximum distance between the region and the facility. Step 4 through Step 6 are **the location steps**. Assigment of the regions to the facilities are known in this steps. Then, the algorithm solves q single facility location problems to locate the facilities. This heuristic repeats allocation and location steps until convergence is achieved or maximum number of iterations is reached.

Algorithm 1 SOCP-H

1: Start with arbitrary initial facility locations $X = \left\{x_1, ..., x_q\right\}$.
2: **repeat**
3: Find q disjoint sets $\left\{S_1, ..., S_q\right\}$ where $S_i = \left\{s_{i_1}, ..., s_{i_{l_i}}\right\}$ is the set of l_i demand regions that are assigned to facility i by considering:

$$i = \text{argmin}_{i=1,...,q} \max_{k \in K_j} \left\| s_j^k - x_i \right\| \; \forall \; j$$

4: **for** $i = 1$ to q **do**
5: Solve i^{th} single facility location problem for the set S_i using the model 3.6 to find the optimal location, $x_i^{'}$.
6: **end for**
7: Update X. $\left\{x_1, ..., x_q\right\} = \left\{x_1^{'}, ..., x_q^{'}\right\}$
8: **until** Convergence is achieved. **OR** Maximum number of iterations is reached.
9: **return** X^*

The major advantage of SOCP-H is that each location step takes polynomial time to solve as it was mentioned in Section 3.1. On the other hand, SOCP-H has some disadvantages. This heuristic may not be suitable for large problem instances. Since the worst case time complexity of a location step is $O(m^4)$, convergence of the heuristic may take a long time. Also, we do not guarantee to find global optimum solution with SOCP-H. Lastly, locations of the facilities are dependent to initialization.

Remark: *SOCP-H can be modified to the case in which sum of the distances is minimized instead of sum of squares of the distances. The framework of the updated algorithm would be very similar to the SOCP-H. The only changing step is Step 5. In this step, single faciliy location problems should be solved with the model 3.7 instead of the model 3.6.*

4.2 Max Point Based Alternate Location Allocation Heuristic (MP-H)

Lets consider the single facility location problem with 3 demand points that tries to minimize the sum of the squares of the Euclidean distances between the demand points and the facility. Let (a_i, b_i) be the coordinates of the demand points and (x, y) be the coordinates of the facility. Then the objective function we are trying to minimize is the following.

$$\text{minimize } f(x,y) = \left(\sqrt{(x-a_1)^2 + (y-b_1)^2}\right)^2 + \left(\sqrt{(x-a_2)^2 + (y-b_2)^2}\right)^2 + \left(\sqrt{(x-a_3)^2 + (y-b_3)^2}\right)^2 \quad (4.1)$$

$f(x, y)$ is a convex function. We obtain the point of minimum by taking the partial derivatives of $f(x, y)$ with respect to x and y, and setting partial derivatives to 0. This point of minimum is $\left(\frac{a_1+a_2+a_3}{3}, \frac{b_1+b_2+b_3}{3}\right)$, i.e. the average of the coordinates of the demand points.

Lets consider equation 3.3 for the single facility location problem. If we know the facility location, we can find the maximum distanced corner of each region to the facility by the following formula.

$$k_j^* = \text{argmax}_{k \in K_j} \left\| s_j^k - x \right\| \tag{4.2}$$

Then, the objective function of the single facility location problem takes the following simpler form. It has the same form with $f(x, y)$ in 4.1.

$$\text{minimize} \sum_{j=1}^{m} \left\| s_j^{k_j^*} - x \right\|^2 \tag{4.3}$$

The optimal value for x is easily determined by taking the average of $s_j^{k_j^*}$. By using this information, we developed another alternate location-allocation heuristic. Basic framework of MP-H is given in **Algorithm 2**.

Algorithm 2 MP-H

1: Start with arbitrary initial facility locations $X = \{x_1, ..., x_q\}$.
2: **repeat**
3: Find q disjoint set $\{S_1, ..., S_q\}$ where $S_i = \{s_{i_1}, ..., s_{i_{l_i}}\}$ is the set of l_i demand regions that are assigned to facility i by considering:

$$i = \text{argmin}_{i=1,...,q} \max_{k \in K_j} \left\| s_j^k - x_i \right\| \ \forall\ j$$

4: **for** $i = 1$ to q **do**
5: **for** $t = 1$ to l_i **do**
6: Find farthest corner of region s_{i_t} in the set S_i by considering equation 4.2.
7: **end for**
8: Calculate the mean of $\left\{ s_{i_1}^{k_{i_1}^*}, ..., s_{i_{l_i}}^{k_{i_{l_i}}^*} \right\}$ to locate i^{th} facility. $x_i' = mean \left\{ s_{i_1}^{k_{i_1}^*}, ..., s_{i_{l_i}}^{k_{i_{l_i}}^*} \right\}$
9: **end for**
10: Update X. $\{x_1, ..., x_q\} = \{x_1', ..., x_q'\}$
11: **until** Convergence is achieved. **OR** Maximum number of iterations is reached.
12: **return** X^*

As you can see, the initialization and the allocation steps of MP-H is same with the initialization and allocation steps of SOCP-H. But location steps, Step 4 through Step 9, are different than SOCP-H. Updated location of a facility is computed by averaging the farthest points of the assigned regions from the previous location of the facility.

MP-H has computation time advantage over SOCP-H. The worst case time complexity of a location step is $O(m)$. Since MP-H avoids solving SOCP problems in location step it is very fast and it finds remarkably good solutions. However, MP-H has its handicaps. Algorithm may cycle without convergence. For example, let us analyze the Figure 4.1. There is a single facility location problem with 3 rectangular demand regions. Let the red plus sign in Figure 4.1(a) be the initial arbitrary facility location. In Figure 4.1(b), farthest corners of each demand region to this facility are marked with blue circles. The facility location of the next iteration that is calculated by averaging the marked corners can be seen from Figure 4.1(c). The farthest corners of the demand regions to new facility and the facility location of the next iteration are represented in Figure 4.1(c) and Figure 4.1(e), respectively. When we take the average of the marked points in Figure 4.1(f), we obtain the facility location in Figure 4.1(c). If we continue to make iterations, facility location cycles between the locations in Figure 4.1(c)and Figure 4.1(e). In this study, we did not try to eliminate cycles. Instead, we limited the maximum number of iterations. Also, final locations of the facilities are dependent to initialization like in SOCP-H. Again, finding global optimum solution is not guaranteed.

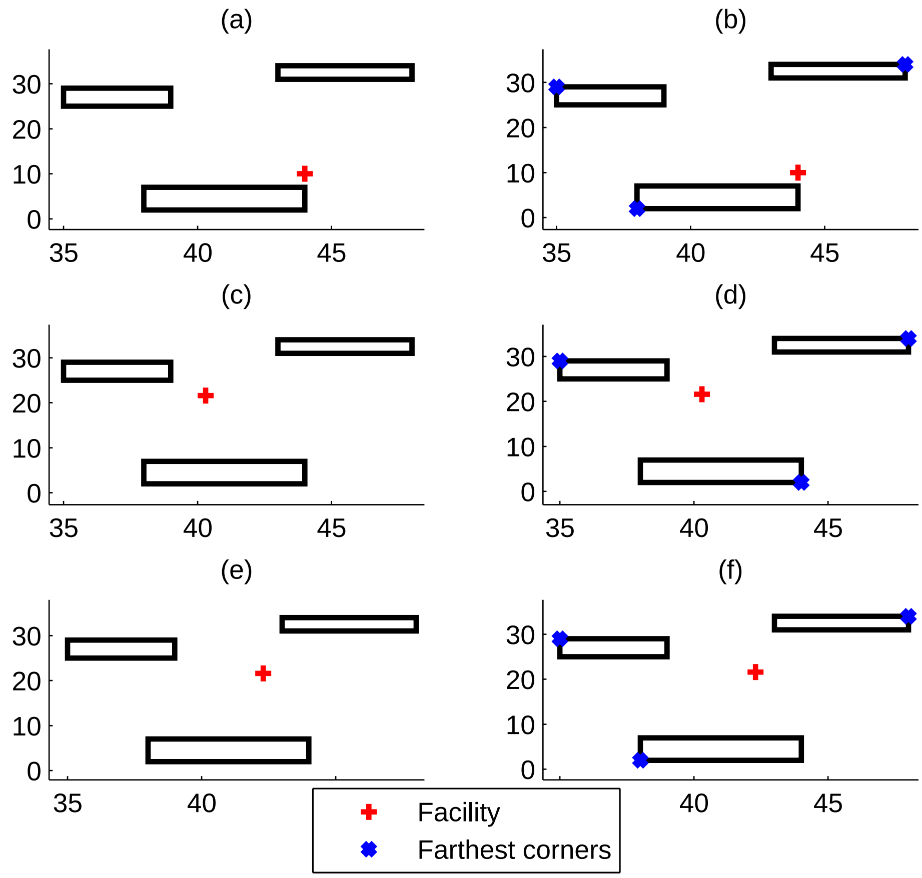

Figure 4.1: An example of cycling of MP-H without convergence

Remark: *MP-H can also be modified so as to minimize the sum of the maximum distances instead of the squares of the maximum distances. With this changed objective, the single facility location problems in the location steps are transformed into the Weber problems. Taking average of the farthest points to locate facilities does not work anymore. As it was explained in Section 2.1.2, the most common procedure to solve the Weber problem is the Weiszfeld Procedure. By using the Weiszfeld Algorithm in the location steps of MP-H and keeping other steps as the same, the required modification can be done.*

4.3 Smoothing Based Heuristic (SBH)

Other than previous two alternate location allocation heuristics, we developed another heuristic. The idea of SBH comes from the fact that the constraints of the multiple facility location problem is nondifferentiable. Consider the distance calculation in equation 3.8 for multiple facility case. In Section 3.2, we mentioned the mixed integer SOCP formulation of the problem. Without using the binary variables, the following is an alternative mathematical formulation for the problem.

$$\text{minimize} \sum_{j=1}^{m} z_j^2$$

$$\text{subject to } z_j = \min_{i=1,\ldots,q} \max_{k \in K_j} \left\| s_j^k - x_i \right\| \; \forall \; j \tag{4.4}$$

Firstly, we smoothed the nondifferentiable functions with the strategy like in [43] to solve model 4.4. In the paper, author smooths the minimum sum-of-squares clustering (MSSC) formulation by substituting the nondifferentiable functions with continuously differentiable functions. We then transformed the smothed problem into an unconstrained problem with the help of the implicit function theorem. Finally, we solved this unconstrained nonlinear problem with a quasi-Newton algorithm that uses Broyden-Fletcher-Goldfarb-Shanno (BFGS) updating method. Basic framework of SBH is:

Algorithm 3 SBH

1: Smooth the problem and transform it to unconstrained optimization problem as we will explain.
2: Start with arbitrary initial facility locations $X = \{x_1, ..., x_q\}$ and choose initial values of smoothing parameters,$\epsilon, \tau, \mu, \gamma$.
3: **repeat**
4: Solve problem with quasi-Newton method to find X^*.
5: Update X, $X = X^*$.
6: Update smoothing parameters, $\epsilon, \tau, \mu, \gamma$: divide them by $\rho_1, \rho_2, \rho_3, \rho_4$.
7: **until** Convergence is achieved. **OR** Maximum number of iterations is reached
8: **return** X.

In the following sections, details of the smoothing procedure, tranformation procedure and quasi-Newton method are mentioned, respectively.

4.3.1 Smoothing Procedure

Consider model 4.4. Since the problem is a minimization problem, when we rewrite the constraints as follows the constraints hold as equality.

$$z_j - \min_{i=1,\ldots,q} \max_{k \in K_j} \left\| s_j^k - x_i \right\| \geq 0 \ \forall \ j \tag{4.5}$$

The following equation converges to 4.5 as ϵ gets smaller.

$$\sum_{i=1}^{q} max \left\{ z_j - \max_{k \in K_j} \left\{ \left\| s_j^k - x_i \right\| \right\}, 0 \right\} = \epsilon \ \forall \ j \tag{4.6}$$

where ϵ is a very small positive number.

In equation 4.6, value of $z_j - \max_{k \in K_j} \left\| s_j^k - x_i \right\|$ will be only positive if region j is assigned to facility i. Otherwise, it will be negative. When it is positive z_j value is a little bit more than the maximum of the distances between the corners of that region and the facility.

There are three nonsmooth functions in equation 4.6. First of them is $max\{y, 0\}$ function. In [43], author defines the following function.

$$\phi(y, \tau) = \frac{y + \sqrt{y^2 + \tau^2}}{2} \tag{4.7}$$

where τ is a very small positive number.

Function ϕ is an approximation of function $max\{y, 0\}$. When τ goes to zero, ϕ function becomes $max\{y, 0\}$ function. While latter function is nondifferentiable, former one is differentiable as long as τ is positive. Figure 4.2 shows the original and smoothed version of the function. By using function ϕ equation 4.6 takes the following form.

$$\sum_{i=1}^{q} \phi \left\{ z_j - \max_{k \in K_j} \left\{ \left\| s_j^k - x_i \right\| \right\}, \tau \right\} = \epsilon \tag{4.8}$$

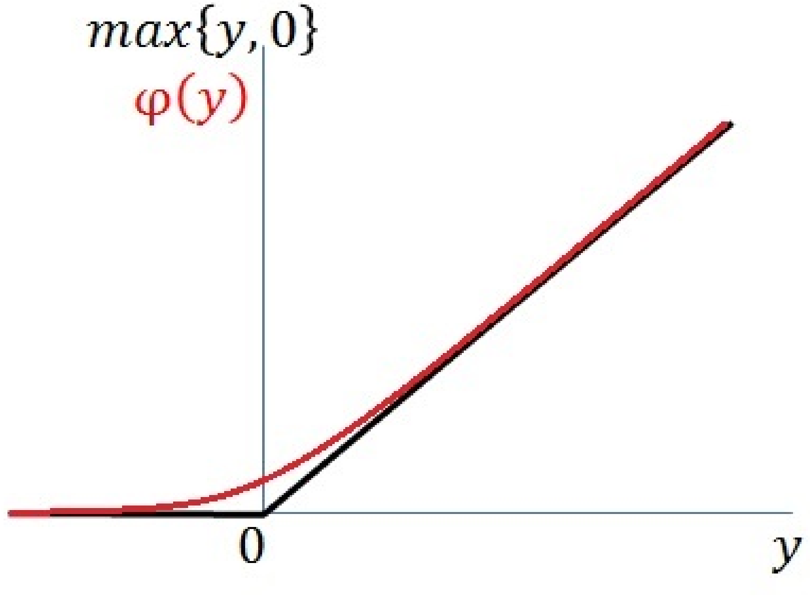

Figure 4.2: Original and smoothed $max\{y, 0\}$ function

Another nonsmooth function in equation 4.6 is $\max_{k \in K_j} \{...\}$. Let us define the following function.

$$\beta(x, y, \mu) = \frac{x + y + \sqrt{(x-y)^2 + \mu^2}}{2} \tag{4.9}$$

for a very small positive number μ.

Function β is an approximation of the function $max\{x, y\}$. As μ goes to zero, β function approaches to $max\{x, y\}$ function and it is differentiable as long as μ is positive. We used nested β functions to approximate $\max_{k \in K_j} \left\| s_j^k - x_i \right\|$ function. The level of nesting depends to $|K_j|$. For example, let us have only tetragons, i.e. $|K_j| = 4$. Equation 4.8 can then be written as following.

$$\sum_{i=1}^{q} \phi \left\{ z_j - \beta \left\{ \beta \left\{ \left\| s_j^1 - x_i \right\|, \left\| s_j^2 - x_i \right\|, \mu \right\}, \beta \left\{ \left\| s_j^3 - x_i \right\|, \left\| s_j^4 - x_i \right\|, \mu \right\}, \mu \right\}, \tau \right\} = \epsilon \tag{4.10}$$

Now, the only remaining nonsmooth function is $\|x - y\|$ function where x and $y \in \mathbb{R}^n$. In [43], author defines the following function:

$$\theta(x, y, \gamma) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2 + \gamma^2} \tag{4.11}$$

where γ is a very small positive number.

Function θ is an approximation of *norm* function. As γ goes to zero, θ function approaches to the *norm* function and it is differentiable everywhere as long as γ is positive. By using θ function, equation 4.10 takes the following form

$$\sum_{i=1}^{q} \phi\left\{z_j - \beta\left\{\beta\left\{\theta\left\{s_j^1, x_i, \gamma\right\}, \theta\left\{s_j^2, x_i, \gamma\right\}, \mu\right\}, \beta\left\{\theta\left\{s_j^3, x_i, \gamma\right\}, \theta\left\{s_j^4, x_i, \gamma\right\}, \mu\right\}, \mu\right\}, \tau\right\} = \epsilon. \tag{4.12}$$

After the smoothing procedure, the smooth formulation of the multiple facility location problem is given below.

$$\text{minimize} \sum_{j=1}^{m} z_j^2$$

subject to $h_j(z_j, X) =$

$$\sum_{i=1}^{q} \phi\left\{z_j - \beta\left\{\beta\left\{\theta\left\{s_j^1, x_i, \gamma\right\}, \theta\left\{s_j^2, x_i, \gamma\right\}, \mu\right\}, \beta\left\{\theta\left\{s_j^3, x_i, \gamma\right\}, \theta\left\{s_j^4, x_i, \gamma\right\}, \mu\right\}, \mu\right\}, \tau\right\} - \epsilon = 0 \ \forall \ j \tag{4.13}$$

4.3.2 Transformation Procedure

In 4.13, we have the following m nonlinear but smooth equations in $m+2q$ variables (m comes from the distances, $2q$ comes from the x and y coordinates of q facility locations).

$$h_j\left(z_j, x_1, x_2, ..., x_q\right) = 0, \quad j = 1, ..., m \tag{4.14}$$

A question arises in our minds, "is it possible to express m variables $(z_1, z_2, ..., z_m)$ in terms of remaining $2q$ variables $(x_{1x}, x_{1y}, x_{2x}, x_{2y}, ..., x_{qx}, x_{qy})$?". In other words, we are seeking for the answer of "do there exist $z_j(X)$ functions such that if we know the coordinates of the facility locations we can calculate the values of the distances?". If this is the case, 4.13 transforms to the following unconstraint problem.

$$\text{minimize } f(X) = \sum_{j=1}^{m} z_j(X)^2 \tag{4.15}$$

The implicit function theorem states when the functions $z_j(X)$ exist. According to the theorem, 4.13 has a solution $z_j(X)$ for all X in some neighborhood of $\bar{X}$ if m x m Jacobian matrix evaluated at $\bar{X}$ is nonsingular. Jacobian matrix is the matrix of the first order partial derivatives of the functions $h_j\left(z_j, x_1, x_2, ..., x_q\right)$ with respect to the variables $z_1, z_2, ..., z_m$. Jacobian matrix for this problem is calculated as follows:

$$J = \begin{bmatrix} \frac{\partial h_1}{\partial z_1} & \frac{\partial h_1}{\partial z_2} & \cdots & \frac{\partial h_1}{\partial z_m} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial h_m}{\partial z_1} & \frac{\partial h_m}{\partial z_2} & \cdots & \frac{\partial h_m}{\partial z_m} \end{bmatrix}$$

h_j only contains z_j. Therefore, only $\frac{\partial h_j}{\partial z_j}$'s have nonzero value, other partial derivatives are 0. Then, Jacobian matrix takes the following form.

$$J = \begin{bmatrix} \frac{\partial h_1}{\partial z_1} & 0 & \cdots & 0 \\ 0 & \frac{\partial h_2}{\partial z_2} & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{\partial h_m}{\partial z_m} \end{bmatrix}$$

As it is seen, J is a nonsingular matrix and allows us to use the implicit function theorem. The theorem states that all $z_j(X)$ functions are differentiable and following equations hold.

$$\frac{\partial h_j}{\partial x_i} = -\sum_{u=1}^{m} \frac{\partial h_j}{\partial z_u}\frac{\partial z_u}{\partial x_i} \quad \forall j = 1, ..., m \tag{4.16}$$

From the Jacobian matrix, we know that $\frac{\partial h_j}{\partial z_u}$ can take a nonzero value only when $u = j$ and it is 0 otherwise. Then, 4.16 becomes:

$$\frac{\partial h_j}{\partial x_i} = -\frac{\partial h_j}{\partial z_j}\frac{\partial z_j}{\partial x_i} \quad \forall j = 1, ..., m \tag{4.17}$$

Although we do not know exact equations for $z_j(X)$'s, we know all first order derivatives of $z_j(X)$'s with respect to x_i's. They are obtained from 4.17 as follows.

$$\frac{\partial z_j(X)}{\partial x_i} = -\frac{\frac{\partial h_j(z_j,X)}{\partial x_i}}{\frac{\partial h_j(z_j,X)}{\partial z_j}} \tag{4.18}$$

Therefore, we can calculate the first order derivative of the objective function in 4.15 although we do not know it explicitly. The first order derivative is:

$$\nabla f(X) = \sum_{j=1}^{m} 2z_j(X)\nabla z_j(X) \tag{4.19}$$

4.3.3 Quasi Newton Method

To solve the model in 4.15 we have used Quasi-Newton method. It is an iterative algorithm based on Newton's method for finding local minima or maxima of a function. In Newton's method, gradient and the Hessian matrix of the function is used. On the other hand, in quasi-Newton method Hessian matrix is not calculated. It is updated by using gradient vectors. In other words, Quasi-Newton method uses only gradient information of the function. Since we know all the first order derivative information of model 4.15, Quasi-Newton method is an appropriate way to solve that model. The procedure for solving the problem is described below.

Algorithm 4 Quasi-Newton Method

1: Set i=0.
2: Start with arbitrary initial facility locations, $X_0 = \left\{x_1, ..., x_q\right\}$.
Set initial Hessian matrix to identity matrix, $H_0 = I$.
3: **repeat**
4: Find a search direction d_i using first order derivative information of the function, $d_i = -H_i \nabla f(X_i)$.
5: Find a step size α_i to move along the search direction.
Here, we have one dimensional search problem to solve for α_i.

$$\text{minimize } f(X_i - \alpha_i H_i \nabla f(X_i))$$

To find optimal point of Problem 4.20, we used bisection search*.
6: Calculate $\Delta X_i = -\alpha_i d_i$ and update X, $X_{i+1} = X_i + \Delta X_i$.
7: Find the gradient at new point, $\nabla f(X_{i+1})$ and calculate the difference between the gradients of successive two iterations, $y_i = \nabla f(X_{i+1}) - \nabla f(X_i)$.
8: Update Hessian matrix with the following BFGS updating formula.

$$H_{i+1} = \left(I - \frac{y_i \Delta X_i^T}{y_i^T \Delta X_i}\right) H_i \left(I - \frac{y_i \Delta X_i^T}{y_i^T \Delta X_i}\right) + \frac{\Delta X_i \Delta X_i^T}{y_i^T \Delta X_i}$$

9: Set $i = i + 1$.
10: **until** Convergence is achieved. **OR** Maximum number of iterations is reached

** In bisection search, there is a need for an interval to search. However, for the problem in Step 5 we do not know the interval in which optimal solution lies. Therefore, before the bisection search, a search with no restriction on the value of the optimal solution should be conducted. There are two kinds of unrestricted search; namely, search with fixed step size and search with accelerated step size. Former one is not efficient for all cases. Because of the absense of the knowledge of the optimal solution's location, to reach the optimum point may require too many iterations in some cases. This drawback can be handled with the search with accelerated step size. In this method, step size is increased gradually as long as the moves* *result in improvement [36]. Because of the computational efficiency of accelarated step size, we used the second unrestricted search.*

When no improvement occurs in the unrestricted search, we conclude the interval of optimal solution by using step sizes in last and previous iterations. To this interval, we apply the bisection search until convergenceor maximum number of iterations is achieved. Result of the bisection search is α_i that we are searching for.

The convergence condition of Quasi-Newton method is same with the convergence condition of SBH. However, maximum number of iterations of the method set is different from the algorithm's maximum number of iterations.

In SBH, as the number of iterations increase, ϵ, τ, μ and γ approaches to 0. This means that the problem in 4.15 approaches to problem in 4.4. Also, as long as the smoothing parameters ϵ, τ, μ and γ are positive, equation 4.19 is applicable.

SBH has time advantage over SOCP-H. Since it uses only first order derivative information it is faster than SOCP-H. Also, experimentally it produces better results than MP-H. The solution qualities of SBH and SOCP-H is nearly same. As a final note, final locations of the facilities are dependent to initialization like in the previous two alternate location allocation heuristics.

Remark: *Like our other heuristics, SBH can be updated to the case where the sum of the maximum distances is minimized. When the case is sum of the distances, only the objective function of model 4.4 is changed. So the first order derivative of the objective function becomes* $\nabla f(X) = \sum_{j=1}^{m} \nabla z_j(X)$.

All of the procedures of SBH -smoothing, transformation, quasi-Newton- remain the same.

4.4 Line Based Heuristic (LBH)

Let's consider our objective function for single facility location problem again. Optimal values of x and y coordinates are independent, since partial derivative of the objective function with respect to x and y is independent of y and x respectively. This means that, we can solve the problem for x and y coordinates independently.

LBH is an alternate location-allocation heuristic for special cases of our problem such that each demand region is a rectangular region with their sides parallel to the coordinate axes. It uses the basic idea of solving for x and y coordinates seperately. Basic framework of LBH is given below.

Algorithm 5 LBH

Start with arbitrary initial facility locations $X = \left\{x_1, ..., x_q\right\}$

repeat

  Find q disjoint set $\left\{S_1, ..., S_q\right\}$ where $S_i = \left\{s_{i_1}, ..., s_{i_{l_i}}\right\}$ is the set of l_i demand regions that are assigned to facility i by considering;

$$i = \text{argmin}_{i=1,...,q} \max_{k \in K_j} \left\| s_j^k - x_i \right\| \ \forall \ j$$

  for $i = 1$ to q **do**

    Divide rectangular regions into line segments on x and y axes.

    Find optimal solution on x dimension by following the procedure below.

    PROCEDURE : *Optimal coordinate on an axis*, **input:**line segments on respective axis

    Find centers of each line segments, $c_1, ..., c_{l_i}$

    Sort line segments in ascending order according to their centers, $c'_1, ..., c'_{l_i}$

    for $t = 1$ to $l_i - 1$ **do**

      Take the interval (c'_i, c'_{i+1})

      Calculate the *average* of the farthest points of line segments to the center of that interval

      if *average* $\in (c'_i, c'_{i+1})$ **then**

        $x'_{ix} = average$

        break

      end if

    end for

    END PROCEDURE

    Find optimal solution on y dimension by following the procedure "Optimal coordinate on an axis".

  end for

  Update $X = \left\{x'_{1x}, x'_{1y}, ..., x'_{qx}, x'_{qy}\right\}$

until Convergence is achieved. **OR** Maximum number of iterations is reached

return X^*

The initialization and the allocation steps of LBH is the same with the initialization and allocation steps of our previous two alternate location-allocation heuristics. But, location step of LBH is quite different than the previous ones. When the assignments of the regions to the facilities are known, algorithm solves q single-facility location problems on the plane. Each problem is converted into two single facility location problems on the line and solved to optimality. These two optimal solutions give the coordinates of the optimal solution of the single facility location problem on the plane.

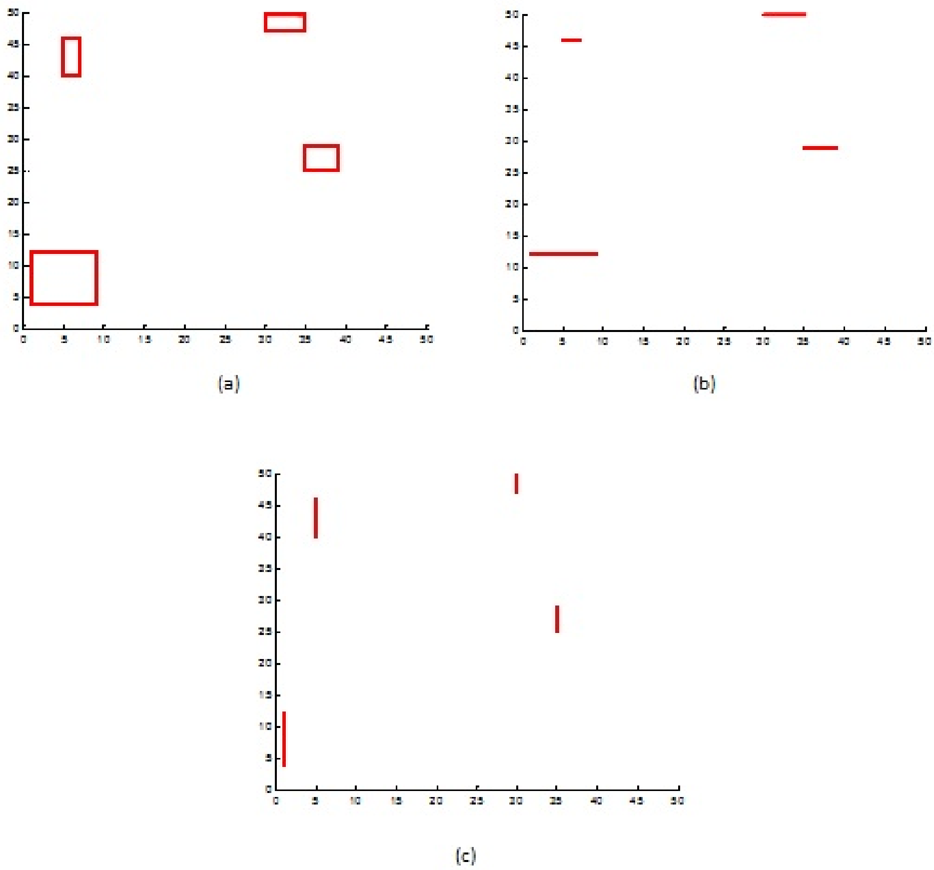

Figure 4.3: Single facility location problem on plane (a), on x coordinate (b) and on y coordinate (c)

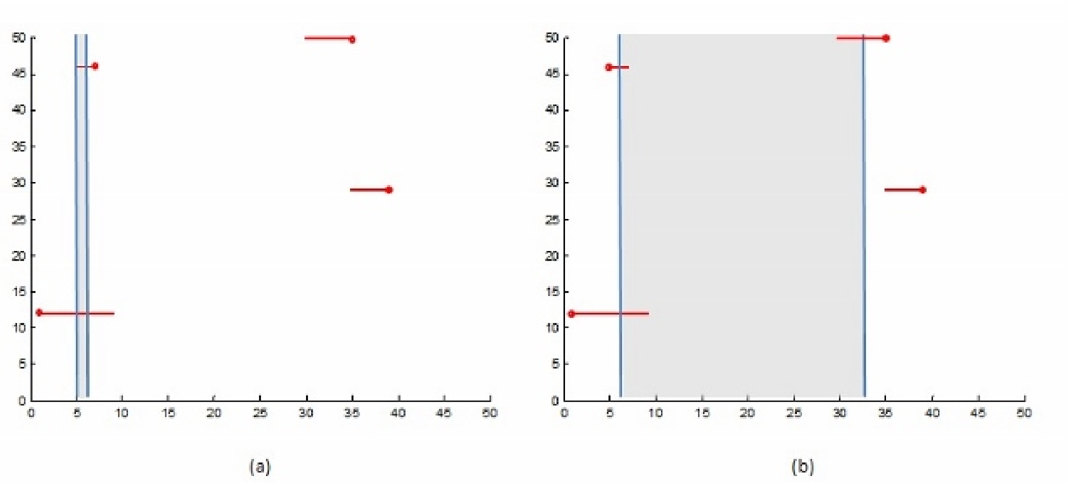

Figure 4.4: Finding optimal value of x coordinate

Details of Step 4 through Step 18 will be described in the following. First operation is the convertion of the problem into two problems on the line. For this purpose, a small example with 4 rectangular regions in Figure 4.3 can be investigated. In Figure 4.3(b), rectangular regions are converted to the line segments on the x coordinate. Taking south or north side of the rectangles does not change the solution. Because the only important thing is x coordinates of the end points of line segments. In 4.3(c), rectangular regions are converted to the line segments on the y coordinate. Again, it does not matter to take east or west side of the rectangles.

After converting the problem, the procedure "Optimal coordinate on an axis" is followed to find optimal locations of the x and y coordinates. For x coordinate, let us consider Figure 4.4. In Figure 4.4(a), algorithm starts with the shaded interval (c'_1, c'_2) which is the first interval on the left. If x coordinate of the facility is in this interval, we can find the farthest points of each line segments. They are marked in the figure. The average of x coordinates of these points is not in the shaded interval. Therefore, we continue with the next interval (c'_2, c'_3). It is shaded in Figure 4.4(b). For this interval, farthest points of the line segments are marked again. Average of these points is in the shaded interval. Now we can stop the procedure and conclude the optimal x coordinate of the facility for this iteration. To conclude the optimal y coordinate of the facility, same procedure is applied. After these operations for each single facility location problem, one location step is finished. The location and allocation steps are repeated until convergence is achieved or maximum number of iterations is reached.

Time complexity of location step of LBH is $O(m \log m)$. This complexity comes from sorting the line segments according to their center points. Although time complexity of location step of MP-H is smaller than LBH, LBH is experimentally faster than MP-H. It has time advantage over all of our three heuristics. It is the fastest heuristic we developed. However, it is only applicable for the special case we explain at the beginning of this section. Also, final locations of the facilities depends to initialization like other heuristics.

CHAPTER 5

APPROXIMATION OF AN ELLIPSE BY A POLYGON

In Chapter 1, we mentioned that instead of having exact locations of demanding entities we may have probability distributions of them. In this study, we restricted ourself with bivariate normal distribution. In Section 5.1, we define the relationship between the bivariate normal distribution and ellipses.

To apply our heuristics, demand regions should be polygons. Therefore, we developed a procedure to approximate elliptical regions with polygons. Details of this procedure is mentioned in Section 5.2.

5.1 Relationship between Bivariate Normal Distribution and Ellipse

Let X be $2x1$ random vector distributed according to a bivariate normal distribution for the x and y coordinates of the location of a demanding entity, μ be the mean vector and Σ be the variance-covariance matrix. The shorthand notation of the distribution is:

$$X \sim N(\mu, \Sigma)$$

The following is the explicit notation of the distribution.

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} \right)$$

where ρ is correlation coefficient, σ_1 is standard deviation of first random variable x and σ_2 is standard deviation of second random variable y.

Then, demanding entity can be represented by a $(1-\alpha)100\%$ prediction ellipse. By the prediction ellipse we mean that with $(1-\alpha)100\%$ probability all demand points will be inside the ellipse. The equation of the prediction ellipse which is centered at μ is given below.

$$(X-\mu)^T\Sigma^{-1}(X-\mu) \leq \chi^2_{2,\alpha} \tag{5.1}$$

where Σ^{-1} is the inverse of the variance-covariance matrix. It is calculated by the following formula.

$$\Sigma^{-1} = \frac{1}{\sigma_1^2\sigma_2^2(1-\rho^2)}\begin{bmatrix}\sigma_1^2 & -\rho\sigma_1\sigma_2 \\ -\rho\sigma_1\sigma_2 & \sigma_2^2\end{bmatrix} \tag{5.2}$$

The other parameters of the ellipse -length of the principal semi-axes and the direction of the principal axes- are directly related with the variance-covariance matrix. The length of the principal semi-axes is calculated from the eigenvalues of Σ. Let λ_1 and λ_2 be the eigenvalues. To find these values, we should solve the following equation.

$$|\Sigma - \lambda I| = 0 \tag{5.3}$$

Since Σ is 2x2 matrix, we end up with a polynomial of order 2. There are two roots of this polynomial, not necessarily each of them unique. These roots are the desired eigenvalues. Then, we can easily find the length of semi-axes from the eigenvalues by the following formula.

$$\sqrt{\lambda_i \chi^2_{2,\alpha}} \text{ for i=1,2} \tag{5.4}$$

The directions of the ellipse's principal axes is equal to eigenvectors of Σ. Let e_1 and e_2 be the eigenvectors. To find these vectors, we should solve the following set of equations for each eigenvalue.

$$(\Sigma - \lambda_i I)\, e_i = 0 \tag{5.5}$$

$$e_i^T e_i = 1 \tag{5.6}$$

Equation 5.6 is required to obtain a unique solution. Otherwise 5.5 does not lead to a unique solution alone.

In Figure 5.1, geometry of a prediction ellipse can be seen. In this figure, correlation coefficient is greater than zero. Therefore, the longest axis of the ellipse has a positive slope. Conversely, the longest axis of the ellipse has a negative slope when correlation coefficient is less than zero. If correlation coefficient is zero, instead of having ellipse we will have a circle. For more information [33] can be investigated.

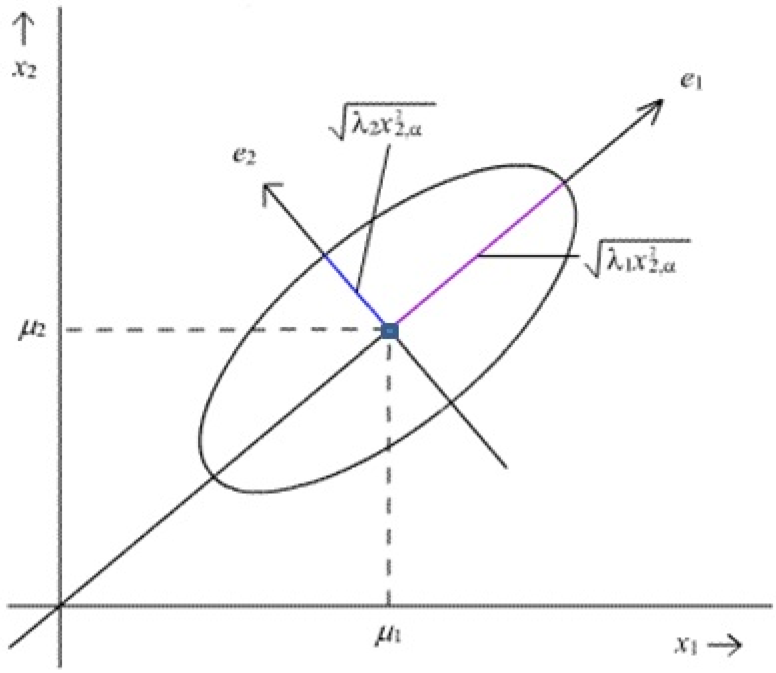

Figure 5.1: Representation of a prediction ellipse

5.2 Polygonal Approximation of an Ellipse

Bivariate normal distribution can be represented by ellipse as we mentioned in the previous section. We develepod our heuristics for polygonal regions, i.e. we need corner points to apply our heuristics. Therefore, we developed an approximation procedure. Basic steps of the procedure is given below.

1. Draw a smaller ellipse in the prediction ellipse with α_s.
2. Draw a larger ellipse out the prediction ellipse with α_l.
3. Draw a polygon between the bounds created by larger and smaller ellipses.

In Figure 5.2, an example for the approximation procedure is given. As the difference between α_s and α_l become smaller, smaller and larger ellipses in the figure get closer to the prediction ellipse. So the number of corners of the polygon increases and polygon approximation of the ellipse gets better. But computational time of our heuristics increase.

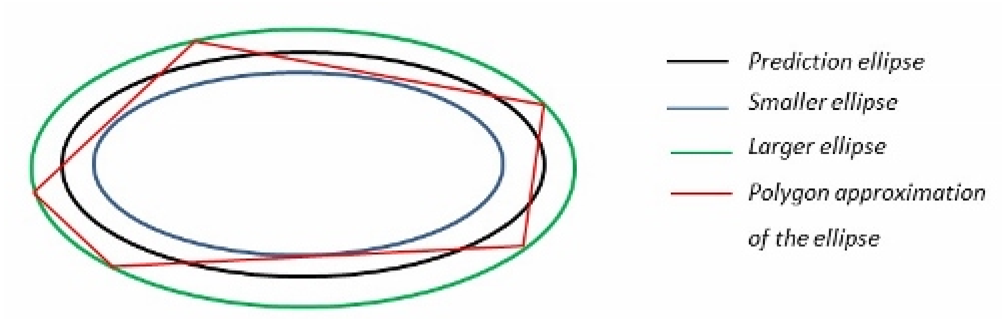

Figure 5.2: An example of polygon approximation to an ellipse

It should be noted that drawing a polygon between the bounds created is not an easy operation. To ease this operation we followed the following steps.

1. Transform all three ellipses in Figure 5.2 to circles by multiplying all the points on the ellipses by a transformation matrix, T.
2. Draw an equilateral polygon with the smallest possible number of corners between the bounds created by larger and smaller circles.
3. Retransform this equilateral polygon by multiplying the corners by the inverse of the transformation matrix, T^{-1}, and obtain a polygon between the bounds created by larger and smaller ellipses.

In Figure 5.3, easy way of drawing a polygon between the bounds created is illustrated.

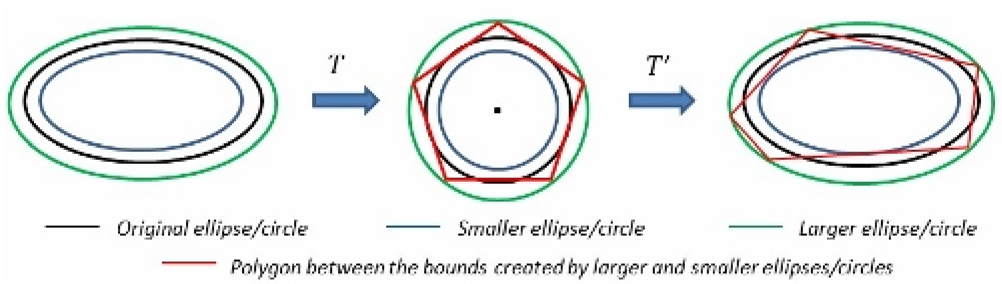

Figure 5.3: Procedure of drawing a polygon between the bounds created

Finding the transformation matrix, T, is an important operation. First thing required for this operation is a useful representation of an ellipse. There are many ways to represent an ellipse. These ways arise naturally in different cases [35].

Let the ellipse be centered at μ; let $2x2$ orthogonal matrix U be the matrix of the unit vectors in the directions of ellipse's principal axes; let A be the diagonal matrix with diagonal elements a_i such that $1/a_i$ is the length of the i^{th} principal semi-axis. Then, equation of the ellipse is given below.

$$(X-\mu)^T\, UAAU^T\, (X-\mu) \leq 1 \tag{5.7}$$

where (by using the findings in previous section), $U = [e_1, e_2]$, $A = \begin{bmatrix} \frac{1}{\sqrt{\lambda_1\, \chi^2_{2,\alpha}}} & 0 \\ 0 & \frac{1}{\sqrt{\lambda_2\, \chi^2_{2,\alpha}}} \end{bmatrix}$

If we rewrite A matrix as $A = \frac{1}{\sqrt{\chi^2_{2,\alpha}}} \begin{bmatrix} \frac{1}{\sqrt{\lambda_1}} & 0 \\ 0 & \frac{1}{\sqrt{\lambda_2}} \end{bmatrix}$ and let $B = \begin{bmatrix} \frac{1}{\sqrt{\lambda_1}} & 0 \\ 0 & \frac{1}{\sqrt{\lambda_2}} \end{bmatrix}$, ellipse equation 5.7 takes the following form.

$$(X-\mu)^T\, UBBU^T\, (X-\mu) \leq \chi^2_{2,\alpha} \tag{5.8}$$

Let $Y = BU^T X$ and $c = BU^T \mu$. Then, equation 5.8 can be written as follows.

$$(Y-c)^T (Y-c) \leq \chi^2_{2,\alpha} \tag{5.9}$$

Equation 5.9 is the equation of a circle centered at c and having radius $\sqrt{\chi^2_{2,\alpha}}$. In other words, we transformed the ellipse to a circle by multiplying all points on the ellipse, X, by transformation matrix, $T = BU^T$.

After the transformation procedure, next important operation is the drawing an equilateral polygon between the bounds created by larger and smaller circles. Lets say that we will divide the circles into equal parts with angle 2θ. The maximum value of 2θ occurs when the case is like in Figure 5.4. In this case, corners of the polygon are on the larger circle and the sides of the polygon are tangent to the smaller circle. By maximizing the value of 2θ, we minimize the number of corners of the polygon we want to find.

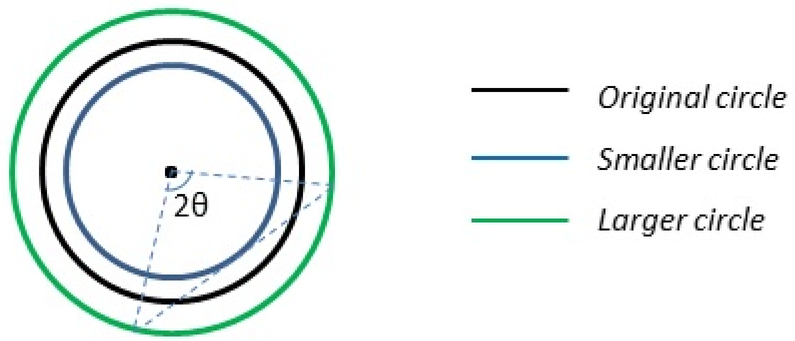

Figure 5.4: Procedure of drawing a polygon with minimum number of corners between the bounds created

The maximum value of θ is calculated by the following formula.

$$\sqrt{\chi_{2,\alpha_l}} \cos\theta = \sqrt{\chi_{2,\alpha_s}} \tag{5.10}$$

Then, the minimum possible number of corners of the polygon is equal to $\left\lceil \frac{360}{2\theta} \right\rceil$. When difference of α_s and α_l is small, degree of the approximation to the prediction ellipse and number of corners of the approximation polygon are high. Conversely, degree of the approximation to the prediction ellipse and the number of corners of the approximation polygon are low if difference of α_s and α_l is high. We chose to work with 95% accuracy for bivariate normal distribution, i.e. α for the prediction ellipses is equal to 0.05. In Table 5.1, some minimum number of corners required to obtain given approximation accuracy can be seen.

Table5.1: Number of minimum corners of the approximating polygon for a given approximation accuracy

α	α_s	α_l	Minimum number of corners
0.05	0.100	0.010	4
0.05	0.075	0.025	6
0.05	0.070	0.030	7
0.05	0.060	0.040	9
0.05	0.055	0.045	13

CHAPTER 6

COMPUTATIONAL STUDIES

There is no benchmark study for our problem in the literature. Authors who worked on demand regions choose their objective function so as to minimize maximum/minimum/average distance between the demand regions and facilities. However, we worked with the square of the distances. Therefore, we can only compare our heuristics with each other. We generated random problem instances. They are described in Section 6.1. In Section 6.2, parameters of the algorithms are set after some preliminary computational studies. Finally, in Section 6.3 with the set parameters computational results are given.

6.1 Problem Instances

To test our heuristics we randomly generate following two sets of problems.

1. All regions are rectangular regions with sides parallel to coordinate axes.
2. All regions are elliptical regions.

In each set there are 6 different problems with 5, 10, 20, 50, 100 and 200 demand regions. Problems with 5 demand regions are solved for 2, 3 and 4 facilities while problems with 10 demand regions are solved for 2, 3, 4 and 5 facilities. All of the remaining problems are solved for 2, 3, 4, 5 and 10 facilities. The summary of problem instances can be seen from Table 6.1.

We define first set of problems to be able to test our special heuristic, LBH, as well as our three generally applicable heuristics. When constructing a rectangular region, we started with randomly generating the coordinates of the southern west corner of the region from the discrete uniform distribution with the specified maximum value and 1 as a minimum value. Then, we randomly generated the side lengths of the rectangle from the discrete uniform distribution with the specified maximum value and 1 as a minimum value. By using the coordinates of southern west corner and side lengths, we constructed the other corners of the region. Parameters for generating the rectangular problem instances can be seen from Table 6.2.

We used second set of problems to compare our three heuristics, namely SOCP-H, MP-H, SBH, with each other when the demanding entities follow a bivariate normal distribution. For the bivariate normal distribution we need mean vector, standard deviations and correlation coefficient as explained in Chapter 5. We randomly generated elements of the mean vector and the standard deviations for variance-covariance matrix from the discrete uniform distributions with the specified maximum values, and 0 and 1 as the minimum values respectively. For the correlation coefficient we used continuous uniform distribution with -0.8 as minimum value and 0.8 as maximum value. We did not use -1 and 1, because when the absolute value of the correlation coefficient gets closer to 1 the shape of the prediction ellipse become more and more elongated. To be able to decrease the number of overlapped demand regions we used -0.8 and 0.8. Parameters for generating the bivariate normal distributions can be seen from Table 6.3.

The parameters of discrete and continuous uniform distributions are selected after preliminary trials. Aim in these trials is to decrease the number of overlapped demand regions.

We used 95% prediction ellipses to represent bivariate normal distributions. Then two level of accuracy to approximate prediction ellipses by a polygon are used; namely 6-corner polygon and 9-corner polygon approximations. The corresponding accuracy levels for these polygons can be seen from the Table 5.1.

Table6.1: Characteristics of the problem instances

Problem instance	Number of regions	Number of facilities
1	5	2
2		3
3		4
4	10	2
5		3
6		4
7		5
8	20	2
9		3
10		4
11		5
12		10
13	50	2
14		3
15		4
16		5
17		10
18	100	2
19		3
20		4
21		5
22		10
23	200	2
24		3
25		4
26		5
27		10

Table6.2: Parameters for rectangular region genaration

Number of the regions	Maximum value for x and y coordinate of the southern west corners	Maximum value for the side lengths
5	50	7
10	100	10
20	150	10
50	200	8
100	300	8
200	500	8

Table6.3: Parameters for bivariate normal distribution generation

Number of the regions	Maximum value for entities of the mean vectors	Maximum value for standard deviations
5	100	4
10	100	5
20	150	6
50	220	2
100	350	2
200	450	2

6.2 Parameter Settings

There are some parameters required to execute our algorithms. After some preliminary computational runs, the chosen parameters for each algorithm is summarized in Table 6.4.

Table6.4: Parameters of the algorithms

Algorithm	Max. number of iterations	Convergence condition	Other parameters
SOCP-H	30	≤ 0.01	-
MP-H	30	≤ 0.01	-
SBH	30	≤ 0.01	Smoothing parameters :$\epsilon = \tau = \mu = \gamma = 0.1$ Updating parameters: $\rho_1 = \rho_2 = \rho_3 = \rho_4 = 0.5$ Max. number of iterations of the main part :20 Max. number of iterations of unrestricted and bisection search in quasi-Newton method : 20 Step-size for unrestricted search : 0.0001 Convergence condition of bisection search :≤ 0.00001
LBH	30	≤ 0.01	-

30 is chosen as the maximum number of iterations of the algorithms. We also tried 50, but it did not change the solution qualities and computational times of the algorithms except MP-H. Algorithms except MP-H generally stops because of the convergence condition in earlier iterations as it can be seen in Table 6.10. On the other hand, for MP-H computational time (not the solution quality) is directly related with the number of maximum iterations since MP-H stops by reason of maximum number of iterations in general. This is the result of cycling. Cycling of the algorithm generally starts after 20 iterations. Therefore, to keep computational times low we select maximum number of iterations as 30 instead of 50.

As a convergence condition of the algorithms we set the rule such that if the difference between objective functions of two consecutive iterations is less than or equal to 0.01, stop. Also, we get computational runs with different values. However, when we consider the trade off between solution quality and computational time, 0.01 seemed more reasonable.

Other than these general parameters, there are some other parameters for SBH. The strategy in SBH is similar with the strategy in [43]. To lessen the preliminary computational runs, we borrowed the smoothing parameters and updating parameters from this paper. The values of the remaining four parameters yield better results than other values we tried when we considered the trade off between solution quality and computational time.

Also, it should be noted that to solve SOCP problems in our computational runs we used CVX system in MATLAB. It is a modelling system for constructing and solving convex programs. This system has four different solvers. Their capabilites to solve different convex programs are summarized in Table 6.5.

Table6.5: Capabilities of CVX solvers to solve different convex programs

Solver name	Convex program				
	LP	QP	SOCP	SDP	Integer
SeDuMi	Yes	Yes	Yes	Yes	No
SDPT3	Yes	Yes	Yes	Yes	No
Gurobi	Yes	Yes	Yes	No	Yes
MOSEK	Yes	Yes	Yes	No	Yes

In SOCP-H, we used SDPT3 solver, since it is more reliable than SeDuMi. Also, Gurobi and MOSEK are more limited solvers. For exact solutions, i.e. for mixed integer SOCP, we used Gurobi solver. We do not change the tolerance levels of the solvers. In CVX there are three tolerance levels and default values of them are as follows.

- $(2.22x10^{-16})^{1/2}$ for the solver tolerance that is requested by the solver.
- $(2.22x10^{-16})^{1/2}$ for the standard tolerance that is the level at which CVX considers the model solved to full precision.
- $(2.22x10^{-16})^{1/4}$ for the reduced tolerance. If this tolerance cannot be achieved, CVX returns a status of failed.

6.3 Computational Results of the Proposed Algorithms

All of the proposed heuristics and their subprocedures have been coded with MATLAB, version R2012a. After setting the algorithms' parameters, they have been run on a computer with Intel Core 2 CPU 6400 2.13 GHz processor and 1.75 GB RAM.

Since there is no benchmark study for our work, we need exact solutions of the problem instances to discuss effectiveness of our heuristics. The exact solutions can be found by mixed integer SOCP formulation proposed in Chapter 3. This formulation is weak. It does not even solve medium-size problems like problem with 20 demand regions and 5 facilities. The last solvable problem is the problem with 20 demand regions and 4 facilities. For other problems out of memory errors are occured because of too large branch-and-bound trees. In Table 6.6 and 6.7 optimal objective function values of solvable problem instances and their computation times are given respectively.

Table6.6: Optimal objective function values of small problem instances

Number of regions	Number of facilities	Rectangular regions	Elliptical regions	
			6-corner polygon	9-corner polygon
5	2	2185.5000	4564.5812	4456.3093
	3	1108.7500	2376.2447	2296.6779
	4	299.5000	741.2856	699.5564
10	2	6716.0000	8472.5469	8408.1255
	3	4231.2500	4091.7634	3958.3114
	4	2842.4166	2825.1871	2745.7280
	5	1845.3333	1967.1337	1896.8869
20	2	37463.5998	49364.3564	48768.9876
	3	19515.9443	33449.1913	32900.8599
	4	15376.4999	25217.4657	-

Table6.7: Time (in seconds) to find optimal solutions of small problem instances

Number of regions	Number of facilities	Rectangular regions	Elliptical regions	
			6-corner polygon	9-corner polygon
5	2	2185.5000	4564.5812	4456.3093
	3	1108.7500	2376.2447	2296.6779
	4	299.5000	741.2856	699.5564
10	2	6716.0000	8472.5469	8408.1255
	3	4231.2500	4091.7634	3958.3114
	4	2842.4166	2825.1871	2745.7280
	5	1845.3333	1967.1337	1896.8869
20	2	37463.5998	49364.3564	48768.9876
	3	19515.9443	33449.1913	32900.8599
	4	49381.0609	25796.8880	-

To obtain lower bounds for problem instances that cannot solvable with MISOCP formulation because of memory errors, we set 20 hours time limit. But solutions found within this time limit are worse than the solutions found with our heuristics. For example, the solution obtained for problem instance with 20 rectangular regions and 5 facilities within 20 hours has 52.8% optimality gap and solutions obtained with our heuristics are better. The larger the problem instances, the larger the optimality gap. Thus, we did not use solutions found within time limit as lower bounds.

As an alternative way, we consider the complete enumeration. Before starting the code we make the estimation about the computational time of the complete enumeration. For this purpose we need two things; namely, the number of possible ways of allocations of the regions to the facilities and computational time of finding optimal solution of an allocation. The number of ways to divide a set of m objects into q nonemty subsets can be calculated with *stirling numbers of the second kind* $S(m, q)$. The explicit formula of $S(m, q)$ is:

$$S(m, q) = \frac{1}{q!} \sum_{j=0}^{q-1} (-1)^j \binom{q}{j} (q - j)^m \tag{6.1}$$

[32] can be reviewed for the proof of the formula and further information.

For example, the number of possible allocations for the last solvable problem instance is $S(20, 4) = 45232115901$. With the mixed integer SOCP formulation, this problem is solved in 49381 seconds for rectangular regions as in Table 6.7. To compete with mixed integer SOCP formulation, we should find the optimal solution of approximately 916000 possible allocations in a second with complete enumeration. When we know the allocations of the regions, we have q single facility location problems to solve. This problems can be solved with an SOCP formulation proposed in Chapter 3. This formulation does not lead the required number of calculations in a second. For example, for the problem instance with 20 rectangular regions and 4 facilities SOCP-H finds solution in 13 seconds and 4.5 iterations on average, Table 6.9 and Table6.10. In other words, the optimal solution of an allocation is found in approximately 3 seconds. As you can guess, the required amount of calculations in a second for complete enumeration is not possible. Therefore, we did not try complete enumeration.

We have made 10 replications for each problem instance. To discuss effectiveness of our heuristics, we used percent deviations from exact solutions for problem instances that can be solvable with mixed integer SOCP. For other problems, we used percent deviations from the best solution found with our heuristics.

6.3.1 Rectangular regions with sides parallel to coordinate axes

Table 6.8, Table 6.9 and Table 6.10 are results for the data set with rectangular regions. In Table 6.8 percent deviations of best solutions found in 10 replicates are reported. Percent deviations of worst and average solutions found in 10 replicates are given in Appendix A and Appendix B respectively. In Table 6.9 total solution time of 10 replicates are reported. In

Table 6.10 average number of iterations in a replicate are stated.

Also, in Appendix C and Appendix D minimum and maximum solution times in 10 replicates are given.

When we look at the penultimate and last rows of Table 6.8, it can be seen that most robust heuristic for different type of problem instances is SBH. Its average and maximum deviations are close to each other. Other heuristics than SBH are not consistent for different problem instances according to these two rows. But only looking at these values may be deceptive. For some problem instances, starting points of heuristic may be too bad and so final solution of the heuristic may be too bad as a result. These too bad solutions may increase average deviations as outliers. Therefore, deviations for each problem instance should be investigated one by one like in Figure 6.1. In the figure, lines for SOCP-H and LBH coincides. These two heuristics generate same results as it can be seen in Table 6.8 and they generate considerably good results except for problem instances 5 and 17. On the other hand, results generated by MP-H is not as good as other heuristics. This may be due to cycling. To sum up, according to solution quality SBH, SOCP-H and LBH are reasonable heuristics and they are better than MP-H.

The location of facilities should be also investigated in addition to objective function values. For example, in Figure 6.2 locations found for problem instance with 5 rectangular regions and 3 facilities can be seen. SOCP-H, SBH and LBH find same locations for all facilities but MP-H cannot find the same location for one facility. Because in the optimal solution of this problem the demand region at southwestern has two farthest corners to the facility. Therefore, MP-H cannot find the optimal location of that facility and cycles. In Figure 6.3 locations found for problem instance with 10 rectangular regions and 5 facilities can be seen. SOCP-H, SBH and LBH find very similar locations for all facilities but MP-H cannot find the same locations with other heuristics except one facility. As in the previous example, in the optimal solution of the problem there are demand regions having two farthest corners to the facilities. Therefore, MP-H cannot find the optimal facility locations. Finally, in Figure 6.4 locations found for problem instance with 20 rectangular regions and 4 facilities can be seen. All of the heuristics find same locations for all facilities. Also, it should be noted that these results are consistent with the results in Table 6.8. Based on these examples, we can say that our heuristics except MP-H find very similar facility locations.

Table6.8: % Deviations (minimum of 10 replicates) from exact solution or best solution found of problems with rectangular regions

Number of regions	Number of facilities	Algorithms			
		SOCP-H	MP-H	SBH	LBH
5	2	0.0001	1.0295	0.0013	0.0000
	3	0.0000	4.1263	0.0003	0.0000
	4	0.0000	27.0451	0.0596	0.0000
10	2	0.0000	0.0000	0.0000	0.0000
	3	10.6068	2.3043	0.0746	10.6068
	4	0.0000	4.2745	0.0609	0.0000
	5	0.1355	12.2878	0.3227	0.1355
20	2	0.0000	0.0000	0.0000	0.0000
	3	0.0000	0.0000	0.0000	0.0000
	4	0.0000	0.0000	0.0000	0.0000
	5	0.0000	0.0000	0.0000	0.0000
	10	3.3519	16.4516	0.0000	3.3519
50	2	0.0000	0.0000	0.0000	0.0000
	3	0.0000	0.0043	0.0001	0.0000
	4	0.0000	0.0062	0.0001	0.0000
	5	1.7138	1.3021	0.0000	1.7138
	10	11.0351	5.6918	0.0000	11.0351
100	2	0.0000	0.0000	0.0000	0.0000
	3	0.0000	0.0000	0.0000	0.0000
	4	0.0000	0.0000	0.0000	0.0000
	5	0.3635	0.3640	0.0000	0.3635
	10	0.0000	0.8204	0.0435	0.0000
200	2	0.0000	0.0000	0.0000	0.0000
	3	0.0000	0.0000	0.0000	0.0000
	4	0.0106	0.0106	0.0000	0.0106
	5	0.0000	2.2662	0.0486	0.0000
	10	0.0446	1.5650	0.0000	0.0446
Average deviation		1.0097	2.9463	0.0227	1.0097
Maximum deviation		11.0351	27.0451	0.3227	11.0351

Table6.9: The computation time (in seconds) to obtain solutions in Table 6.8

Number of regions	Number of facilities	Algorithms			
		SOCP-H	MP-H	SBH	LBH
5	2	24,9110	0,0803	3,9497	0,0695
	3	27,9483	0,1760	5,4457	0,0267
	4	28,2968	0,2034	10,1858	0,0406
10	2	36,3307	0,1028	4,3029	0,0382
	3	47,7689	0,2413	10,4476	0,0511
	4	62,7708	0,3908	18,7433	0,0705
	5	70,4925	0,4535	19,9420	0,0819
20	2	82,9654	0,0739	7,3979	0,1012
	3	139,6744	0,1127	22,3425	0,1564
	4	121,4162	0,4365	31,4296	0,1512
	5	138,7831	0,5727	37,0342	0,1756
	10	171,0821	1,4592	158,2972	0,3301
50	2	261,0745	0,2363	30,8934	0,3526
	3	354,9161	0,9146	58,9222	0,4929
	4	425,5047	0,8201	71,2029	0,5850
	5	461,0334	1,1593	108,3299	0,6399
	10	735,3288	3,3290	273,5900	1,2088
100	2	670,1187	0,5326	56,6597	0,7977
	3	892,6176	1,1976	72,7703	1,2111
	4	1002,4766	1,3961	126,5905	1,5529
	5	1092,5923	3,2148	195,2714	1,6620
	10	1326,8167	4,2067	471,9317	2,4296
200	2	1560,0911	3,1794	125,6423	2,4491
	3	1604,1073	1,6023	165,6391	2,2436
	4	2037,9926	4,7371	254,2409	2,7837
	5	2812,2265	4,2522	399,5727	4,2693
	10	2907,0245	8,8838	817,3341	5,8232

Table6.10: Average number of iterations in a replicate for problems with rectangular regions

Number of regions	Number of facilities	Algorithms			
		SOCP-H	MP-H	SBH	LBH
5	2	3	16.9	5.4	3
	3	3.1	30	8.5	3.1
	4	3	30	17.5	3
10	2	3.1	12.3	5.9	3.1
	3	3.4	22.4	9	3.4
	4	3.8	30	13.7	3.8
	5	3.9	30	13.8	3.9
20	2	4.1	5.3	4.7	4.1
	3	5.4	6.2	9.2	5.4
	4	4.5	18.1	11.4	4.5
	5	4.6	20.1	11	4.6
	10	4.4	30	29.7	4.4
50	2	5.5	6.7	7.8	5.5
	3	6.7	18.7	12.1	6.7
	4	7.4	14.2	10.6	7.4
	5	7.4	17	14.1	7.4
	10	8.9	27.9	19.7	8.9
100	2	6.5	7.5	8.6	6.5
	3	8.5	12.4	7.8	8.5
	4	9.3	12.4	9.7	9.3
	5	9.7	23.5	12.7	9.7
	10	9.6	18.2	15.6	9.6
200	2	9.6	20.2	10.7	9.6
	3	7.9	8.8	9.1	8
	4	8.9	19.4	11.9	8.9
	5	12.3	15.9	13.5	12.3
	10	11.6	19.3	14	11.6

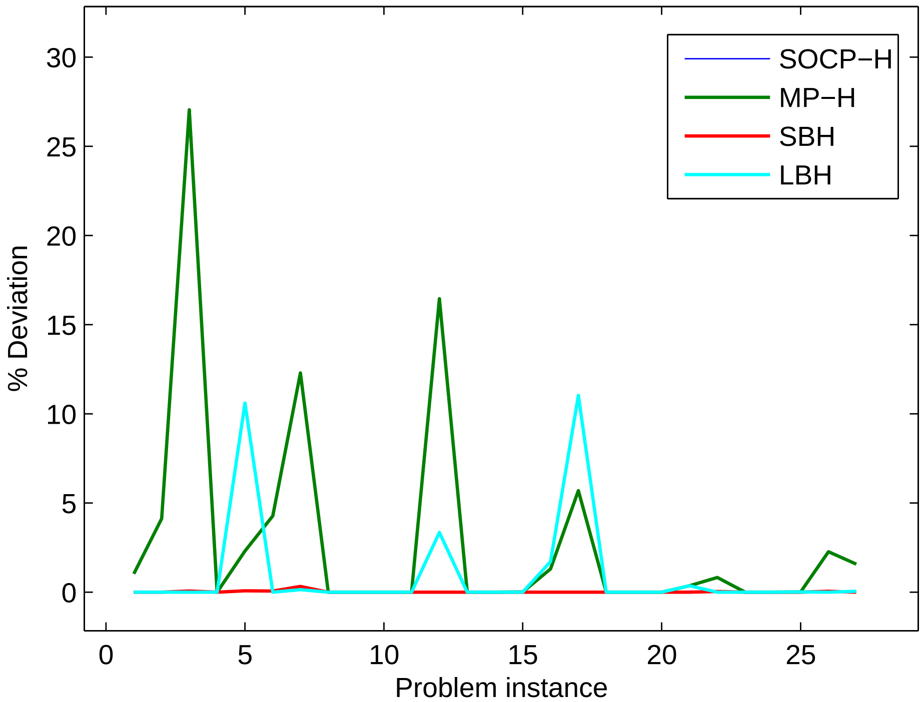

Figure 6.1: % Deviations of heuristics with respect to different problem instances

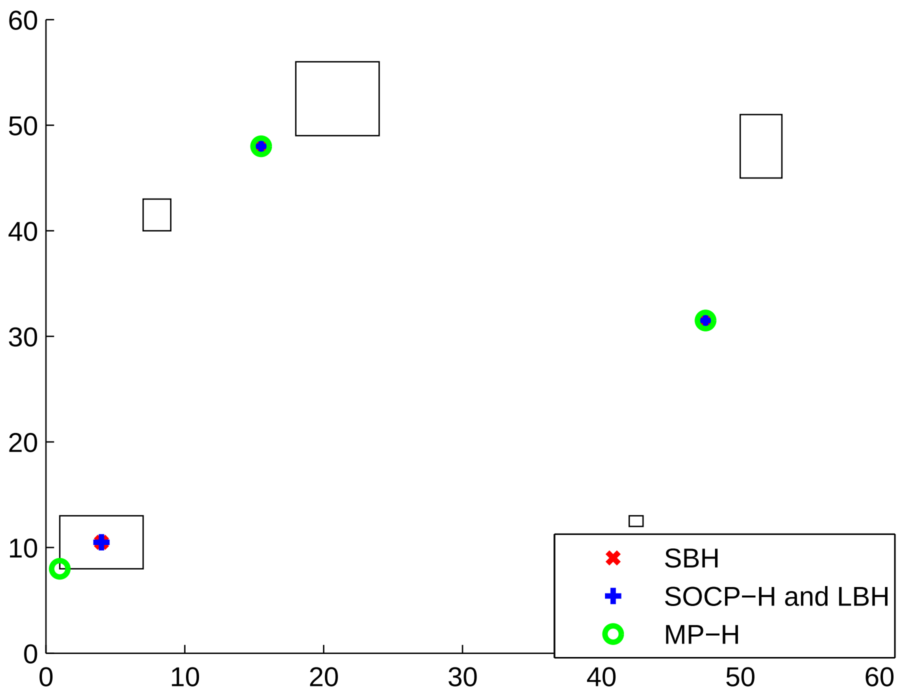

Figure 6.2: % Facility locations found with heuristics for problem instance with 5 rectangular regions and 3 facilities

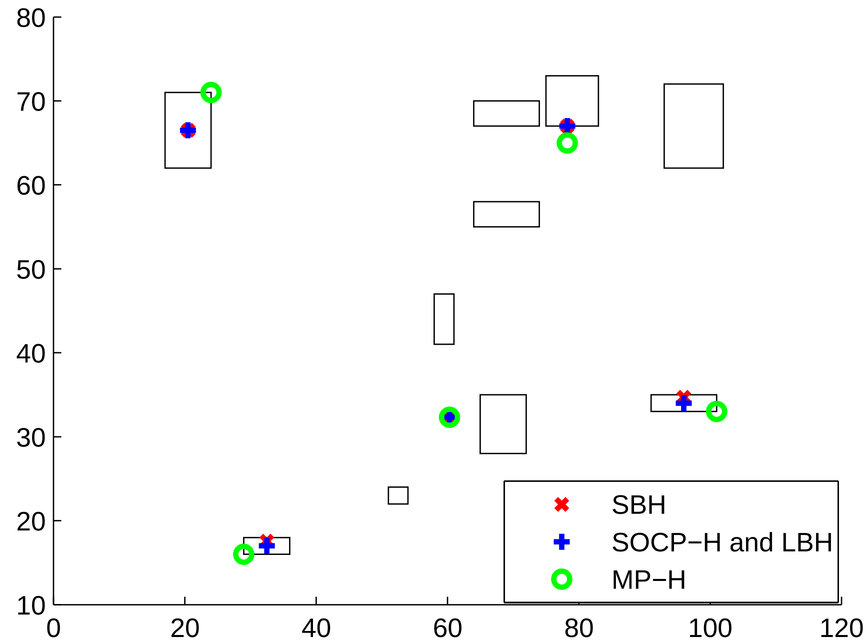

Figure 6.3: % Facility locations found with heuristics for problem instance with 10 rectangular regions and 5 facilities

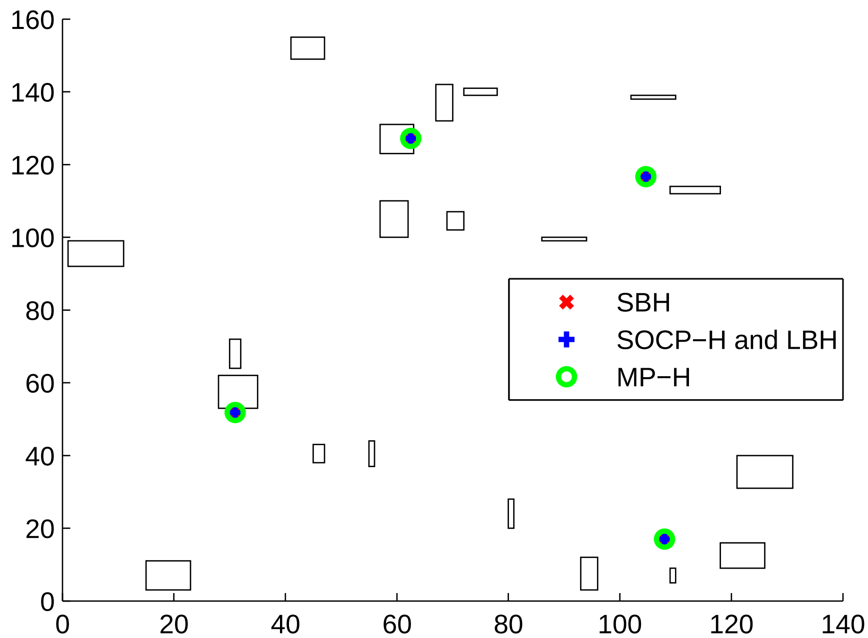

Figure 6.4: % Facility locations found with heuristics for problem instance with 20 rectangular regions and 4 facilities

When we consider solution times, SOCP-H is the slowest heuristic as it can be seen in Figure 6.5. Even with this slowest heuristic we can solve large problem instances in 5 minutes. SOCP-H is followed by SBH. In the same figure, lines for MP-H and LBH seem to coincide. When we zoom in as in Figure 6.6, the fastest heuristic is LBH and MP-H is the second fastest one.

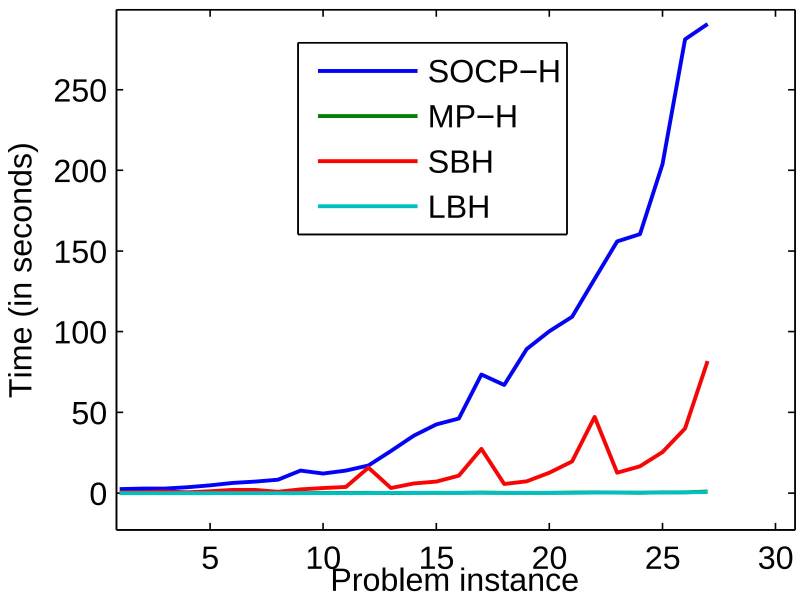

Figure 6.5: Solution times of heuristics with respect to different problem instances

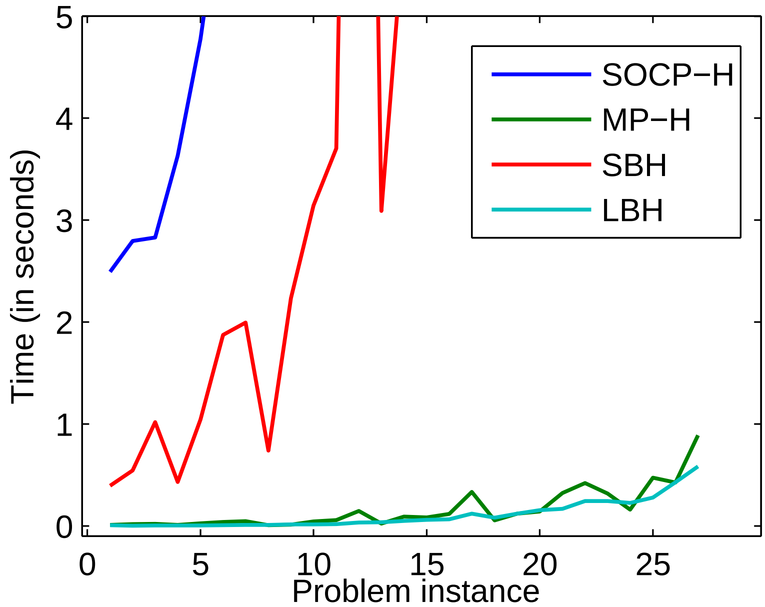

Figure 6.6: Zoomed version of Figure 6.5

The change in solution times of our heuristics with respect to the number of regions and the number of facilities is also important. In Figures 6.7, 6.8, 6.9 and 6.10 these changes can be seen. By looking at these figures, following two results are obtained.

- When the number of regions is constant, solution time increases with the number of facilities in SOCP-H, SBH and LBH. The same inference cannot be made for MP-H.
- When the number of facilities is fixed, solution time increases with the number of regions in all heuristics.

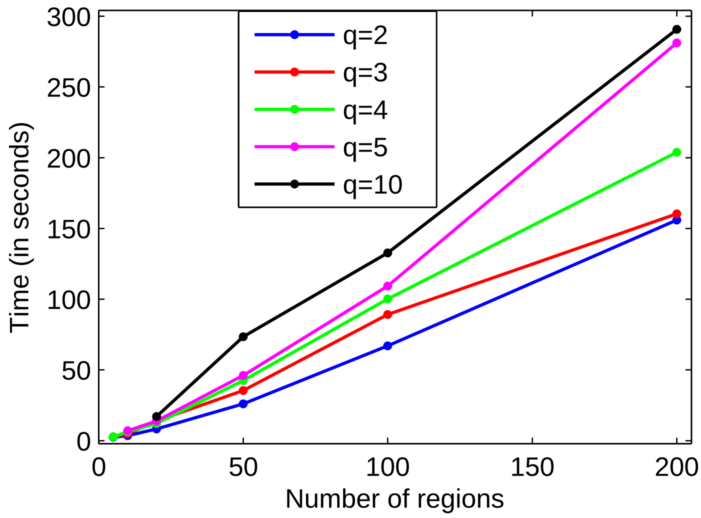

Figure 6.7: Change in solution time of SOCP-H with respect to the number of regions and the number of facilities

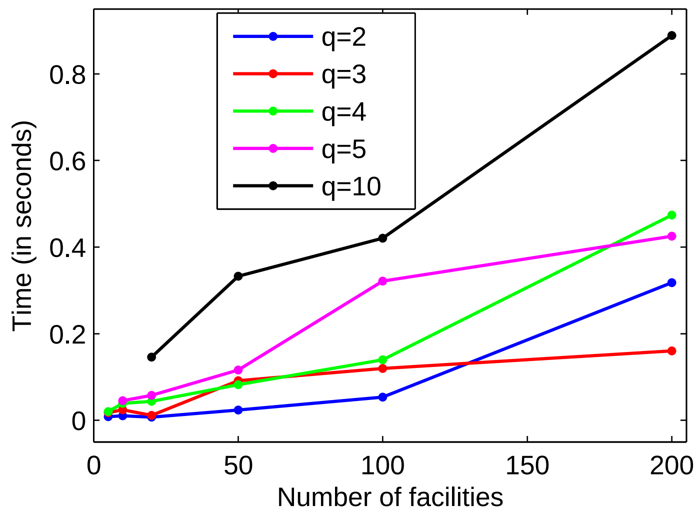

Figure 6.8: Change in solution time of MP-H with respect to the number of regions and the number of facilities

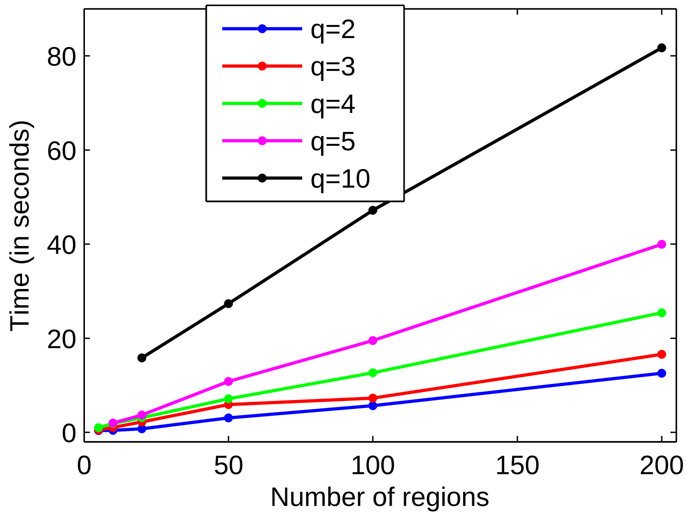

Figure 6.9: Change in solution time of SBH with respect to the number of regions and the number of facilities

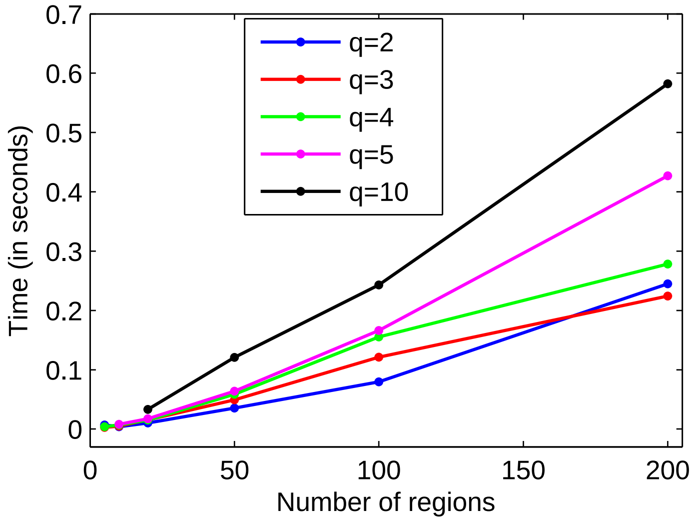

Figure 6.10: Change in solution time of LBH with respect to the number of regions and the number of facilities

Also, how many times our heuristics finds good solutions among many replications is another important issue. For example, in Figure 6.11 behaviour of heuristics for problem instance with 10 rectangular regions and 5 facilities can be seen. We made 150 replications. All of our heuristics find near optimal solutions most of the time. In Figure 6.12 behaviour of heuristics for problem instance with 50 rectangular regions and 5 facilities can be seen. The results are not as good as in the previous example. However, histograms are right skewed as it can be seen. Our heuristics finds better solutions most of the times.

To sum up, best heuristic is LBH for the problem instances in which all of the demand regions are rectangular regions with sides parallel to coordinate axes. The solution quality of it is satisfactory for the most of the problem instances, it is the fastest heuristic and behaviour of the heuristic towards the number of regions and facilities is predictable. One can say that when we consider the solution quality LBH is not as robust as SBH heuristic and its percent deviation is high for some problem instances. But solution time of LBH is approximately one hundredth of solution time of SBH. By increasing the number of replications in LBH, solution quality may be further increased.

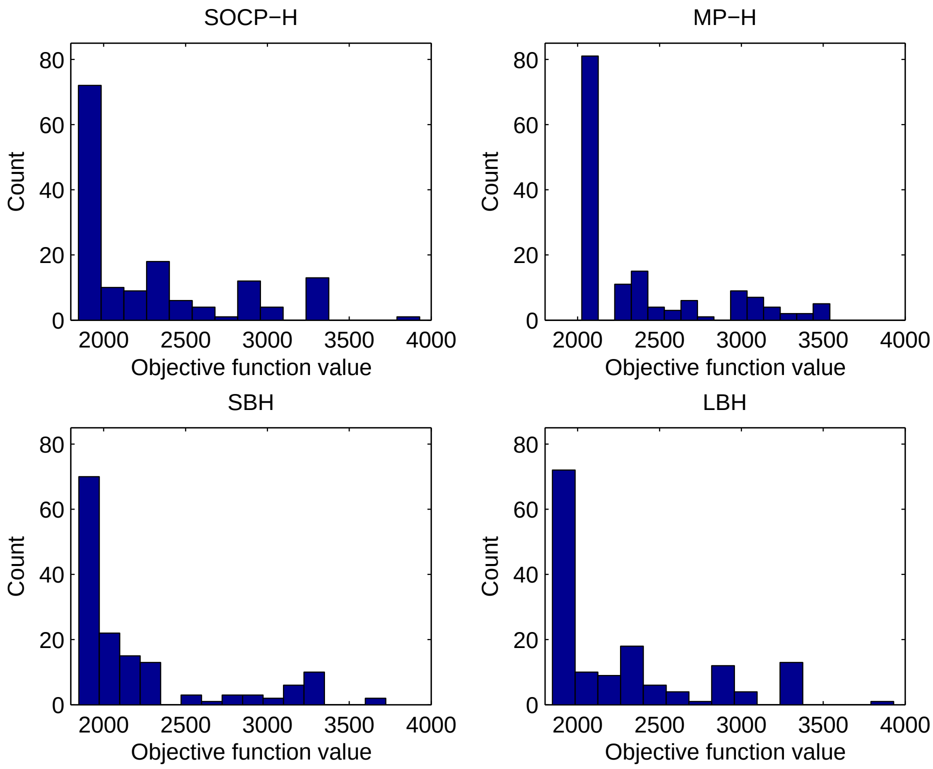

Figure 6.11: Behaviour of heuristics towards finding good solutions for proplem instance with 10 rectangular regions and 5 facilities

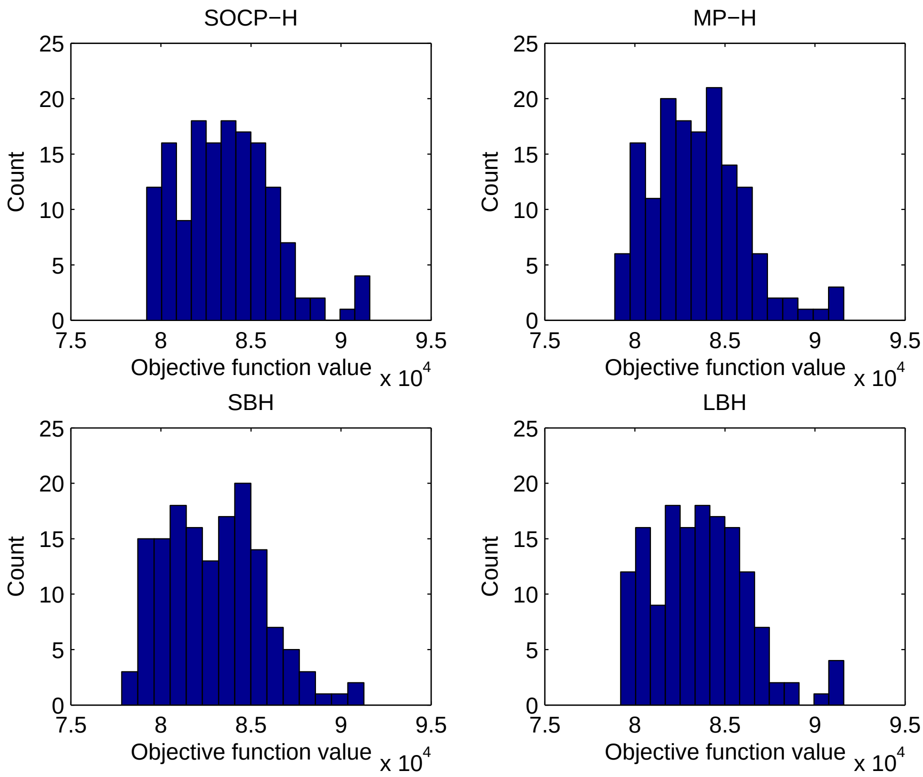

Figure 6.12: Behaviour of heuristics towards finding good solutions for proplem instance with 50 rectangular regions and 5 facilities

6.3.2 Elliptical regions

As we said before, we used 6-corner polygon and 9-corner polygon approximation to elliptical regions. Solution quality of heuristics for both accuracy level of approximation is similar with the performances for rectangular regions. This can be seen from Figures 6.13 and 6.14. Like in the previous subsection, SOCP-H and SBH generate similar and good results. They are better than MP-H. Also, when we consider the locations of facilities, SOCP-H and SBH locate facilities very similar and locations found by MP-H are usually different than locations found by other heuristics.

When we consider the solution times, MP-H is the fastest heuristic and it is followed by SBH. The slowest one is SOCP-H as it can be seen in Figures 6.15 and 6.16.

Even with the slowest heuristic, we can solve large problem instances in 15 minutes for high level of accuracy (9-corner polygon approximation). Behaviour of heuristics towards to change in the number of regions and facilities is the same with the results in Section 6.3.1. When the number of regions or the number of facilities increase, the solution times of heuristics also increase in general.

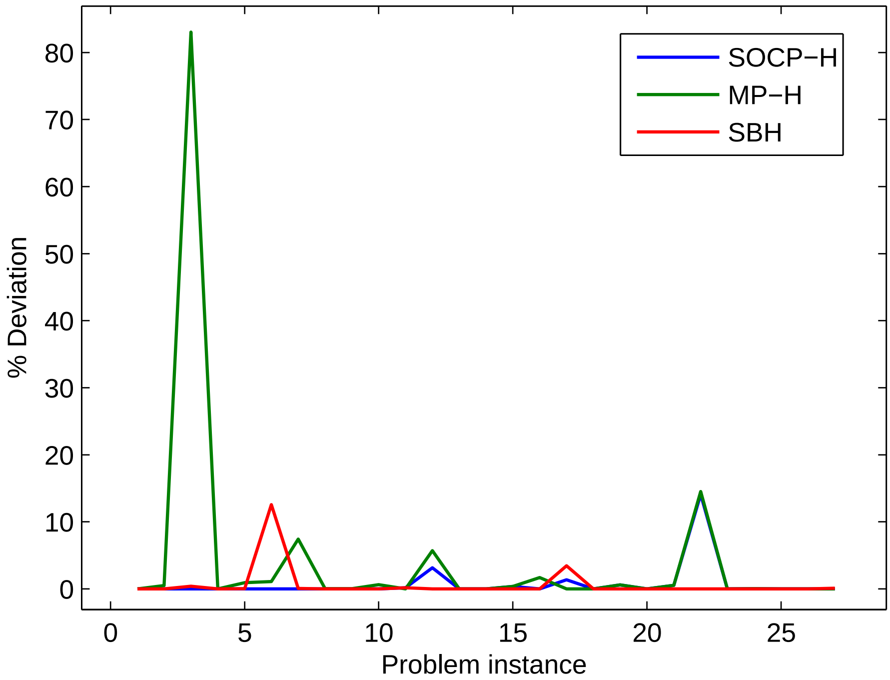

Figure 6.13: % Deviations of heuristics with respect to different problem instances with 6-corner polygon approximation to the elliptical regions

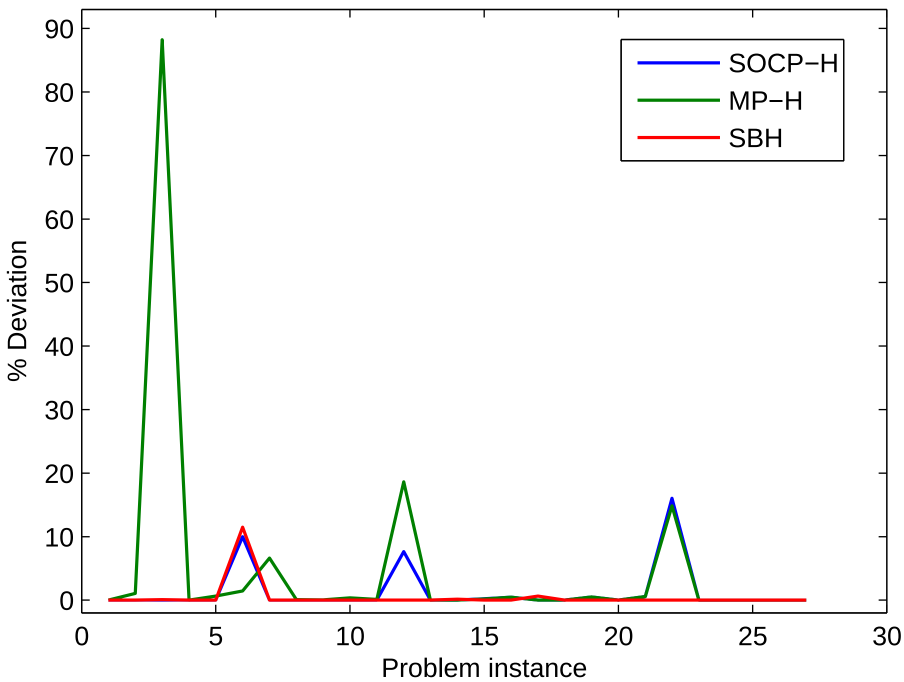

Figure 6.14: % Deviations of heuristics with respect to different problem instances with 9-corner polygon approximation to the elliptical regions

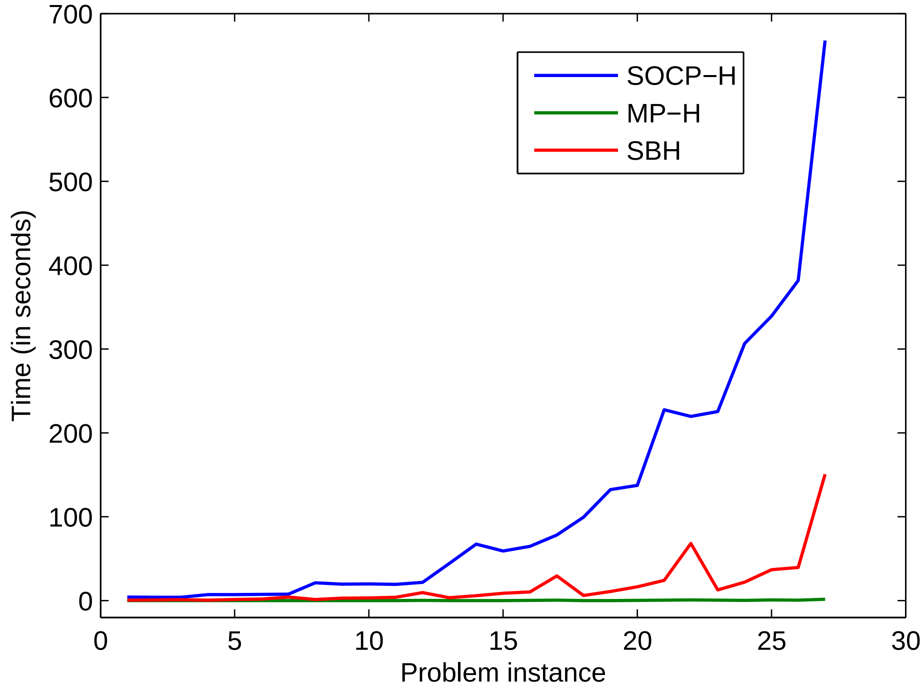

Figure 6.15: Solution times of heuristics with respect to different problem instances with 6-corner polygon approximation to elliptical regions

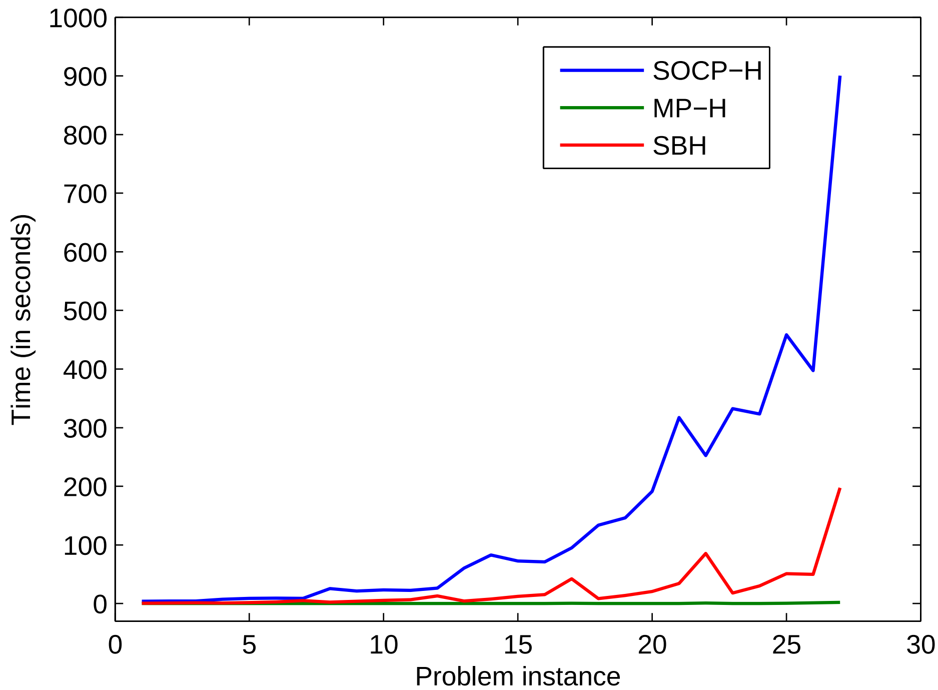

Figure 6.16: Solution times of heuristics with respect to different problem instances with 9-corner polygon approximation to elliptical regions

In Appendix E, percent deviations of best solutions found in 10 replicates, total solution time of 10 replicates and average number of iterations in a replicate are given for both accuracy level of approximation.

Finally, it should be noted that when the level of approximation accuracy increase, the solution times of heuristics increase. In other words, larger the number of corners of the approximation polygon, higher the computation time as it can be seen in the Figure 6.17. One can increase or decrease the level of accuracy as desired in exchange for computational time.

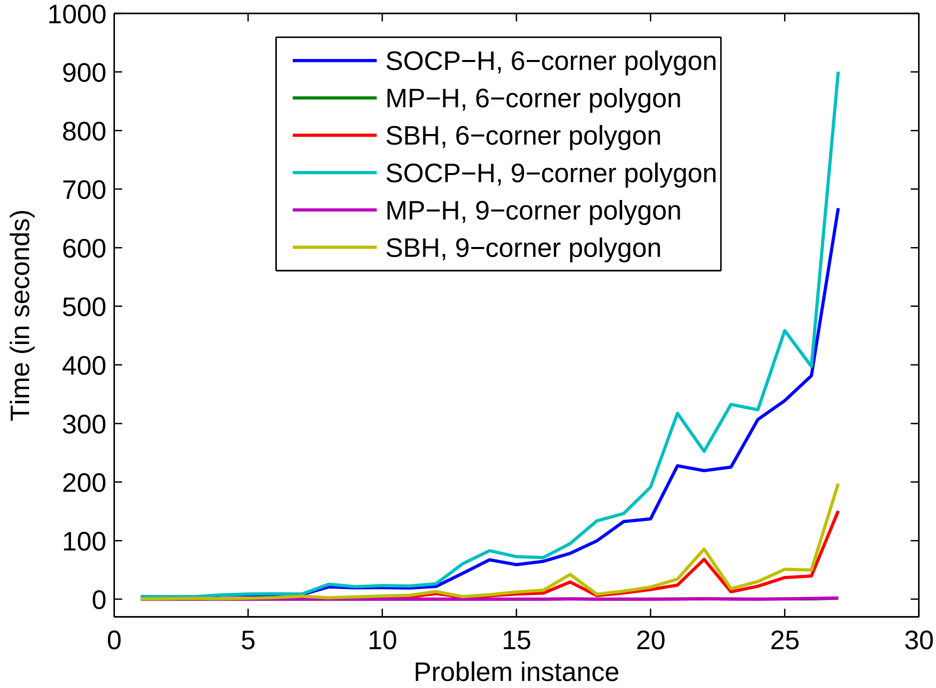

Figure 6.17: Solution times of heuristics with respect to different problem instances for two level of accuracy

To conclude, SBH is the best heuristic in general problems with polygonal regions. It generates considerably good results in reasonable solution times.

CHAPTER 7

CONCLUSION AND FURTHER RESEARCH DIRECTIONS

Location models have been studied for a long time. They have been extensively discussed by a variety of disciplines and they are taught in university departments of mathematics, management science, computer science, geography etc. [16]. Modern location theory is assumed to begin with Alfred Weber's thesis in 1909 [19].

The facility location problem can be briefly defined as follows. To serve m demanding entities, locate q facilities so as to optimize a given objective. There are many variants of facility location models since they are used in several different research fields. The characteristics of the problem we worked on in this study are as follows.

- We want to locate q facilities where $q > 1$ to serve m polygonal demand regions on the plane instead of demand points.
- The location of a facility can be anywhere on the plane.
- Our objective is to minimize the sum of the squares of maximum Euclidean distances between facilities and demand regions.
- The distance calculation is done by calculating the farthest distance between the demand region and the facility it is allocated to.
- Each demand region can be assigned to exactly one facility.
- Each demand region has equal weights.

There are mainly three motivations behind working with the demand regions. First of all, in some situations, the size of demanding entities cannot be negligible like in the problem of locating emergency centers to serve districts or cities. Second of all, the location of demanding entities may not be deterministic. They can follow a bivariate distribution on the plane. Finally, there may be too many demand points in some problems. To treat each of them one by one makes problems inconvenient. In such situations, representing demand as an area may be convenient.

As we explained in previous chapters. There are three ways of calculating distances between the demand regions and the facilities; namely farthest, closest and expected distance. Our motivation in using farthest distance is the desire of optimizing the worst-case scenarios.

It should be also noted that our mathematical programming formulations and heuristics do not include weight parameters since we assumed that each demand region has equal weights. If this is not the case and demand regions have different weights, our mathematical programming formulations and heuristics can be easily modified. For the mathematical programming formulations in Chapter 3, multiplying each term in objective function by corresponding weights is required. For SOCP-H, instead of using model 3.6 in Step 5, using SOCP formulation that minimizes the weighted sum of distance squares is enough. For MP-H and LBH, we should take the weighted average of farthest points in location steps instead of taking just average. Finally, for SBH, each term in objective function should be multiplied with corresponding weights in model 4.4. So the only change is in the first order derivative of the objective function. All of the procedures of SBH -smoothing, transformation, quasi-Newton- remain the same.

First, we give SOCP formulation of the single facility location problem. Thereby, we showed that the single facility location problem can be solved in polynomial time since SOCP problems are solvable in polynomial time. Then, we propose a mixed integer SOCP formulation for our problem. Inability of the formulation to solve even medium size problems leads us to find heuristics. We propose three heuristics for our problem. All of our heuristics can be modified to minimize the sum of maximum Euclidean distances between facilities and demand regions. We also mentioned briefly these modified versions. Also, we develope a special heuristic for the case when the demand regions are of rectangular shape. To test effectiveness of our special heuristic we conducted a computational study with rectangular demand regions. Except that we restrict ourselves to the case where demand regions follows bivariate normal distributions. All of our heuristics are developed for polygonal regions, i.e. we need corner points of demand regions. Therefore, we mention the relation between bivariate normal distribution and ellipse in the study. After that, we explained our approach to approximate elliptical regions with polygons that allows us to use our heuristics.

To the best of our knowledge, there is no benchmark study for our work. Therefore, we compare our heuristics with each other. To test the performances of the heuristics we generated random problem instances. For the special problem instances with rectangular regions having sides parallel to the coordinate axes, best heuristic is the LBH. Its percent deviation from exact solution or best solution found with our heuristics is less than 0.5% for 23 out of 27 problem instances. And it is the fastest heuristic. Indeed, the solution time of LBH is one hundredth of the second fastest heuristic. The time advantage of LBH makes possible to further increase the solution quality of it by increasing the number of replications. Also, with the predictability of LBH's solution time with regards to change in the number of regions and facilities, it becomes even more remarkable. For the general problem instances with bivariate normally distributed regions, the best heuristic is SBH. We used two level of accuracy of approximation for these problem instances; namely 6-corner polygon approximation and

9-corner polygon approximation. For the first approximation level SBH's percent deviation is less than 1% for 25 out of 27 problem instances. It is less 1% for 26 out of 27 problem instances for the second approximation level. Its solution time is also reasonable. Even for the largest problem instance we generated only 3 minutes are required to find solution. Like LBH for special problem instances, behaviour of the solution time of SBH towards change in the number of regions and facilities is foreseeable.

There are a number of future research directions. They can be listed as follows.

- Other than bivariate normal distirbution, different bivariate distributions can be used.
- The assumption that each demand region can be assigned to exactly one facility may be relaxed. Demand regions may be assigned to more than one facilities.
- Instead of using farthest distance between the demand region and the facility, closest or expected distance may be used.
- Instead of minimizing the sum of distance squares, other functions of the distances can be used. For this objective, we have ideas on how to update our heuristics as mentioned in the study.
- Problem instances with national barriers can be considered. In other words, algorithms can be modified such that facility locations are restricted with some constraints.
- Instead of locating point facilities, locating regional facilities can be studied. After selecting an appropriate distance calculation method for distance between regional demanding entities and regional facilities, algorithms can be modified accordingly.
- To increase the effectiveness of our heuristics, instead of using arbitrary initial facility locations a method for choosing starting facilities may be developed.
- A procedure can be found to prevent cycling of MP-H.

REFERENCES

[1] Alizadeh, F., Goldfarb, D. *Second order cone programming [PDF document]*. Retrieved from: http://retis.sssup.it/ bini/teaching/optim2010/cone-programming.pdf

[2] Aloise, D., Deshpande, A., Hansen, P., Popat, P., *Np-hardness of Euclidean sum-of-squares clustering*, Machine Learning **75** (2009), 245-248.

[3] Aly, A.A., Marucheck, A.S., *Generalized weber problem with rectangular regions*, The Journal of the Operational Research Society **33** (1982), 983-989.

[4] Bennett, C.D., Mirakhor, A., *Optimal facility location with respect to several regions*, Journal of Regional Science **14** (1974), 131-136.

[5] Brimberg, J., Wesolowsky, G.O., *Locating facilites by minimax relative to closest points of demand areas*, Computers and Operations Research **29** (2002), 625-636.

[6] Brimberg, J., Wesolowsky, G.O., *Minisum location with closest euclidean distances*, Annals of Operations Research **11** (2002), 151-165.

[7] Carrizosa, E., Conde, E., Munoz-Marquez, M., Puerto, J., *The generalized weber problem with expected distances*, RAIRO Operations Research **29** (1995), 35-57.

[8] Carrizosa, E., Munoz-Marquez, M., Puerto, J., *The weber problem with regional demand*, European Journal of Operational Research **104** (1998), 358-365.

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

[10] Cooper, L., *Location-allocation problems*, Operations Research **11** (1963), 331-343.

[11] Cooper, L., *Heuristics methods for location-allocation problems*, SIAM Reviews **6** (1964), 37-53.

[12] Cooper, L.,*A random locational equilibrium problem*, Journal of Regional Sciences **14** (1974), 47-54.

[13] Daskin, M.S. (1995). *Network and discrete location:Models, algorithms, and applications*. New York: Wiley.

[14] Drezner, Z. (1995). *Facility location:A survey of applications and methods*. New York:Springer.

[15] Drezner, Z., Wesolowsky, G.O., *Location models with groups of demand points*, INFOR **38** (2000), 359-372.

[16] Drezner, Z., Hamacher, H.W.(2002). *Facility location:Applications and theory*. Berlin ; New York : Springer.

[17] El-Shaieb, A.M., *The single source weber problem:Survey and extensions*, The Journal of the Operational Research **29** (1978), 469-476.

[18] Eyster, J., White, J., Wierwille, J., *On solving multifacility location problem using a hyperboloid approximation procedure*, AIIE Transactions **5** (1973), 1-6.

[19] Farahani, R.Z., Hekmarfar, M. (2009). *Facility location:Concepts, models, algorithms and case studies*. Verlag Berlin Heidelberg:Springer.

[20] Guccione, A., Gillen, W., *An economic interpretatiton of Kuhn's dual for the steiner-weber problem:A note*, Journal of Regional Sciences **31** (1991), 93-95.

[21] Hadley, G. (1964). *Nonlinear and dynamic programming*. United States of America: Addison-Wesley.

[22] Hale, T.S., Moberg, C.R., *Location science research:A review*, Annals of Operations Research **123** (2003), 21–35.

[23] Hillsman, E.L., Rhoda, R., *Errors in measuring distances from populations to service centers*, The Annals of Regional Sciences **12** (1978), 74-88.

[24] Jiang, J., Yuan, X., *Minisum location problem with farthest euclidean distances*, Mathematical Methods of Operations Research **64** (2006), 285-308.

[25] Jiang, J., Yuan, X., *A barzilai-borwein-based heuristic algorithm for locating multiple facilities with regional demand*, Computational Optimization and Applications **51** (2012), 1275-1295.

[26] Kuhn, H.W., *A note on Fermat's problem*, Mathematical Programming **4** (1973), 98-107.

[27] Lobo, M.S., Vandenberghe, L., Boyd, S., Lebret, H., *Applications of second-order cone programming*, Linear Algebra and Its Applications **284** (1998), 193-228.

[28] Love, R.F., *A computational procedure for optimally locating a facility with respect to several rectangular regions*, Journal of Regional Sciences **12** (1972), 233-242.

[29] Luangkesorn, L. (2009). *Weber problem [PDF document]*. Retrieved from: http://www.pitt.edu/ lol11/ie1079/notes/ie2079-weber-slides.pdf

[30] Megiddo, N., Supowit, K.J., *On the complexity of some common geometric location problems*, SIAM Journal on Computing **13** (1984), 182-196.

[31] Melo, M.T., Nickel, S., Saldanha-da-Gama, F., *Facility location and supply chain management-A review*, European Journal of Operational Research **196** (2009), 401–412.

[32] Moll, V.H. (2012). *Numbers and functions:From a classical-experimental mathematician's point of view*. United States of America:AMS.

[33] *Multivariate normal distribution*. Retrieved from Lecture Notes Online Web site: https://onlinecourses.science.psu.edu/stat505/node/12

[34] Owen, S.H., Daskin, M.S., *Strategic facility location:A review*, European Journal of Operational Research **111** (1998), 423-447.

[35] Pope, S.B. (2008). *Algorithms for ellipsoids*, Cornell University report FDA-08-01.

[36] Rao, S.S. (1996). *Engineering optimization: Theory and practice*. New York:Wiley.

[37] ReVelle, C.S., Eiselt, H.A., *Location analysis:A synthesis and survey*, European Journal of Operational Research **1165** (2005), 1-19.

[38] Rosen, J.B., Xue, G.L., *On the convergence of a hyperboloid approximation prcedure for the perturbed euclidean multifacility location problem*, Operations Research **41** (1993), 1164-1171.

[39] Suzuki, A., Drezner, Z., *The p-center location problem in an area*, Location Science **4** (1996), 69-82.

[40] Tellier, L., *The weber problem : Solution and interpretation*, Geographical Analysis **4** (1972), 215-233.

[41] Weiszfeld, E., *Sur le point pour lequel la somme des distances de n points donnes est minimum*, The Tohoku Mathematical Journal **43** (1936), 355-386.

[42] Wesolowsky, G.O., Love, R.F., *A nonlinear approximation method for solving a generalized rectangular distances*, Management Science **18** (1972), 656-663.

[43] Xavier, A.E., *The hyperbolic smoothing clustering method*, Pattern Recognition **43** (2010), 731-737.

APPENDIX A

COMPARISON OF HEURISTICS ACCORDING TO MAXIMUM OBJECTIVE VALUE OF 10 REPLICATES

TableA.1: % Deviations (maximum of 10 replicates) from exact solution or best solution found of problems with rectangular regions

Number of regions	Number of facilities	Algorithms			
		SOCP-H	MP-H	SBH	LBH
5	2	33.7833	33.7833	33.7833	33.7833
	3	41.6685	47.0575	41.6857	41.6685
	4	187.3957	227.2121	271.2319	187.3957
10	2	14.9055	16.8821	14.9176	14.9055
	3	43.1118	43.9705	43.6723	43.1118
	4	55.8152	54.1235	28.1223	55.8152
	5	76.4948	90.9230	76.6488	76.4948
20	2	1.4285	1.4285	1.4285	1.4285
	3	62.4271	62.4271	0.0002	62.4271
	4	14.1072	19.6107	14.1277	14.1072
	5	21.4869	25.0321	31.9328	21.4869
	10	160.7272	174.5313	162.9505	160.7272
50	2	5.1182	5.1182	0.0000	5.1182
	3	4.1522	4.1522	4.4662	4.1522
	4	15.4142	15.4142	11.0562	15.4142
	5	7.7872	7.5537	8.3801	7.7872
	10	57.1645	57.3209	21.5104	57.1645
100	2	6.6639	6.6639	4.1428	6.6639
	3	9.2183	9.2183	9.2083	9.2183
	4	1.1478	1.1036	1.1478	1.1478
	5	12.9634	12.9638	14.0745	12.9634
	10	25.8361	23.9318	17.7104	25.8361
200	2	1.9070	1.9071	1.9070	1.9070
	3	8.7954	8.7954	8.4970	8.7954
	4	25.8921	25.8921	25.7982	25.8921
	5	7.3899	7.3899	5.8877	7.3899
	10	19.2469	19.2469	19.8302	19.2469

TableA.2: % Deviations (maximum of 10 replicates) from exact solution or best solution found of problems with 6-corner polygon approximation to the elliptical regions

Number of regions	Number of facilities	Algorithms		
		SOCP-H	MP-H	SBH
5	2	103.6005	103.6005	103.6005
	3	66.1856	93.6481	66.3150
	4	173.4457	247.3590	181.7068
10	2	0.0001	0.1470	0.0002
	3	95.4082	103.4592	95.4615
	4	96.6539	189.4190	172.3686
	5	279.1088	311.4625	107.1494
20	2	37.5890	37.2492	0.0040
	3	34.2696	34.8641	18.6883
	4	19.3883	15.4420	41.0285
	5	68.0275	71.4841	58.5379
	10	76.1883	77.6167	112.4574
50	2	0.6296	0.6296	0.6296
	3	15.1163	15.1163	15.1163
	4	32.5593	32.5592	28.0360
	5	16.1184	16.1184	14.7180
	10	71.0683	72.7567	48.2196
100	2	10.0356	13.2521	10.7456
	3	2.1933	1.8213	1.8213
	4	27.5390	27.5392	0.2720
	5	14.0557	14.0557	14.0557
	10	43.8794	38.0925	38.8060
200	2	7.6376	7.6376	7.6177
	3	3.4085	3.3723	10.0432
	4	35.1503	35.0341	0.0286
	5	7.1560	7.1560	10.8945
	10	19.5884	19.5896	17.5436

TableA.3: % Deviations (maximum of 10 replicates) from exact solution or best solution found of problems with 9-corner polygon approximation to the elliptical regions

Number of regions	Number of facilities	Algorithms		
		SOCP-H	MP-H	SBH
5	2	107.4852	107.4852	107.4852
	3	70.0394	51.3379	70.1692
	4	182.4720	258.7769	192.1435
10	2	0.0001	0.0056	0.0003
	3	100.1274	107.5267	100.1622
	4	98.3229	193.4054	186.8618
	5	291.0089	324.2330	109.0892
20	2	39.7773	39.7773	0.0049
	3	34.4687	34.9746	34.9387
	4	18.5615	15.1146	40.5725
	5	69.1995	72.3813	59.0509
	10	84.2558	78.6959	98.6320
50	2	0.6904	0.6904	0.4215
	3	15.1435	15.1434	15.1434
	4	32.6612	32.6613	27.8564
	5	14.6921	14.6920	14.3605
	10	67.8153	46.5793	57.9606
100	2	10.0757	13.3216	10.7587
	3	2.8926	2.1041	1.7332
	4	26.2545	26.2545	22.6422
	5	14.0531	14.0531	14.0531
	10	44.8010	38.7331	38.2396
200	2	7.6955	7.6955	7.6774
	3	9.8726	9.8726	10.0160
	4	34.9484	34.9484	0.0278
	5	44.8079	44.8079	10.8572
	10	19.6198	19.6209	17.3634

APPENDIX B

COMPARISON OF HEURISTICS ACCORDING TO AVERAGE OBJECTIVE VALUE OF 10 REPLICATES

TableB.1: % Deviations (average of 10 replicates) from exact solution or best solution found of problems with rectangular regions

Number of regions	Number of facilities	Algorithms			
		SOCP-H	MP-H	SBH	LBH
5	2	10.6299	11.3151	10.0996	10.6299
	3	18.6119	23.1462	15.5353	18.6118
	4	107.6711	131.6528	97.0185	107.6711
10	2	3.1442	3.3611	2.3234	3.1441
	3	23.1917	20.5098	22.3640	23.1917
	4	19.5286	24.1379	13.1696	19.5286
	5	20.3694	31.8935	14.4811	20.3694
20	2	0.7143	0.7143	0.2857	0.7143
	3	6.2427	6.2427	0.0000	6.2427
	4	5.5056	7.0792	5.9876	5.5056
	5	7.8300	12.2202	10.0944	7.8300
	10	69.4544	69.1836	59.4109	69.4544
50	2	1.5355	1.5355	0.0000	1.5355
	3	0.9625	0.9650	1.3582	0.9625
	4	3.9003	3.6043	2.3547	3.9003
	5	4.1286	4.3084	3.4787	4.1286
	10	21.8085	22.1323	14.1819	21.8085
100	2	1.5266	1.5266	1.2699	1.5266
	3	3.7995	3.7878	3.5957	3.7995
	4	0.4993	0.2540	0.7346	0.4993
	5	4.8220	4.8471	4.1325	4.8220
	10	9.9975	8.6155	6.6652	9.9974
200	2	1.1288	1.1288	1.3165	1.1288
	3	3.4381	3.4381	2.1791	3.4381
	4	2.8220	2.8220	2.6893	2.8219
	5	3.6484	4.1288	2.3007	3.6484
	10	6.3790	7.1295	4.0463	6.3790

TableB.2: % Deviations (average of 10 replicates) from exact solution or best solution found of problems with 6-corner polygon approximation to the elliptical regions

Number of regions	Number of facilities	Algorithms		
		SOCP-H	MP-H	SBH
5	2	10.3601	10.3601	10.3601
	3	15.5659	29.4278	11.4309
	4	104.0674	168.8236	91.5285
10	2	0.0001	0.0777	0.0001
	3	32.0229	28.0862	35.9808
	4	38.4470	40.9886	52.3262
	5	62.2898	66.9609	23.5373
20	2	4.0570	3.7618	0.0017
	3	14.4916	13.5771	8.6522
	4	7.9487	6.7654	11.1112
	5	17.2374	10.6063	10.2688
	10	38.8274	38.7974	40.6589
50	2	0.2683	0.2487	0.1593
	3	7.4800	7.4801	6.3932
	4	6.5419	6.4721	5.5916
	5	7.3514	7.3171	5.6111
	10	26.4609	19.8707	18.0442
100	2	4.9900	6.3152	6.0250
	3	1.0239	0.9822	0.8458
	4	5.4911	5.4911	0.1133
	5	2.8916	2.8618	3.3128
	10	23.1711	22.9700	16.9937
200	2	2.1303	2.1303	2.1283
	3	2.0582	2.0365	3.6959
	4	3.5277	3.5160	0.0145
	5	2.3183	2.3205	3.5436
	10	5.8073	5.8590	5.0783

TableB.3: % Deviations (average of 10 replicates) from exact solution or best solution found of problems with 9-corner polygon approximation to the elliptical regions

Number of regions	Number of facilities	Algorithms		
		SOCP-H	MP-H	SBH
5	2	10.7485	11.1969	10.7485
	3	11.9024	17.4230	11.9746
	4	109.4832	171.3168	96.9615
10	2	0.0001	0.0056	0.0001
	3	26.6959	28.0835	36.7341
	4	34.9360	41.8961	53.2324
	5	64.3898	61.8817	23.1339
20	2	7.7599	4.0546	0.0012
	3	14.9550	13.4293	10.1992
	4	5.8837	5.4120	6.3162
	5	17.2218	16.8385	10.4066
	10	40.7942	40.3712	42.6728
50	2	0.2680	0.2303	0.0617
	3	7.5974	7.5358	6.7862
	4	6.3644	6.2850	5.5056
	5	6.2485	5.9154	4.0309
	10	25.1522	15.3561	20.3851
100	2	5.0021	6.3342	6.0428
	3	1.0415	0.9591	0.7890
	4	5.0149	4.9979	2.3918
	5	2.8074	2.8074	3.2970
	10	26.0777	22.5742	17.8226
200	2	2.1636	2.1636	2.1618
	3	3.2533	3.2533	4.0045
	4	3.5419	3.5205	0.0131
	5	6.6136	6.6172	3.7271
	10	5.8488	5.8623	6.9089

APPENDIX C

COMPARISON OF HEURISTICS ACCORDING TO MINIMUM SOLUTION TIME OF 10 REPLICATES

TableC.1: Minimum solution time of a replicate in seconds for problems with rectangular regions

Number of regions	Number of facilities	Algorithms			
		SOCP-H	MP-H	SBH	LBH
5	2	2.1900	0.0010	0.1982	0.0021
	3	2.4063	0.0167	0.3597	0.0025
	4	2.7239	0.0198	0.5235	0.0030
10	2	3.3501	0.0027	0.2704	0.0037
	3	3.7390	0.0034	0.6614	0.0043
	4	4.3591	0.0381	1.0811	0.0049
	5	4.8379	0.0439	1.1076	0.0055
20	2	5.0506	0.0051	0.4407	0.0068
	3	6.4104	0.0066	1.2639	0.0079
	4	6.9614	0.0103	1.9043	0.0090
	5	10.8365	0.0120	2.8125	0.0140
	10	9.8697	0.1435	11.8701	0.0265
50	2	17.4241	0.0166	2.1437	0.0233
	3	18.5566	0.0258	3.3706	0.0275
	4	26.3595	0.0310	4.6765	0.0381
	5	28.7008	0.0363	6.2083	0.0419
	10	55.1447	0.0985	17.3193	0.0885
100	2	46.4378	0.0409	3.8221	0.0595
	3	35.6892	0.0409	3.5378	0.0519
	4	60.3892	0.0738	9.8534	0.1082
	5	63.1142	0.1143	14.9359	0.0999
	10	92.0537	0.1947	35.2008	0.1721
200	2	107.2309	0.1161	9.4887	0.1743
	3	116.5833	0.1232	11.2848	0.1643
	4	111.4997	0.1304	20.5438	0.1492
	5	152.7129	0.1996	22.8185	0.2360
	10	163.0826	0.3872	63.1264	0.3396

TableC.2: Minimum solution time of a replicate in seconds for problems with 6-corner polygon approximation to the elliptical regions

Number of regions	Number of facilities	Algorithms		
		SOCP-H	MP-H	SBH
5	2	3.0958	0.0018	0.3052
	3	3.4800	0.0196	0.3556
	4	3.5866	0.0233	0.7620
10	2	5.2791	0.0300	0.4412
	3	5.1370	0.0377	0.9900
	4	5.3983	0.0450	1.1275
	5	5.6383	0.0528	3.1623
20	2	8.2924	0.0572	1.0048
	3	9.8448	0.0101	1.8743
	4	15.1697	0.0122	2.1573
	5	15.7287	0.0179	2.6490
	10	18.8514	0.1786	3.4236
50	2	27.3871	0.0190	2.5100
	3	40.6645	0.0304	4.5554
	4	31.3067	0.0295	5.7920
	5	45.2195	0.0438	7.0343
	10	63.2238	0.4398	18.5844
100	2	40.6566	0.0381	4.6392
	3	84.9823	0.0605	8.2795
	4	95.5718	0.1024	10.2624
	5	100.8260	0.1032	15.6566
	10	118.2207	0.2092	48.0177
200	2	164.2698	0.1456	10.1959
	3	190.7699	0.1956	16.2268
	4	200.2979	0.2657	26.4065
	5	203.5575	0.2408	31.5257
	10	195.5710	1.2542	91.9522

TableC.3: Minimum solution time of a replicate in seconds for problems with 9-corner polygon approximation to the elliptical regions

Number of regions	Number of facilities	Algorithms		
		SOCP-H	MP-H	SBH
5	2	3.2992	0.0020	0.4580
	3	3.6395	0.0240	0.5766
	4	4.0726	0.0295	0.8350
10	2	5.3652	0.0376	0.7109
	3	5.7568	0.0084	0.8047
	4	6.3146	0.0579	1.5020
	5	6.7351	0.0679	3.6520
20	2	10.9918	0.0166	1.2501
	3	10.8784	0.0230	2.2230
	4	17.1707	0.0198	3.5554
	5	18.4705	0.0279	2.7630
	10	22.0612	0.2349	9.0025
50	2	39.4701	0.0306	3.2188
	3	52.5620	0.0395	4.9897
	4	38.8000	0.0667	7.4276
	5	53.2440	0.0679	11.9471
	10	60.1930	0.5779	30.5249
100	2	59.5184	0.0503	5.5138
	3	96.3767	0.0769	11.3508
	4	128.8485	0.1129	15.3588
	5	134.1511	0.1347	19.9141
	10	142.2979	0.4063	57.5311
200	2	243.9203	0.1703	15.4599
	3	199.8027	0.1981	22.9957
	4	242.6498	0.3060	37.8585
	5	211.5202	0.3176	41.0709
	10	305.0651	1.0383	129.5877

APPENDIX D

COMPARISON OF HEURISTICS ACCORDING TO MAXIMUM SOLUTION TIME OF 10 REPLICATES

TableD.1: Maximum solution time of a replicate in seconds for problems with rectangular regions

Number of regions	Number of facilities	Algorithms			
		SOCP-H	MP-H	SBH	LBH
5	2	2.7938	0.0150	1.1986	0.0386
	3	3.9771	0.0201	0.7905	0.0036
	4	2.9301	0.0226	1.5893	0.0112
10	2	5.2103	0.0264	0.7289	0.0052
	3	6.0959	0.0339	1.4031	0.0071
	4	8.1426	0.0401	2.5120	0.0103
	5	10.1517	0.0481	3.4220	0.0102
20	2	16.3735	0.0120	1.1712	0.0184
	3	22.5488	0.0171	4.1337	0.0240
	4	20.6942	0.0745	7.0248	0.0235
	5	19.7544	0.0878	5.9371	0.0243
	10	25.3397	0.1478	19.0671	0.0419
50	2	35.4963	0.0290	4.1707	0.0452
	3	56.5259	0.1499	8.9488	0.0758
	4	66.8571	0.1810	12.3750	0.0891
	5	71.4389	0.2102	14.7295	0.0969
	10	111.5123	0.3599	40.1190	0.1684
100	2	95.6315	0.0737	7.8707	0.1134
	3	188.4771	0.3248	13.2928	0.2460
	4	182.0872	0.3546	19.3863	0.2560
	5	188.9044	0.4156	25.0377	0.2788
	10	216.6410	0.7118	59.2798	0.3765
200	2	234.0755	0.4822	17.2028	0.3637
	3	217.0627	0.2048	26.5647	0.3141
	4	309.9160	0.9057	32.9454	0.4128
	5	515.9972	0.8281	56.3843	0.7819
	10	570.2798	1.4121	110.5689	1.1459

TableD.2: Maximum solution time of a replicate in seconds for problems with 6-corner polygon approximation to the elliptical regions

Number of regions	Number of facilities	Algorithms		
		SOCP-H	MP-H	SBH
5	2	5.2786	0.0030	0.6119
	3	5.4039	0.0257	1.1169
	4	5.3784	0.0296	1.5314
10	2	10.2497	0.0378	0.7354
	3	10.7978	0.0412	1.6042
	4	11.1983	0.0469	3.4823
	5	9.5164	0.0545	5.2451
20	2	43.2033	0.0606	2.7130
	3	29.2132	0.0736	3.9187
	4	37.5403	0.0900	4.3646
	5	22.6634	0.1062	6.7480
	10	28.1735	0.1813	13.4835
50	2	84.5295	0.1383	5.4268
	3	100.8880	0.1766	7.1178
	4	87.2821	0.2146	13.0245
	5	113.9807	0.2815	12.9706
	10	108.4681	0.4442	41.9504
100	2	164.5484	0.1148	10.3623
	3	180.5588	0.1512	13.6911
	4	216.4282	0.4251	22.5015
	5	335.0324	0.5019	28.4392
	10	353.0692	0.9024	89.2593
200	2	359.4182	0.5661	15.9886
	3	747.0492	0.6851	31.7309
	4	730.8497	0.8598	47.1271
	5	615.3544	1.4368	54.0767
	10	1181.8227	1.7389	262.1124

TableD.3: Maximum solution time of a replicate in seconds for problems with 9-corner polygon approximation to the elliptical regions

Number of regions	Number of facilities	Algorithms		
		SOCP-H	MP-H	SBH
5	2	5.1718	1.5423	0.7321
	3	5.6819	0.0284	1.7761
	4	6.2633	0.0325	1.9666
10	2	11.1907	0.0396	1.2835
	3	15.5746	0.0489	2.1670
	4	16.6507	0.0595	6.4195
	5	10.8377	0.1049	6.5313
20	2	58.1809	0.0785	3.8822
	3	32.6180	0.0965	7.6151
	4	41.1996	0.1169	9.5253
	5	30.4256	0.1380	12.6566
	10	30.7841	0.2474	21.7914
50	2	112.3003	0.0613	5.9639
	3	128.4642	0.2290	10.0220
	4	107.6055	0.2807	21.1532
	5	120.5470	0.3634	21.9849
	10	149.4326	0.5876	69.8063
100	2	219.5036	0.1555	14.2944
	3	206.2278	0.4507	18.8488
	4	289.1266	0.2255	28.8656
	5	464.1908	0.6492	41.9447
	10	369.6611	1.2502	145.9835
200	2	525.7585	0.3404	20.5518
	3	448.5447	0.3833	50.1625
	4	1021.4236	1.1257	56.9824
	5	709.9619	2.1126	57.5489
	10	1452.1357	2.3108	332.7955

APPENDIX E

COMPUTATIONAL RESULTS FOR ELLIPTICAL REGION

TableE.1: % Deviations (minimum of 10 replicates) from exact solution or best solution found of problems with 6-corner polygon approximation to the elliptical regions

Number of regions	Number of facilities	Algorithms		
		SOCP-H	MP-H	SBH
5	2	0.0000	0.0000	0.0000
	3	0.0000	0.5008	0.0011
	4	0.0000	83.0352	0.4169
10	2	0.0001	0.0084	0.0001
	3	0.0002	0.9466	0.0543
	4	0.0173	1.0934	12.5557
	5	0.0004	7.4371	0.0563
20	2	0.0000	0.0337	0.0002
	3	0.0000	0.0520	0.0008
	4	0.0001	0.6414	0.0015
	5	0.1558	0.0000	0.1558
	10	3.1668	5.7083	0.0000
50	2	0.0000	0.0000	0.0000
	3	0.0000	0.0000	0.0000
	4	0.3537	0.3539	0.0000
	5	0.0000	1.6805	0.0000
	10	1.3670	0.0000	3.4539
100	2	0.0000	0.0000	0.0000
	3	0.6008	0.6008	0.0000
	4	0.0000	0.0000	0.0000
	5	0.5409	0.5409	0.0000
	10	14.0957	14.5206	0.0000
200	2	0.0000	0.0000	0.0000
	3	0.0248	0.0248	0.0000
	4	0.0000	0.0000	0.0000
	5	0.0000	0.0000	0.0000
	10	0.0000	0.0001	0.0877

TableE.2: The computation time (in seconds) to obtain solutions in Table E.1

Number of regions	Number of facilities	Algorithms		
		SOCP-H	MP-H	SBH
5	2	42,1014	0,0213	4,6365
	3	38,8330	0,2093	6,9317
	4	39,1180	0,2455	10,9828
10	2	70,9342	0,3128	5,9414
	3	72,0975	0,3878	13,0593
	4	73,6364	0,4600	19,7964
	5	75,8812	0,5362	41,4346
20	2	211,8245	0,5801	14,3076
	3	196,2729	0,6079	28,6233
	4	199,1729	0,8115	33,0121
	5	193,0513	0,9577	40,1077
	10	216,7832	1,7996	96,6725
50	2	442,5453	0,3865	35,1479
	3	673,3582	0,6813	58,7880
	4	591,0461	1,1165	87,6738
	5	647,4713	1,8125	103,3589
	10	783,6463	4,4221	295,1352
100	2	995,6140	0,7185	62,4837
	3	1324,5010	0,9899	108,7562
	4	1372,6193	2,9986	163,3545
	5	2276,7983	4,2672	242,0043
	10	2195,9010	7,2489	680,5739
200	2	2253,9129	4,3965	128,2012
	3	3066,0014	3,0562	220,9881
	4	3391,0462	6,9616	369,5170
	5	3815,7340	6,2978	396,1484
	10	6676,6731	16,8759	1506,3848

TableE.3: Average number of iterations in a replicate for problems with 6-corner polygon approximation to the elliptical regions

Number of regions	Number of facilities	Algorithms		
		SOCP-H	MP-H	SBH
5	2	3.4	4.6	6.6
	3	3.2	30	8.9
	4	3.1	30	14.8
10	2	3.7	30	6.1
	3	3.7	30	8.2
	4	3.6	30	11.3
	5	3.5	30	22.6
20	2	5.9	30	7.8
	3	5	25.1	10.8
	4	4.8	27.5	10.1
	5	4.6	27.6	9.6
	10	4.3	30	14.9
50	2	5.7	9.1	7.2
	3	7.8	12.2	9.2
	4	6.5	16.1	10.5
	5	6.7	21.4	10.7
	10	6.9	30	16.7
100	2	7.7	8.3	7.4
	3	7.8	8.9	8.6
	4	8	21.5	9.9
	5	12.5	25.8	12.2
	10	10.3	24.6	18.5
200	2	9.3	23.6	9.1
	3	12.1	13.5	9.7
	4	11.8	24.6	13
	5	11.9	18	11
	10	17.9	29.2	20.1

TableE.4: % Deviations (minimum of 10 replicates) from exact solution or best solution found of problems with 9-corner polygon approximation to the elliptical regions

Number of regions	Number of facilities	Algorithms		
		SOCP-H	MP-H	SBH
5	2	0.0000	0.0000	0.0000
	3	0.0000	1.0668	0.0008
	4	0.0000	88.2333	0.0724
10	2	0.0001	0.0052	0.0000
	3	0.0000	0.6502	0.0136
	4	9.9988	1.4535	11.4888
	5	0.0000	6.6124	0.0036
20	2	0.0000	0.0703	0.0001
	3	0.0001	0.0586	0.0000
	4	0.0000	0.3615	0.0003
	5	0.0000	0.1140	0.0030
	10	7.6560	18.6100	0.0000
50	2	0.0000	0.0000	0.0000
	3	0.0000	0.0003	0.1477
	4	0.2264	0.1869	0.0000
	5	0.4795	0.4795	0.0000
	10	0.0000	0.0234	0.6379
100	2	0.0000	0.0000	0.0000
	3	0.4926	0.4926	0.0000
	4	0.0000	0.0000	0.0000
	5	0.5618	0.5618	0.0000
	10	16.0220	14.8488	0.0000
200	2	0.0000	0.0000	0.0000
	3	0.0203	0.0202	0.0000
	4	0.0000	0.0000	0.0000
	5	0.0000	0.0000	0.0000
	10	0.0000	0.0000	0.0000

TableE.5: The computation time (in seconds) to obtain solutions in Table E.4

Number of regions	Number of facilities	Algorithms		
		SOCP-H	MP-H	SBH
5	2	41,1842	1,6688	5,3687
	3	43,0674	0,2509	9,9638
	4	43,2465	0,3047	12,8942
10	2	74,2237	0,3801	9,6990
	3	88,3145	0,2903	15,6506
	4	92,5951	0,5873	28,0616
	5	87,0980	0,7287	50,4624
20	2	255,8934	0,6852	22,5040
	3	215,2495	0,8710	38,3177
	4	231,4379	0,9676	55,5771
	5	226,6926	1,1462	65,9708
	10	264,3513	2,3872	130,4306
50	2	603,3091	0,3738	42,4155
	3	830,0344	1,4252	76,6245
	4	724,4100	1,7556	121,1156
	5	711,3999	2,3919	154,0786
	10	949,4446	5,8253	423,2668
100	2	1337,8680	0,9492	86,2144
	3	1461,1124	2,4391	136,8547
	4	1913,5258	1,5501	206,2945
	5	3174,3348	3,1550	343,7829
	10	2525,0386	9,0433	855,7308
200	2	3325,3011	2,2666	180,6078
	3	3231,9998	2,8495	300,4942
	4	4586,1821	6,7823	509,3648
	5	3974,3998	11,6391	498,1172
	10	9006,0162	19,2059	1973,4363

TableE.6: Average number of iterations in a replicate for problems with 9-corner polygon approximation to the elliptical regions

Number of regions	Number of facilities	Algorithms		
		SOCP-H	MP-H	SBH
5	2	3.4	22.6	6.5
	3	3.3	30	10.2
	4	3.1	30	12.4
10	2	3.7	30	8.2
	3	3.9	18.3	7.9
	4	3.8	30	11.7
	5	3.5	30	18.8
20	2	5.8	27.7	8.6
	3	4.9	27.6	10.4
	4	5	25.4	13.1
	5	4.7	25.5	12
	10	4.5	30	14.8
50	2	5.7	7.1	7.2
	3	7.4	19.1	8.9
	4	6.5	19.3	10.3
	5	6.3	21.7	11.7
	10	7.3	30	17.5
100	2	7.7	8.9	8.1
	3	7.6	16.8	8.4
	4	8.4	9.2	9.5
	5	12.7	15.1	12
	10	9.9	23.4	15.8
200	2	9.3	10.3	9.7
	3	9	9.9	10.2
	4	12.2	18.7	11.7
	5	10.5	24.5	10.4
	10	17.9	25.2	19.4