Mathematical Programming Formulations of the Planar Facility Location Problem

by

Margarita Zvereva

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2007

Author's declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The facility location problem is the task of optimally placing a given number of facilities in a certain subset of the plane. In this thesis, we present various mathematical programming formulations of the planar facility location problem, where potential facility locations are not specified. We first consider mixed-integer programming formulations of the planar facility locations problems with squared Euclidean and rectangular distance metrics to solve this problem to provable optimality. We also investigate a heuristic approach to solving the problem by extending the K-means clustering algorithm and formulating the facility location problem as a variant of a semidefinite programming problem, leading to a relaxation algorithm. We present computational results for the mixed-integer formulations, as well as compare the objective values resulting from the relaxation algorithm and the modified K-means heuristic. In addition, we briefly discuss some of the practical issues related to the facility location model under the continuous customer distribution.

Acknowledgements

Through the unexpected difficulties, my thesis advisor, Professor Shioda, was always very helpful and eager to offer advice. Her knowledge inspired and often surprised me. I could not have asked for a better mentor and I thank her for her guidance, time and patience.

I would like to thank Steven Dejak for giving me confidence, for his tireless encouragement, support and stimulating discussions, as well as Igor Gorodezky for his friendship and help.

My family, as always, played a large part in the completion of this thesis. I would like to thank my grandparents, including N.I.K, for taking pride in all my achievements, no matter how small. Most importantly, I would like to express my deep appreciation to my parents, for their unconditional love, support, involvement and good advice.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The facility location problem is the task of optimally placing a predetermined number of facilities to satisfy the demand of a given number of clients. In more detail, suppose customers are located in a region $\Omega \subseteq \mathbb{R}^2$. Let $c_j = (c_{j1}, c_{j2})$ represent location of customer j, and the variable $f_i = (f_{i1}, f_{i2})$ represent location of facility i. Let $d : \Omega \times \Omega \to \mathbb{R}$ be a metric. Suppose we are given a demand function $D : \Omega \to \mathbb{R}$ as well as the number of facilities to be opened, k. The facility location problem aims to find the placement of facilities f_i, for $i = 1, \ldots, k$ and connect them to customers in a way to minimize the associated transportation costs and satisfy the demand of each customer $i = 1, \ldots, n$. The transportation costs will be measured by the distance metric $d(\cdot, \cdot)$. Throughout our analysis we will deal with the uncapacitated facility location problem, meaning that each facility has no demand limit. In addition, we make an assumption that there are no fixed costs associated with opening a facility.

In this chapter, we introduce several formulations of the facility location problem as well as review some of the progress that has been made over the years. Considering the interdisciplinary nature of the problem, it is important to reflect in our discussions the wide spectrum of influences and recognize contributions from different areas like data mining, operations research and others.

We begin our discussion with the assumption that the number of customers in the region n is finite, which implies that the associated demand function D is a discrete function $D : \{c_1, \ldots, c_n\} \to \mathbb{R}$. The facility locations f_i will be specified to be continuous or discrete variables, resulting in different formulations of the problem.

1.1 Discrete Formulation

In the discrete version of the facility location problem the task is to choose the best subset of a given set of potential location cites which satisfies the demand of customers and minimizes associated costs. The problem was first formulated by Hakimi [6] in 1964. Soon after, ReVelle and Swain [11] constructed the following integer programming formulation of the problem:

$$\text{minimize} \sum_{i=1}^{n}\sum_{j=1}^{n} D_j d_{ij} x_{ij}$$

subject to:

$$\sum_{i=1}^{n} x_{ij} = 1,\ \sum_{i=1}^{n} x_{ii} = k,$$
$$x_{ij} \le x_{ii},\ x_{ij} \in \{0,1\}.$$

where $x_{ij} = 1$ if customer j is served by facility i, 0 otherwise, and $x_{ii} = 1$ if facility is opened at location i and 0 otherwise, and the constraints are defined for all $i, j = 1, \ldots, n$.

Even though the formulation can be seen as restrictive due to the limited location possibilities, it provides a realistic model. Companies often perform so-called solution space aggregation by considering a short list of potential facility sites due to limited resources, as mentioned by the authors in [5]. It is also worth pointing out that the resulting integer programming problem can be solved in a time efficient manner, allowing for the possibility of improving the accuracy of the model by expanding the set of potential facility locations.

1.2 The Planar Facility Location Problem

Next we consider the facility locations as continuous variables with domain Ω, while keeping the number of customers finite. The resulting problem, which will be referred to as the planar facility location problem, has a long history, which we briefly discuss below.

1.2.1 Single Facility: the Weber Problem

In this simple case we are to locate a single facility, given a finite number of clients in the region, along with associated demand values. Hence, the problem is to find the point $f^* \in \Omega$ which minimizes the sum of weighted Euclidean distances from itself to n customers at locations $c_1, \ldots, c_n$ with corresponding demand values $D_1, \ldots, D_n$. The problem can be stated as:

$$\min_{f\in\Omega} \sum_{j=1}^{n} D_j d(f, c_j)$$

Drezner,Klamroth, Schöbel, and Wesolowsky in [4] present a complete historical overview of the problem. Many credit Pierre de Fermat for its original formulation, but a variety of mathematicians including Torricelli, Cavalieri, Viviani and Roberval worked on the problem at the time so it is not known who proposed the problem first [4].

Several solutions to the planar facility location problem have been proposed. Torricelli is usually credited with a geometric solution to the unweighed problem with 3 customers, resulting in the spatial median point often called "Torricelli point" or "Fermat point". Another creative solution is due to Varignon. It came in the form of a device, called a Varignon Frame, which under certain physical assumptions, located the centre of mass of the given set of points. The simplest and most often used procedure, however, is called "Weiszfeld Algorithm" is based on calculation of partial derivatives of the objective function.

The Weber Problem has been well studied for a variety of different metrics and numerous approximations and iterative methods have been proposed over the years. We now consider the planar facility location problem with more than one facility.

1.2.2 Multiple Facilities

A more realistic situation arises when the task is to place more than one facility. One way to accommodate for multiple facilities is to adopt Location-Allocation model, which introduces a variable describing which customer is to be served by which facility. In more detail, let D_{ij} be amount shipped from facility i to customer j. The problem of finding k facility locations $f_1, \ldots, f_k$ to service customers $c_1, \ldots, c_n$ with demand values $D_1, \ldots, D_n$ can be described as:

$$\min_{f_j, D_{ij}} \quad \sum_{i=1}^{k} \sum_{j=1}^{n} D_{ij} d(f_i, c_j)$$

subject to

$$\begin{aligned} &\textstyle\sum_{i=1}^{k} D_{ij} = D_j \quad \text{for } i = 1, \ldots, k, \\ &D_{ij} \geq 0, \quad \text{for } i = 1, \ldots, k, \; j = 1, \ldots, n. \end{aligned}$$

The above problem is NP-hard. It is particularly difficult to solve as the objective function is neither convex nor concave and can have multiple local minima. Many heuristic procedures have been proposed over the years. Bozkaya, Zhang and Erkut [2] applied a Genetic Algorithm, a random search technique, to the p-median problem with promising results. Arora, Raghavan, Rao [1] discuss a quadtree-based approximation algorithm, implemented with dynamic programming. Other heuristic methods included sequential location-allocation procedure, local search methods, modifications of the objective function, clustering methods, projection methods, Tabu search and p-Median plus Weber heuristic. An extensive summary of heuristic and exact solution methods can be found in [4].

There are numerous other extensions and generalizations of the planar facility location problem, including objective function modifications due to more general transportation costs, introduction of stochastic elements, introduction of various barriers and forbidden regions and others.

We will formulate the planar facility location problem as a mixed-integer programming problem with quadratic and linear objective functions, depending on the metric used. The resulting formulations will allow us to obtain exact solutions, although practical computational considerations will impose restrictions on the size of the problems.

We will now briefly discuss contributions to the planar facility location problem from the area of data mining and semidefinite programming.

1.2.3 Connection to Data Mining

Data mining is a tool for organizing, classifying and extracting patterns from large sets of data. Clustering, or grouping data based on similarity, is one of the most important parts of data mining, with numerous applications in a variety of disciplines [7].

The planar facility location problem can be considered a clustering problem, adopting the minimization of the sum of squared distances from each data point to its cluster centre as the clustering criterion. The task of locating k facilities then becomes the task of grouping customers into k clusters. In other words, given n customers in the plane $c_1, \ldots, c_n$ the goal of the clustering algorithm is to divide these data points into k groups, assigning each data entry to exactly one cluster. After the clusters have been formed, it is a simple task to find the cluster centre, and hence establish the locations of the facilities.

One of the most popular and widely used clustering algorithms is the K-means algorithm, an iterative descent method for grouping data points in the plane. The authors in [10] provide a concise summary of the algorithm. Since the K-means algorithm attributes equal weight to every data point it is not immediately applicable to the planar facility location problem. We will extend the well-known K-means clustering heuristic to take into account the demand associated with each customer.

We begin by considering n points in the plane $\{c_j \mid j = 1, \ldots, n\}$ and corresponding weights $\{D_j \mid D_j \in \mathbb{R}_+,\ j = 1, \ldots, n\}$. Our goal is to divide the points into k clusters so as to minimize the sum of weighted squared Euclidian distances from each point to the corresponding cluster centre. Each data point is assigned to one of k clusters, giving rise to the cluster assignment function

$$C : \{1, \ldots, n\} \to \{1, \ldots, k\}.$$

Let μ_i be the centre of ith cluster, our goal is to find C^* and $\{\mu_i\}_{i=1}^k$ that minimize

$$W(C, \{\mu_i\}_{i=1}^k) = \sum_{i=1}^{k} \sum_{C(j)=i} D_j \parallel c_j - \mu_i \parallel^2 \tag{1.2.1}$$

Weighted K-means algorithm is an iterative procedure used to find the optimal assignment function and cluster centres, which works by minimizing (1.2.1) alternately with respect to C and $\{\mu_i\}_{i=1}^k$ until a convergence criterion is met. The iterative method is based on the following observations.

First we notice that for a fixed assignment function $\hat{C}$ we can easily find the set of cluster centres $\{\mu_i\}_{i=1}^k$ minimizing W. In more detail, if data points indexed by set S belong to some cluster i the task of minimizing W with respect to μ_i is equivalent to solving

$$\min_{\mu_i} \sum_{j \in S} D_j \parallel c_j - \mu_i \parallel^2$$

which can be done using elementary calculus by setting the partial derivative $\frac{\partial W}{\partial \mu_i}$ equal to zero and solving for μ_i. In other words, we solve the equation

$$\frac{\partial W}{\partial \mu_i} = \sum_{j \in S} 2D_j(c_j - \mu_i) = 0$$

to obtain the optimal cluster centre

$$\mu_i = \frac{\sum_{j \in S} D_j c_j}{\sum_{j \in S} D_j}$$

which is simply the centre of mass of the data points in the ith cluster. Repeating the same process for the remaining clusters we can minimize W with respect to $\{\mu_i\}_{i=1}^k$.

Next, we observe that for a fixed set of cluster centres $\{\mu_i\}_{i=1}^k$ minimizing W with respect to the assignment function C simply involves associating each data point with the closest cluster centre.

These two ideas form the basis for the weighted K-means algorithm, which starts by randomly partitioning data points into k initial sets and then repeatedly performs the optimization subroutines discussed above. At each step the objective function is reduced, guaranteeing convergence in a finite number of iterations. The algorithm terminates when no further improvement in the objective function W can be achieved by reassigning data points to different clusters.

We summarize the above ideas by the following algorithm:

Weighted K-means Clustering

1. Given cluster assignment C, minimize (1.2.1) with respect to $\{\mu\}_{i=1}^k$ obtaining centres of mass of current clusters
$\mu_i = \arg\min_\mu \sum_{C(j)=i} D_j \parallel c_j - \mu \parallel^2$.

2. Given $\{\mu\}_{i=1}^k$, minimize (1.2.1) with respect to C by assigning each observation to the closest centre of mass.
$C(j) = \arg\min_{1 \le i \le k} \parallel c_j - \mu_i \parallel^2$.

3. Repeat 1. and 2. until the assignments do not change.

The algorithm often converges to a local minimum and in those cases does not result in an optimal clustering assignment. In practice, however, the heuristic has been observed to converge quickly, allowing for the possibility to perform multiple runs with different initial cluster assignments, and choose the best solution.

1.2.4 Connection to Semidefinite Programming

Recall that Semidefinite Programming problem refers to an optimization problems of the following form:

$$\begin{aligned} &\text{minimize } \operatorname{Tr}(WZ) \\ &\text{subject to} \\ &\qquad \operatorname{Tr}(B_i Z) = b_i \quad \text{for } i = 1, \dots, m \\ &\qquad Z \succeq 0 \end{aligned}$$

where variable Z and data matrices B_i and W are real symmetric $n \times n$ matrices, and $b \in \mathbb{R}^m$. $\operatorname{Tr}(\cdot)$ denotes the trace of the matrix and the inequality $Z \succeq 0$ means that the matrix Z is positive semidefinite.

Peng and Wei in [10] established the equivalence between the classical K-means clustering problem and the following optimization problem:

$$\begin{aligned} &\text{minimize } \operatorname{Tr}(WW^T(I - Z)) \\ &\text{subject to} \\ &\qquad Ze = e,\ \operatorname{Tr}(Z) = k, \\ &\qquad Z \geq 0,\ Z^T = Z,\ Z^2 = Z. \end{aligned}$$

where $W \in \mathbb{R}^{n \times m}$ denotes the matrix whose jth row contains coordinates of the jth data point c_j. We will refer to the above problem as Projective Semidefinite Programming problem (PSDP), reflecting the fact that the solution of the above problem is an orthogonal projection matrix.

In chapter 3 we will show that the planar facility location problem can also be formulated as a PSDP problem as well as discuss a possible relaxation algorithm. We will then demonstrate on a small example a procedure to obtain an exact solution to the above programming problem, based on its geometrical properties.

1.3 Extending the Planar Facility Location Problem

Continuous facility location problem attempts to find the optimal placement of facility locations in some region $\Omega \in \mathbb{R}^2$, assuming that customers are spread continuously over the entire region, which results in the continuous demand function $D : \Omega \to \mathbb{R}$.

We will demonstrate on an illustrative example one way in which data aggregation can be used on some real-life data to obtain a continuous demand function reflecting population density and find an approximate solution to the problem using the discrete formulation of the facility location problem discussed in Section 1.1.

Chapter 2

Mixed-Integer Programming Formulations

In this chapter, we explore mixed-integer programming formulations for the planar facility location problem. As before, suppose there are n customers, with given locations $\{c_j \mid j = 1, \ldots, n\}$ located in some subset of $\mathbb{R}^2$. Without loss of generality, we will assume that $c_j \geq 0$, for all $j = 1, \ldots, n$. We denote the associated customer demand values by $D_1, \ldots, D_n$. We also assume, without the loss of generality, that each demand value D_j is strictly positive, for a customer with zero demand can be excluded from our analysis without any impact.

Our task is to find k facility locations, where k is a predefined positive integer, that minimizes the total transportation cost. We define the transportation cost to be the distance between a customer and its assigned facility multiplied by the total demand of the customer. The facility locations to be found will be denoted by $\{f_i \mid i = 1, \ldots, k\}$. Let $d(\cdot, \cdot)$ be our distance metric and let us also assume without loss of generality that each customer is served by exactly one facility, which is motivated by the fact that facilities have unlimited capacity. Thus, if customer j is serviced by facility i, the corresponding transportation cost is $D_j d(f_i, c_j)$.

To describe the model in more detail, we begin by letting

$$\begin{aligned} a_1 &= \min_{j=1,\ldots,n} c_{j1}, & a_2 &= \min_{j=1,\ldots,n} c_{j2} \\ b_1 &= \max_{j=1,\ldots,n} c_{j1} & b_2 &= \max_{j=1,\ldots,n} c_{j2} \end{aligned}$$

and conclude that the customers are located in the following region

$$\Omega := \{(x, y) \mid a_1 \leq x \leq b_1,\ a_2 \leq y \leq b_2\}.$$

It follows that $f_i \in \Omega$, for all values of i.

For all of our formulations, we will us the following binary assignment variable:

$$x_{ij} = \begin{cases} 1 & \text{if customer } j \text{ is seviced by facility } i \\ 0 & \text{otherwise} \end{cases}$$

The planar facility location problem aims to find f_i and x_{ij} that solves

$$\begin{array}{rc} \text{minimize} & \sum_{i=1}^{k} \sum_{j=1}^{n} D_j d(c_j, f_i) x_{ij} \\ \text{subject to :} & \\ & f_i \in \Omega, \\ & \sum_{i=1}^{k} x_{ij} = 1, \\ & x_{ij} \in \{0,1\}, \end{array} \tag{2.0.1}$$

for all values of $i = 1, \dots, k, \ j = 1, \dots, n$.

Before we take a closer look at the above model, we point out that a stronger statement can be made about the optimal facility locations in the case when the Euclidean metric is used. With the above assumption we can conclude that f_i will be located in the convex hull of the set of customer locations. Thus, the new demand Ω can be described as:

$$\begin{aligned} \Omega &= \operatorname{conv}\{c_j \mid j = 1, \dots, n\} \\ &= \left\{ \sum_{j=1}^{n} \alpha_{ij} c_j \;\middle|\; \sum_{j=1}^{n} \alpha_{ij} = 1, \ \alpha_{ij} \geq 0 \text{ for all } j = 1, \dots n \right\}. \end{aligned}$$

Although the above definition of Ω results in a tighter convex relaxation of the problem, it increases the per-node computation time, and for that reason will not be presented in the computational results at the end of the chapter. Thus, throughout our discussion we will consider Ω to be the rectangular region described in the beginning of the section.

We present models where $d(\cdot, \cdot)$ is the squared Euclidean distance in Section 2.1 and rectangular or Manhattan distance in Section 2.2.

2.1 Squared Euclidean Distance

Using the squared Euclidean metric, the objective function can be written as

$$\sum_{i=1}^{k} \sum_{j=1}^{n} D_j \|f_i - c_j\|^2 x_{ij},$$

which can be equivalently modeled as

$$\sum_{i=1}^{k} \sum_{j=1}^{n} D_j \|f_i x_{ij} - c_j x_{ij}\|^2.$$

Now to transform the objective function into a convex quadratic, we introduce the following auxiliary variables:

$$z_{ij1} := f_{i1} x_{ij}, \quad z_{ij2} := f_{i2} x_{ij}$$

along with the following constraints defined for all $i = 1, \dots, k,\ j = 1, \dots, n$:

$$0 \le z_{ij1} \le b_1 x_{ij},\ 0 \le z_{ij2} \le b_2 x_{ij},$$
$$f_{i1} - b_1(1 - x_{ij}) \le z_{ij1},\ f_{i2} - b_2(1 - x_{ij}) \le z_{ij2}.$$

It is possible to obtain a stronger formulation of the problem by adding the following constraints:

$$\begin{aligned} a_1 x_{ij} &\le z_{ij1} \le f_{i1} - a_1(1 - x_{ij}),\\ a_2 x_{ij} &\le z_{ij2} \le f_{i2} - a_2(1 - x_{ij}). \end{aligned}$$

However, the preliminary computational experiments indicate that the above constraints increase the per-node computation time, and for this reason will not be included in our further discussion.

Using the new variables we can write the objective function as

$$\begin{aligned} \sum_{i=1}^{k}\sum_{j=1}^{n} D_j \|f_i x_{ij} - c_j x_{ij}\|^2 &= \sum_{i=1}^{k}\sum_{j=1}^{n} D_j((f_{i1}x_{ij} - c_{j1}x_{ij})^2 + (f_{i2}x_{ij} - c_{j2}x_{ij})^2)\\ &= \sum_{i=1}^{k}\sum_{j=1}^{n} D_j((z_{ij1} - c_{j1}x_{ij})^2 + (z_{ij2} - c_{j2}x_{ij})^2). \end{aligned}$$

Incorporating the new constraints we obtain a mixed-integer quadratic programming formulation of the facility location problem.

Formulation 1. Given $c_1, \dots, c_n,\ D_1, \dots, D_n,\ k,$ and $d(\cdot, \cdot)$ as the squared Euclidean metric, the planar facility location problem can be formulated as:

$$\text{minimize} \sum_{i=1}^{k}\sum_{j=1}^{n} D_j((z_{ij1} - c_{j1}x_{ij})^2 + (z_{ij2} - c_{j2}x_{ij})^2)$$

subject to

$$\begin{aligned} &0 \le z_{ij1} \le b_1 x_{ij},\\ &0 \le z_{ij2} \le b_2 x_{ij},\\ &f_{i1} - b_1(1 - x_{ij}) \le z_{ij1},\\ &f_{i2} - b_2(1 - x_{ij}) \le z_{ij2},\\ &a_1 \le f_{i1} \le b_1,\\ &a_2 \le f_{i2} \le b_2,\\ &\sum_{i=1}^{k} x_{ij} = 1,\\ &x_{ij} \in \{0, 1\} \end{aligned}$$

where the above constraints are defined for $i = 1, \dots, k$ and $j = 1, \dots, n$.

Next, by reorganizing the terms in the objective function and introducing additional variables, we can reformulate the planar facility location problem (2.0.1) as a mixed-integer *linear* programming problem.

We first point out that based on the discussion in Section 1.2.3 we can conclude that given an optimal assignment $\{x_{ij}\}$, we can write the resulting optimal facility locations f_i as

$$f_i = \frac{\sum_{j=1}^n D_j c_j x_{ij}}{\sum_{j=1}^n D_j x_{ij}},$$

together with the assumption that each facility will serve at least one customer, which can be expressed as the constraint $\sum_{j=1}^n x_{ij} \geq 1$ for all values of $i = i, \ldots, k$. Then, the objective function using the squared Euclidean metric can be expanded as follows

$$\begin{aligned}
&\sum_{i=1}^k \sum_{j=1}^n D_j x_{ij} \|f_i - c_j\|^2 \qquad (2.1.1)\\
&= \sum_{i=1}^k \sum_{j=1}^n x_{ij} D_j \left\| c_j - \frac{\sum_{q=1}^n x_{iq} D_q c_q}{\sum_{q=1}^n x_{iq} D_q} \right\|^2 \\
&= \sum_{i=1}^k \sum_{j=1}^n x_{ij} D_j \left[\|c_j\|^2 - 2\,(c_j)^T \left(\frac{\sum_{q=1}^n x_{iq} D_q c_q}{\sum_{p=1}^n x_{ip} D_p} \right) + \left\| \frac{\sum_{q=1}^n x_{iq} D_q c_q}{\sum_{p=1}^n x_{ip} D_p} \right\|^2 \right] \\
&= \sum_{j=1}^n D_j \|c_j\|^2 \left(\sum_{i=1}^k x_{ij} \right) - 2 \sum_{i=1}^k \left(\sum_{j=1}^n x_{ij} D_j c_j \right)^T \left(\frac{\sum_{q=1}^n x_{iq} D_q c_q}{\sum_{p=1}^n x_{ip} D_p} \right) \\
&\qquad + \sum_{i=1}^k \left(\sum_{j=1}^n x_{ij} D_j \right) \frac{\left\| \sum_{q=1}^n x_{iq} D_q c_q \right\|^2}{\left(\sum_{p=1}^n x_{ip} D_p \right)^2} \\
&= \sum_{j=1}^n D_j \|c_j\|^2 \left(\sum_{i=1}^k x_{ij} \right) - 2 \sum_{i=1}^k \frac{\left\| \sum_{q=1}^n x_{iq} D_q c_q \right\|^2}{\left(\sum_{p=1}^n x_{ip} D_p \right)} + \sum_{i=1}^k \frac{\left\| \sum_{q=1}^n x_{iq} D_q c_q \right\|^2}{\left(\sum_{p=1}^n x_{ip} D_p \right)} \\
&= \sum_{j=1}^n D_j \|c_j\|^2 \left(\sum_{i=1}^k x_{ij} \right) - \sum_{i=1}^k \frac{\left\| \sum_{q=1}^n x_{iq} D_q c_q \right\|^2}{\left(\sum_{p=1}^n x_{ip} D_p \right)} \qquad \left(\text{since} \sum_{i=1}^k x_{ij} = 1 \right) \\
&= \sum_{j=1}^n D_j \|c_j\|^2 - \sum_{i=1}^k \frac{\left\| \sum_{q=1}^n x_{iq} D_q c_q \right\|^2}{\left(\sum_{p=1}^n x_{ip} D_p \right)}. \qquad (2.1.2)
\end{aligned}$$

Using the new expression for the objective function we note that Formulation 1 is equivalent to the following optimization problem

$$\begin{aligned}
\text{maximize} & \sum_{i=1}^{k} \frac{\|\sum_{l=1}^{n} x_{il} D_l c_l\|^2}{\sum_{l=1}^{n} x_{il} D_l} \qquad (2.1.3)\\
\text{subject to :} & \sum_{i=1}^{k} x_{ij} = 1,\\
& \sum_{j=1}^{n} x_{ij} \geq 1\\
& x_{ij} \in \{0,1\},
\end{aligned}$$

where the constraints are defined for all values of $i = 1, \ldots, k$ and $j = 1, \ldots, k$.

Next we let

$$t_i := \frac{\|\sum_{l=1}^{n} x_{il} D_l c_l\|^2}{\sum_{l=1}^{n} x_{il} D_l}$$

and consider the problem of maximizing $\sum_{i=1}^{k} t_i$ subject to

$$t_i \leq \frac{\|\sum_{l=1}^{n} x_{il} D_l c_l\|^2}{\sum_{l=1}^{n} x_{il} D_l}, \text{ for all } i = 1, \ldots, k$$

which can also be expressed as

$$\sum_{j=1}^{n} t_i x_{ij} D_j \leq \|\sum_{l=1}^{n} x_{il} D_l c_l\|^2 = \sum_{j=1}^{n} x_{ij} D_j^2 \|c_j\|^2 + 2 \sum_{j,l:j<l} D_j D_l c_j^T c_l x_{ij} x_{il}.$$

To linearize this constraint, let

$$z_{ijl} := x_{ij} x_{il} \text{ for } j < l \text{ and } u_{ij} := t_i x_{ij},$$

and observe that as a product of two binary 0-1 variables z_{ijl} should satisfy the following constraints for all $i = 1, \ldots, k$, and $j, l = 1, \ldots, n$ such that $j \neq l$:

$$z_{ijl} \leq x_{ij},\ z_{ijl} \leq x_{il},\ z_{ijl} \geq x_{ij} + x_{il} - 1,$$

and similarly the variable u_{ij} satisfies

$$u_{ij} \leq t_i,\ u_{ij} \leq M x_{ij},\ u_{ij} \geq t_i - M(1 - x_{ij})$$

for all values of $i = 1, \ldots, k,\ j = 1, \ldots, n$. Combining above observations we can formulate (2.1.3) as a mixed-integer linear programming problem.

Formulation 2. Given $c_1, \ldots, c_n,\ D_1, \ldots, D_n,\ k,$ and $d(\cdot,\cdot)$ as the squared Euclidean metric, the planar facility location problem can be formulated as:

$$\text{maximize} \sum_{i=1}^{k} t_i \tag{2.1.4}$$

$$\begin{aligned}
&\text{subject to :}\\
&\qquad \sum_{j=1}^{n} D_j u_{ij} \leq \sum_{j=1}^{n} x_{ij} D_j^2 \|c_j\|^2 + 2 \sum_{j,l:j<l} D_j D_l c_j^T c_l z_{ijl},\\
&\qquad z_{ijl} \leq x_{ij},\ z_{ijl} \leq x_{il}\\
&\qquad z_{ijl} \geq x_{ij} + x_{il} - 1,\\
&\qquad 0 \leq u_{ij} \leq t_i,\\
&\qquad u_{ij} \leq M x_{ij},\\
&\qquad u_{ij} \geq t_i - M(1 - x_{ij})\\
&\qquad z_{ijl} \geq 0\\
&\qquad \sum_{i=1}^{k} x_{ij} = 1,\ \sum_{j=1}^{n} x_{ij} \geq 1,\ x_{ij} \in \{0, 1\}
\end{aligned}$$

where M is a sufficiently large positive constant and all constraints are defined for all values of $i = 1, \ldots, k$, and $j, l = 1, \ldots, n$ such that $j \neq l$.

To find appropriate values for M, we need to find a valid upper-bound for optimal values of t_i. From (2.1.2), we see that

$$\sum_{j=1}^{n} D_j \|c_j\|^2 - \sum_{i=1}^{k} t_i = \sum_{i=1}^{k} \sum_{j=1}^{n} D_j \|f_i - c_j\|^2 x_{ij} \geq 0.$$

Thus, a simple upper-bound for t_i is $\sum_{j=1}^{n} D_j \|c_j\|^2$. This upper-bound can be tightened if we find a stronger lower-bound for the optimal objective value of Formulation 1 by solving the relaxation of that problem.

2.2 Rectangular Metric

Let $d_r : \Omega \times \Omega \to \mathbb{R}$ be a metric defined as follows: if $x_1 = (a_1, b_1),\ x_2 = (a_2, b_2)$ are two points in $\mathbb{R}^2$, then $d_r(x_1, x_2) = |a_1 - a_2| + |b_1 - b_2|$. With this so-called rectangular metric, we can write the objective function in (2.0.1) as:

$$\sum_{i=1}^{k} \sum_{j=1}^{n} D_j d_r(c_j, f_i) x_{ij} = \sum_{i=1}^{k} \sum_{j=1}^{n} D_j (|c_{j1} - f_{i1}| + |c_{j2} - f_{i1}|) x_{ij}.$$

Using the variables $t_{ij1},\ t_{ij2}$ to model the absolute values $|c_{j1} - f_{i1}|,\ |c_{j2} - f_{i1}|$, the planar facility location problem can be stated as:

$$\text{minimize } \sum_{i=1}^{k}\sum_{j=1}^{n} D_j(t_{ij1} + t_{ij2})x_{ij}$$

subject to:

$$\begin{aligned}
& t_{ij1} \geq c_{j1} - f_{i1},\ t_{ij1} \geq f_{i1} - c_{j1}, \\
& t_{ij2} \geq c_{j2} - f_{i2},\ t_{ij2} \geq f_{i2} - c_{j2}, \\
& \sum_{i=1}^{k} x_{ij} = 1, \\
& x_{ij} \in \{0, 1\},
\end{aligned}$$

where the constraints are defined for all $i = 1, \ldots, k$ and $j = 1, \ldots, n$.

Now to linearize the objective function, we introduce the following auxiliary variables:

$$z_{ij1} := t_{ij1}x_{ij}, \quad z_{ij2} := t_{ij2}x_{ij}$$

along with the constraints defined for all $i = 1, \ldots, k$ and all $j = 1, \ldots, n$:

$$\begin{aligned}
0 \leq z_{ij1} &\leq M_1 x_{ij}, \\
0 \leq z_{ij2} &\leq M_2 x_{ij}, \\
t_{ij1} - M_1(1 - x_{ij}) &\leq z_{ij1}, \\
t_{ij2} - M_2(1 - x_{ij}) &\leq z_{ij2}, \\
z_{ij1} &\leq t_{ij1}, \\
z_{ij2} &\leq t_{ij2},
\end{aligned}$$

where $M_p = \max\{c_{jp}, b_p - c_{jp}\}$ for $p = 1, 2$.

As a result we obtain a mixed-integer linear programming formulation of the Facility Location problem:

Formulation 3. Given $c_1, \ldots, c_n$, $D_1, \ldots, D_n$, k, and $d(\cdot, \cdot)$ as the rectangular metric, the planar facility location problem can be formulated as:

$$\text{minimize} \sum_{i=1}^{k} \sum_{j=1}^{n} D_j (z_{ij1} + z_{ij2}) \tag{2.2.1}$$

subject to

$$\begin{aligned}
&t_{ij1} \geq c_{j1} - f_{i1},\ t_{ij1} \geq f_{i1} - c_{j1},\\
&t_{ij2} \geq c_{j2} - f_{i1},\ t_{ij1} \geq f_{i1} - c_{j1},\\
&0 \leq z_{ij1} \leq M_1 x_{ij},\ 0 \leq z_{ij2} \leq M_2 x_{ij},\\
&t_{ij1} - M_1(1 - x_{ij}) \leq z_{ij1},\ t_{ij2} - M_2(1 - x_{ij}) \leq z_{ij2},\\
&z_{ij1} \leq t_{ij1},\ z_{ij2} \leq t_{ij2},\\
&\sum_{i=1}^{k} x_{ij} = 1,\ f_j \in \Omega,\ x_{ij} \in \{0, 1\},
\end{aligned}$$

where $M_p = \max\{c_{jp}, b_p - c_{jp}\}$ for $p = 1, 2$ and $i = 1, \ldots, k$ and $j = 1, \ldots, k$.

We will now introduce an alternative formulation, which uses the following constraints

$$t_{ijp} \geq c_{jp} x_{ij} - f_{ip} x_{ij},\ t_{ij1} \geq f_{ip} x_{ij} - c_{jp} x_{ij},$$

where all the variables are the same as before. Further, letting $v_{ijp} := f_{ip} x_{ij}$, we linearize the constraints to

$$\begin{aligned}
t_{ijp} \geq c_{jp} x_{ij} - v_{ijp},\ t_{ij1} \geq v_{ijp} - c_{jp} x_{ij},\\
v_{ijp} \leq b_p x_{ij},\ v_{ijp} \leq f_{ip},\\
v_{ijp} \geq f_{ip} - b_p(1 - x_{ij})
\end{aligned}$$

and obtain the following formulation of the planar facility location problem:

Formulation 4. Given $c_1, \ldots, c_n$, $D_1, \ldots, D_n$, k, the planar facility location problem can be formulated as:

$$\text{minimize} \sum_{i=1}^{k} \sum_{j=1}^{n} D_j (t_{ij1} + t_{ij2}) \tag{2.2.2}$$

subject to

$$\begin{aligned}
&t_{ij1} \geq c_{j1} x_{ij} - v_{ij1},\ t_{ij1} \geq v_{ij1} - c_{j1} x_{ij},\\
&t_{ij2} \geq c_{j2} x_{ij} - v_{ij1},\ t_{ij1} \geq v_{ij2} - c_{j2} x_{ij},\\
&0 \leq v_{ij1} \leq b_1 x_{ij},\ 0 \leq z_{ij2} \leq b_2 x_{ij},\\
&f_{j1} - b_1(1 - x_{ij}) \leq v_{ij1},\ f_{j2} - b_2(1 - x_{ij}) \leq v_{ij2},\\
&v_{ij1} \leq f_{j1},\ v_{ij2} \leq f_{j2},\\
&\sum_{i=1}^{k} x_{ij} = 1,\ f_j \in \Omega,\ x_{ij} \in \{0, 1\}
\end{aligned}$$

with the constraints defined as usual for all $i = 1, \ldots, k$ and $j = 1, \ldots, n$.

2.3 Symmetry Breaking Constraints

All of the above mixed-integer formulations are highly symmetric, in that we can renumber the facility labels without changing the problem. Such problems lead to unnecessarily many subproblems in the branch-and-bound setting [12, 8]. In this section, we propose some symmetry breaking constraints that impose some artificial ordering of the facilities.

In our first proposal, we arbitrarily assigned customer 1 to facility 1, i.e., we include the constraint

$$x_{11} = 1. \tag{2.3.1}$$

Unfortunately, the labeling for facilities $2, \ldots, k$ can still be rearranged. One possibility of imposing an ordering for the facilities is to order them according to their first coordinates:

$$f_{11} \leq f_{21} \leq \cdots \leq f_{k1}. \tag{2.3.2}$$

Another ordering possibility is by the total number of customers assigned to each facility:

$$\sum_{j=1}^{n} x_{1j} \leq \sum_{j=1}^{n} x_{2j} \leq \cdots \leq \sum_{j=1}^{n} x_{kj}. \tag{2.3.3}$$

These are clearly just a few of the possible symmetry breaking constraints, but from preliminary computation, they are effective in decreasing the total number of nodes explored. Also, it appears that (2.3.1) and (2.3.2) are more effective than (2.3.3). We illustrate the effectiveness of the symmetry breaking constraints (2.3.1) and (2.3.2) in Section 2.4.

2.4 Computational Experiments

We illustrate the computational properties of Formulations 1 – 4 using CPLEX 10.1, run with default parameter values except for a two hour time limit. All computations are made on an SGI Altix 3800 with 64 Intel Itanium-2 processors each running at 1.3GHz and 122GB of RAM running SuSE Linux with SGI ProPack 4. No parallelization is used. We generated customer location and demand data from a uniform distribution.

Tables 2.1 through 2.4 present the computational results. In these tables, "n" is the number of customers, "k" is the number of facilities, "symmetry" is the type of symmetry breaking constraint from Section 2.3 added, "time" is the CPU seconds required to solve to optimality up to 2 hours, "node" is the number of nodes explored, "bestnode" is the incumbent node, and "gap" is the final optimality gap in percentages.

Formulation 2 is significantly slower than Formulation 1 in solving the squared Euclidean metric. The relaxation of Formulation 2 seems much weaker than that of Formulation 1, perhaps due to the so called "big-M" constant and the larger number of variables. The larger problem size also leads to longer per-node computation time. Thus, the mixed-integer quadratic programming formulation of this problem appears to perform significantly better than its mixed-integer linear programming reformulation.

Tables 2.3 and 2.4 also show that Formulation 1 solves faster than Formulation 3 and 4 for the same problem instance. This was initially surprising, since the latter two formulations have linear objective functions with only slightly more variables than Formulation 1. It again appears that the continuous relaxation of Formulation 1 is stronger. However, surprisingly, the per-node computation time also seems faster for Formulation 1 than the last two formulations. There are no significant computational differences between Formulation 3 and 4.

All of the formulations could only solve small problem instances to provable optimality within the time limit. The key cause seems to be the weakness of the continuous relaxation. For example, for Formulation 1, 3, and 4, the continuous relaxations gave a trivial lower-bound of zero at the root node for all instances. The symmetry breaking constraints seem to help in general, but neither one dominated the other for any of the models. In an attempt to speed up computation time of the first two formulations, a feasible solution was obtained using weighted K-means algorithm and provided as a starting point for CPLEX. However, the preliminary computational results suggest that providing a feasible solution produced by the algorithm did not improve computation time. Moreover, CPLEX was able to find good feasible solutions quickly, thus strengthening the formulations and/or introducing valid inequalities would be the apparent next step in speeding up the solution time.

n	k	symmetry	time	nodes	bestnode	gap
16	2	none	0.39	275	128	0.00
		(2.3.1)	0.19	121	0	0.00
		(2.3.2)	0.28	206	0	0.00
16	3	none	2.65	2314	129	0.00
		(2.3.1)	1.18	1007	234	0.00
		(2.3.2)	3.12	2835	739	0.00
25	2	none	5.43	5151	4431	0.00
		(2.3.1)	2.38	2288	10	0.00
		(2.3.2)	3.38	2944	0	0.00
25	3	none	104.72	75761	61728	0.01
		(2.3.1)	45.45	38466	37238	0.01
		(2.3.2)	27.90	19306	18815	0.00
36	2	none	63.39	46490	450	0.01
		(2.3.1)	34.56	27597	14878	0.00
		(2.3.2)	38.09	27088	190	0.00
36	3	none	962.17	523298	11900	0.01
		(2.3.1)	432.97	253270	205306	0.01
		(2.3.2)	326.22	177278	64586	0.01
49	2	none	208.46	105327	180	0.01
		(2.3.1)	95.46	48558	0	0.01
		(2.3.2)	138.56	68639	0	0.01
49	3	none	7200	2528991	1637260	15.61
		(2.3.1)	3781.10	1510325	1502770	0.01
		(2.3.2)	5686.92	2138032	2126017	0.01
64	2	none	4908.32	1993068	50	0.01
		(2.3.1)	4233.89	1732198	1483115	0.01
		(2.3.2)	3095.94	1226247	94	0.01
64	3	none	7200	1680601	1631500	37.36
		(2.3.1)	7200	1782901	1731100	39.29
		(2.3.2)	7200	1963501	1914400	27.85
81	2	none	7200	2062401	130	15.64
		(2.3.1)	7200	2108282	0	9.65
		(2.3.2)	7200	1974601	0	13.17
81	3	none	7200	1138501	1107500	45.34
		(2.3.1)	7200	1244201	1208100	48.05
		(2.3.2)	7200	1206418	1166900	47.52
100	2	none	7200	1656501	1623600	40.60
		(2.3.1)	7200	1700701	1669100	39.12
		(2.3.2)	7200	1591401	1558400	42.28
100	3	none	7200	935357	905300	66.97
		(2.3.1)	7200	974701	949700	68.64
		(2.3.2)	7200	1069901	1043400	58.16

Table 2.1: Form. 1. “n”:num. customers,“k” :num. facilities,“symmetry”:symmetry breaking constraint, “time”:CPU sec. to solve to optimality up to 2 hrs, “node”:num. of nodes explored, “bestnode”:incumbent node, “gap”:final optimality gap (%).

n	k	symmetry	time	nodes	bestnode	gap
16	2	none	9.22	4164	2487	0.00
		(2.3.1)	4.76	2335	2295	0.00
16	3	none	1185.42	461905	459064	0.01
		(2.3.1)	1139.33	553872	544825	0.01
25	2	none	7200	1749601	1663200	10.83
		(2.3.1)	7200	2122468	1752445	4.49
25	3	none	7200	510937	492000	135.35
		(2.3.1)	7200	649809	606300	132.68
36	2	none	7200	779095	739200	60.48
		(2.3.1)	7200	738801	702000	47.55
36	3	none	7200	197501	186400	171.50
		(2.3.1)	7200	202242	189000	168.60
49	2	none	7200	211001	201100	78.19
		(2.3.1)	7200	255201	243200	72.82
49	3	none	7200	71354	67400	188.64
		(2.3.1)	7200	94759	91400	196.47
64	2	none	7200	83053	81200	94.29
		(2.3.1)	7200	78429	76100	93.88
64	3	none	7200	20239	20200	200.85
		(2.3.1)	7200	24065	24000	215.26
81	2	none	7200	26856	26500	97.09
		(2.3.1)	7200	30427	30100	99.82
81	3	none	7200	5067	5000	237.08
		(2.3.1)	7200	7399	7300	235.00
100	2	none	7200	10089	10000	118.20
		(2.3.1)	7200	12357	12200	106.12
100	3	none	7200	969	900	256.77
		(2.3.1)	7200	943	900	252.07

Table 2.2: Formulation 2. “n”:number of customers, “k”:number facilities, “symmetry”:symmetry breaking constraint added, “time” :CPU seconds required to solve to optimality up to 2 hours, “node”:number of nodes explored, “bestnode” : incumbent node, “gap” : final optimality gap (%).

n	k	symmetry	time	nodes	bestnode	gap
16	2	none	0.50	335	0	0.00
		(2.3.1)	0.30	168	157	0.00
		(2.3.2)	0.38	252	0	0.00
16	3	none	4.59	4250	300	0.00
		(2.3.1)	1.42	749	280	0.00
		(2.3.2)	4.00	4396	280	0.00
25	2	none	83.16	52278	46880	0.01
		(2.3.1)	31.32	26933	26610	0.01
		(2.3.2)	42.90	37474	27950	0.01
25	3	none	7200	2959301	4230	15.42
		(2.3.1)	638.63	297289	119019	0.01
		(2.3.2)	3032.50	2430585	2392010	0.01
36	2	none	2835.42	1300984	261021	0.01
		(2.3.1)	1443.08	654736	515139	0.01
		(2.3.2)	2095.55	1089697	242226	0.01
36	3	none	7200	2366401	2352900	35.51
		(2.3.1)	7200	1939202	1878500	20.47
		(2.3.2)	7200	2222909	1150610	31.06
49	2	none	7200	1251501	11565	15.11
		x11	7200	2020704	480	3.43
		(2.3.2)	7200	1136949	1123600	10.16
49	3	none	7200	1593901	1582300	57.38
		(2.3.1)	7200	819101	810600	46.54
		(2.3.2)	7200	1009701	1002600	54.15
64	2	none	7200	757801	750400	33.37
		(2.3.1)	7200	778301	769200	30.43
		(2.3.2)	7200	659301	652300	33.63
64	3	none	7200	894201	889000	64.75
		(2.3.1)	7200	770501	765500	66.43
		(2.3.2)	7200	1094601	1090200	61.12
81	2	none	7200	503101	1280	49.95
		(2.3.1)	7200	492901	78820	43.02
		(2.3.2)	7200	586601	4020	42.85
81	3	none	7200	780601	774700	73.73
		(2.3.1)	7200	541013	536800	71.36
		(2.3.2)	7200	712001	706800	70.17
100	2	none	7200	368301	365700	60.06
		(2.3.1)	7200	489101	485200	56.09
		(2.3.2)	7200	461165	459800	55.58
100	3	none	7200	699401	699300	82.06
		(2.3.1)	7200	338201	336300	86.01
		(2.3.2)	7200	549501	546700	79.62

Table 2.3: Form 3. “n”:num. customers, “k”:num. facilities, “symmetry” :symmetry breaking constraint, “time”:CPU sec. to solve to optimality up to 2 hrs, “node”: num. of nodes explored,“bestnode”:incumbent node,“gap”:final optimality gap (%).

n	k	symmetry	time	nodes	bestnode	gap
16	2	none	0.57	434	0	0.00
		(2.3.1)	0.38	212	50	0.00
		(2.3.2)	0.45	348	10	0.00
16	3	none	10.55	8540	440	0.00
		(2.3.1)	2.18	1261	490	0.00
		(2.3.2)	6.39	4441	1900	0.00
25	2	none	100.54	106882	105954	0.01
		(2.3.1)	43.44	47916	40758	0.00
		(2.3.2)	64.51	67882	50049	0.01
25	3	none	7200	5463600	4947100	8.75
		(2.3.1)	1647.18	1070358	1033410	0.01
		(2.3.2)	2466.52	1881502	22940	0.01
36	2	none	5523.62	3553487	3169994	0.01
		(2.3.1)	2718.60	1688680	43498	0.01
		(2.3.2)	2633.38	2033689	41553	0.01
36	3	none	7200	2220801	2208700	34.76
		(2.3.1)	7200	1559301	3720	20.96
		(2.3.2)	7200	2321901	2310200	33.11
49	2	none	7200	2039701	10	15.02
		(2.3.1)	7200	3213850	80	4.48
		(2.3.2)	7200	2064301	20	13.92
49	3	none	7200	1134949	996930	59.20
		(2.3.1)	7200	789601	783500	48.91
		(2.3.2)	7200	1179433	1170600	56.74
64	2	none	7200	932201	925200	32.54
		(2.3.1)	7200	1078101	1070400	33.64
		(2.3.2)	7200	1097501	70	30.55
64	3	none	7200	860801	858200	67.78
		(2.3.1)	7200	731401	725800	64.17
		(2.3.2)	7200	834601	830100	64.26
81	2	none	7200	832601	100	42.41
		(2.3.1)	7200	877101	0	41.70
		(2.3.2)	7200	780001	20	40.44
81	3	none	7200	510701	505900	72.96
		(2.3.1)	7200	605201	604800	69.81
		(2.3.2)	7200	679201	674700	72.71
100	2	none	7200	890301	885600	57.75
		(2.3.1)	7200	852201	848800	58.44
		(2.3.2)	7200	885701	883300	55.41
100	3	none	7200	646901	644600	84.05
		(2.3.1)	7200	588901	586500	81.35
		(2.3.2)	7200	662401	658200	81.42

Table 2.4: Form 4.“n”:num.customers, “k”:num. facilities, “symmetry”:symmetry breaking constraint, “time”:CPU sec.to solve to optimality up to 2 hrs, “node”:number of nodes explored, “bestnode”:incumbent node, “gap”:final optimality gap (%).

Chapter 3

Facility Location as PSDP

In this chapter we will extend the work of authors in [10] and show that solving the planar facility location is equivalent to solving a projective semidefinite programming problem. In the last section we will consider an algorithm to solve a relaxation of the resulting PSDP problem.

We have mentioned before that there is an alternative way of modeling the weighted clustering problem using the assignment matrix

$$X = \{x_{ij}\} \in \{0,1\}^{k\times n}$$

where

$$x_{ij} = \begin{cases} 1, & \text{if point j is assigned to cluster i,} \\ 0, & \text{othewise.} \end{cases}$$

As was pointed out in Section 2.1, we have the following mathematical programming formulation of the planar facility location problem

$$\begin{aligned} \min_{x_{ij}} \text{imize} \quad & \sum_{i=1}^{k}\sum_{j=1}^{n} x_{ij} D_j \left\| c_j - \frac{\sum_{q=1}^{n} x_{iq} D_q c_q}{\sum_{q=1}^{n} x_{iq} D_q} \right\|^2 \\ \text{subject to} \quad & \sum_{i=1}^{k} x_{ij} = 1, \\ & \sum_{j=1}^{n} x_{ij} \geq 1, \\ & x_{ij} \in \{0,1\}, \end{aligned} \tag{3.0.1}$$

where the constraints are defined for all $i = 1, \ldots, k$ and $j = 1, \ldots, n$. Note that the constraint $\sum_{j=1}^{n} x_{ij} \geq 1$ simply states that each facility is to serve at least one customer.

Since the problem above is equivalent to the weighted K-means problem , it is possible to use the weighted K-means heuristic to obtain a feasible solution to the planar facility location problem with squared Euclidean distance.

3.1 PSDP Formulation of the Planar Facility Location Problem

In this section, we describe the process of formulating the weighted K-means clustering problem as PSDP. Recall that projective semidefinite programming problem has the following form

$$\begin{aligned} &\text{minimize } \operatorname{Tr}(WZ) \\ &\text{subject to} \\ &\qquad \operatorname{Tr}(B_i Z) = b_i \quad \text{for } i = 1, \ldots, m \\ &\qquad Z^2 = Z,\ Z = Z^T,\ Z \succeq 0. \end{aligned}$$

To formulate our problem as a PSDP we begin by working on the objective function (3.0.1) to express it as a trace of a matrix, and then we concentrate on the constraints. Recall from Section 2.1 that the objective function can be written as

$$\sum_{i=1}^{k}\sum_{j=1}^{n} x_{ij} D_j \left\| c_j - \frac{\sum_{q=1}^{n} x_{iq} D_q c_q}{\sum_{q=1}^{n} x_{iq} D_q} \right\|^2 = \sum_{j=1}^{n} D_j \left\| c_j \right\|^2 - \sum_{i=1}^{k} \frac{\left\| \sum_{q=1}^{n} x_{iq} D_q c_q \right\|^2}{\left(\sum_{p=1}^{n} x_{ip} D_p \right)}. \tag{3.1.1}$$

Next we need to find matrices which will allow us to write the above expression in the compact trace form. We begin by considering the first term in (3.1.1) and defining $C \in \mathbb{R}^{n\times 2}$ to be the matrix whose jth row represents coordinates of the jth customer c_j and $D = \operatorname{diag}(D_1, \ldots, D_n) \in \mathbb{R}^{n\times n}$ to be the matrix containing customer demand values in its diagonal. It follows that the first term in (3.1.1) can be written as

$$\sum_{j=1}^{n} D_j \left\| c_j \right\|^2 = \operatorname{Tr}\left((D^{\frac{1}{2}} C)(D^{\frac{1}{2}} C)^T \right). \tag{3.1.2}$$

Further, to express the second term of the objective function in the desired form we use the assignment matrix $X = \{x_{ij}\} \in \{0,1\}^{k\times n}$ to let $Y := XD^{\frac{1}{2}}$ and to define

$$Z := Y^T \left(YY^T \right)^{-1} Y \in \mathbb{R}^{n\times n}. \tag{3.1.3}$$

Using above definition of Z we can write the second term in (3.1.1) as

$$\sum_{i=1}^{k} \frac{\left\| \sum_{q=1}^{n} x_{iq} D_q c_q \right\|^2}{\left(\sum_{q=1}^{n} x_{iq} D_q \right)} = \operatorname{Tr}\left((D^{\frac{1}{2}} C)(D^{\frac{1}{2}} C)^T Z \right). \tag{3.1.4}$$

Indeed, we observe that

$$\begin{aligned} \{YY^T\}_{ij} &= \sum_{l=1}^{n} x_{il} D_l^{\frac{1}{2}} x_{jl} D_l^{\frac{1}{2}} \\ &= \begin{cases} \sum_{l=1}^{n} x_{il} D_l & \text{if } i = j \\ 0 & \text{othewise, since } \ x_{il} x_{jl} = 0 \text{ for } i \neq j, \end{cases} \end{aligned}$$

to conclude that

$$YY^T = \operatorname{diag}\left(\sum_{l=1}^{n} x_{1l}D_l, \ldots, \sum_{l=1}^{n} x_{kl}D_l\right),$$

which in turn implies (3.1.4). We point out that the constraint $\sum_{j=1}^{n} x_{ij} \geq 1$ together with the assumption that each demand value is positive guarantees that the matrix YY^T is invertible.

Combining (3.1.2) and (3.1.4) we conclude that the objective function (3.1.1) can be written as

$$\operatorname{Tr}\left((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T(I-Z)\right). \tag{3.1.5}$$

Now we shift the focus to the constraints in (3.0.1). The first constraint, which states that each customer can be assigned to only one facility, can be written in matrix form as follows

$$e^kX = e^n, \quad \text{where } e^r = (1, \ldots, 1) \in \mathbb{R}^r.$$

In order to capture the effect of the above constraint on Z we observe that

$$e^nD^{\frac{1}{2}}Z = e^kXD^{\frac{1}{2}}Z = e^kYY^T\left(YY^T\right)^{-1}Y = e^kY = e^kXD^{\frac{1}{2}} = e^nD^{\frac{1}{2}}. \tag{3.1.6}$$

In addition, we note that matrix Z is symmetric, as can be seen from:

$$Z^2 = Y^T\left(YY^T\right)^{-1}YY^T\left(YY^T\right)^{-1}Y = Y^T\left(YY^T\right)^{-1}Y = Z.$$

Now to simplify notation, we let

$$d = (e^nD^{\frac{1}{2}})^T,$$

and conclude that the restriction that each customer is to be served by one facility imposes the following constraint on Z

$$Zd = d.$$

Next, we note that since the ijth entry of Z is of the form

$$Z_{ij} = \sum_{r=1}^{k} \frac{x_{ri}D_i^{\frac{1}{2}}x_{rj}D_j^{\frac{1}{2}}}{\sum_{m=1}^{n} x_{rm}D_m}, \tag{3.1.7}$$

the trace of Z is equal to the number of facilities, since

$$\operatorname{Tr}(Z) = \sum_{q=1}^{n} Z_{qq} = \sum_{q=1}^{n}\sum_{r=1}^{k} \frac{x_{rq}D_q}{\sum_{m=1}^{n} x_{rm}D_m} = \sum_{r=1}^{k} \frac{\sum_{q=1}^{n} x_{rq}D_q}{\sum_{m=1}^{n} x_{rm}D_m} = \sum_{r=1}^{k} 1 = k.$$

Further investigating the effect of the constraints of X on Z, from (3.1.7) we see that since each entry of X is either 0 or 1 and $\sum_{j=1}^{n} x_{ij} \geq 1$, thus each entry of Z must be nonnegative.

In summary, based on the definition of Z (3.1.3) we have translated the constraints on the assignment matrix X into statements about Z as well as deduced several properties about its trace, symmetry and nonnegativity. Now declaring Z to be the variable matrix and putting together the desired properties of Z as constraints, we consider the following Projective Semidefinite Programming problem.

$$\begin{array}{ll} \text{minimize} & \operatorname{Tr}\Big((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T(I-Z)\Big) \\ \text{subject to} & \\ & Z^2 = Z,\ Z^T = Z, \\ & \operatorname{Tr}(Z) = k,\ Zd = d \\ & Z \geq 0. \end{array} \tag{3.1.8}$$

Here $Z \geq 0$ means that Z is a matrix with nonnegative entries.

Remark 1. Recall that an orthogonal projection matrix A is a symmetric matrix satisfying $A^2 = A$. Thus, the constraints of the above optimization problem indicate that its feasible solutions are orthogonal projection matrices. Moreover, the constraint $Zd = d$ indicates that $d = e^n D^{\frac{1}{2}}$ is an eigenvector of Z corresponding to its largest eigenvalue 1.

Even though the constraints on Z in (3.1.8) were deduced from the properties of the assignment matrix X, it is not obvious that the PSDP (3.1.8) is equivalent to the planar facility location problem (3.0.1). In the next section we will show that the two problems are in fact equivalent.

3.2 Equivalence of PSDP and MIP

In this section we prove that the PSDP formulation of the planar facility location problem is in fact equivalent to the planar facility location problem. We begin by stating a well-known lemma which will be used at a later stage.

Lemma 1. *For any positive semidefinite matrix A there exists an index $k \in \{1, \dots, n\}$ such that*

$$A_{kk} = \max_{i,j} |A_{ij}|$$

The lemma simply states that the largest entry of a positive semidefinite matrix will always occur on the diagonal. We now state a standard proof of the lemma.

Proof. Suppose A is positive semidefinite, and there exist i, j such that $i \neq j$ and $|A_{ij}| > |A_{pq}|$ for all values of p, q. Consider 2×2 submatrix $\begin{bmatrix} A_{ii} & A_{ij} \\ A_{ij} & A_{jj} \end{bmatrix}$. By assumption, $A_{ii}A_{jj} - A_{ij}^2 < 0$, which implies that the above submatrix is not positive definite, contradicting the assumption that A is and implying the statement of the lemma. $\square$

Next, we will take a closer look at the set of feasible solution to (3.1.8). We will investigate the constraints in the PSDP and show that they enforce a special structure on the feasible solutions, which will be instrumental in establishing the equivalence of the two optimization problems. We will prove the following important lemma.

Lemma 2. *Let Z be a feasible solution to PSDP* (3.1.8). *By rearranging rows and columns of Z we can obtain a block diagonal matrix of the form*

$$Z_k = diag\,(Z_{\mathcal{J}_1\mathcal{J}_1} \dots Z_{\mathcal{J}_k\mathcal{J}_k})\,, \tag{3.2.1}$$

where each $Z_{\mathcal{J}_i\mathcal{J}_i}$ is a square submatrix of Z indexed by sets $\mathcal{J}_i$ for each $i = 1,\dots,k$. Moreover, each submatrix $Z_{\mathcal{J}_i\mathcal{J}_i}$ is an orthogonal projection matrix with trace 1.

Proof. Let Z be a feasible solution to PSDP (3.1.8). It follows from the constraints that $D^{-\frac{1}{2}}ZD^{-\frac{1}{2}}$ is positive semidefinite. Indeed, for any $y \in \mathbb{R}^n$ we have

$$y^TD^{-\frac{1}{2}}ZD^{-\frac{1}{2}}y = y^TD^{-\frac{1}{2}}ZZD^{-\frac{1}{2}}y = (ZD^{-\frac{1}{2}}y)^T(ZD^{-\frac{1}{2}}y) = \parallel ZD^{-\frac{1}{2}}y \parallel^2 \geq 0.$$

Using Lemma 1 we conclude that there exist an index $i_1 \in \{1,\dots,n\}$ such that

$$(D^{-\frac{1}{2}}ZD^{-\frac{1}{2}})_{i_1i_1} = \max_{i,j}(D^{-\frac{1}{2}}ZD^{-\frac{1}{2}})_{ij} > 0, \tag{3.2.2}$$

or simply

$$(D_{i_1})^{-\frac{1}{2}}Z_{i_1i_1}(D_{i_1})^{-\frac{1}{2}} \geq (D_i)^{-\frac{1}{2}}Z_{ij}(D_j)^{-\frac{1}{2}} \text{ for all } i,j \in \{1,\dots,n\}. \tag{3.2.3}$$

We next consider the following index set, locating all the positive entries in the i_1th row of Z

$$\mathcal{J}_1 = \{j|\ Z_{i_1j} > 0\}.$$

Our goal now is to find an explicit expression for the entries in the submatrix $Z_{\mathcal{J}_1\mathcal{J}_1} = (Z_{ij})_{i,j\in\mathcal{J}_1}$, which will form the first block in (3.2.1). We will show that each element of the above submatrix is positive, and all entries Z_{ij} such that $i \in \mathcal{J}_1,\ j \notin \mathcal{J}_1$ and $j \in \mathcal{J}_1,\ i \notin \mathcal{J}_1$ are equal to zero, which will guarantee the desired block structure.

We begin by noting that from $Z^2 = Z$, $Z^T = Z$ and from the definition of $\mathcal{J}_1$ we have

$$\sum_{j\in\mathcal{J}_1}(Z_{i_1j})^2 = \sum_{j=1}^{n} Z_{i_1j}Z_{ji_1} = Z_{i_1i_1},$$

which implies that

$$\sum_{j\in\mathcal{J}_1}\left(\frac{Z_{i_1j}}{Z_{i_1i_1}}\right)Z_{i_1j} = 1.$$

Above expression can also be written as

$$\sum_{j\in\mathcal{J}_1} \frac{(D_{i_1})^{\frac{1}{2}} Z_{i_1 j}}{D_j^{\frac{1}{2}} Z_{i_1 i_1}} \frac{D_j^{\frac{1}{2}}}{(D_{i_1})^{\frac{1}{2}}} Z_{i_1 j} = 1 \tag{3.2.4}$$

Furthermore, the constraints $Zd = d$ and $Z^T = Z$ imply that

$$\sum_{j\in\mathcal{J}_1} D_j^{\frac{1}{2}} Z_{i_1 j} = (D_{i_1})^{\frac{1}{2}},$$

which in turn gives

$$\sum_{j\in\mathcal{J}_1} \frac{D_j^{\frac{1}{2}}}{(D_{i_1})^{\frac{1}{2}}} Z_{i_1 j} = 1. \tag{3.2.5}$$

From (3.2.3) we deduce that

$$\frac{(D_{i_1})^{\frac{1}{2}} Z_{i_1 j}}{D_j^{\frac{1}{2}} Z_{i_1 i_1}} \leq 1 \quad \text{for all } j \in \mathcal{J}_1.$$

Since all demand values are positive and all entries in Z are nonnegative, combining above inequality with equations (3.2.4) and (3.2.5) we conclude that

$$\frac{(D_{i_1})^{\frac{1}{2}} Z_{i_1 j}}{D_j^{\frac{1}{2}} Z_{i_1 i_1}} = 1 \quad \text{for all } j \in \mathcal{J}_1,$$

which implies that

$$Z_{i_1 j} = \frac{D_j^{\frac{1}{2}}}{(D_{i_1})^{\frac{1}{2}}} Z_{i_1 i_1} \quad \text{for all } j \in \mathcal{J}_1. \tag{3.2.6}$$

Now from (3.2.5) we obtain

$$Z_{i_1 i_1} = \frac{D_{i_1}}{\sum_{p\in\mathcal{J}_1} D_p}, \tag{3.2.7}$$

and finally arrive at the following description of all nonzero elements in i_ith row of Z

$$Z_{i_1 j} = \frac{(D_{i_1})^{\frac{1}{2}} D_j^{\frac{1}{2}}}{\sum_{p\in\mathcal{J}_1} D_p} \quad \text{for all } j \in \mathcal{J}_1. \tag{3.2.8}$$

We will use the above result to obtain a similar expression for the remaining elements of the submatrix $Z_{\mathcal{J}_1\mathcal{J}_1}$, starting with the entries in the diagonal.

Consider 2×2 submatrix

$$\begin{pmatrix} Z_{i_1 i_1} & Z_{i_1 j} \\ Z_{j i_1} & Z_{jj} \end{pmatrix} \qquad \text{where } j \in \mathcal{J}_1 \text{ and } j \neq i_1.$$

Being a submatrix of a positive semidefinite matrix Z it is also positive semidefinite with a nonnegative determinant. In other words

$$Z_{i_1 i_1} Z_{jj} - Z_{i_1 j} Z_{i_1 j} \geq 0$$

which implies that

$$Z_{jj} \geq \frac{Z_{i_1 j}^2}{Z_{i_1 i_1}} = \frac{D_j}{\sum_{p \in \mathcal{J}_1} D_p}.$$

On the other hand, from (3.2.3) we know that

$$Z_{jj} \leq \frac{D_j}{D_{i_1}} Z_{i_1 i_1} = \frac{D_j}{\sum_{p \in \mathcal{J}_1} D_p}.$$

Combining the two inequalities we conclude that

$$Z_{jj} = \frac{D_j}{\sum_{p \in \mathcal{J}_1} D_p} \qquad \text{for all } j \in \mathcal{J}_1. \tag{3.2.9}$$

Finally, using the information about the diagonal elements in $Z_{\mathcal{J}_1 \mathcal{J}_1}$ and the elements in the i_1th row, we will obtain an expression for the remaining terms in the submatrix.

To find Z_{ij}, where $i, j \in \mathcal{J}_1$ such that $i, j \neq i_1$ and $i \neq j$ we will consider the following 3×3 submatrix

$$\begin{pmatrix} Z_{i_1 i_1} & Z_{i_1 i} & Z_{i_1 j} \\ Z_{i i_1} & Z_{ii} & Z_{ij} \\ Z_{j i_1} & Z_{ji} & Z_{jj} \end{pmatrix}.$$

The above matrix is positive semidefinite, since Z is, and hence has a nonnegative determinant. Therefore,

$$-Z_{i_1 i_1} Z_{ij}^2 + 2 Z_{i_1 j} Z_{i_1 i} Z_{ij} + Z_{i_1 i_1} Z_{ii} Z_{jj} - Z_{i_1 i}^2 Z_{jj} - Z_{i_1 j}^2 Z_{ii} \geq 0;$$

substituting expressions for $Z_{i_1 i}$, $Z_{i_1 j}$, Z_{ii}, Z_{jj} from (3.2.9) and (3.2.8) we obtain

$$\frac{D_{i_1}}{\sum_{p \in \mathcal{J}_1} D_p} \left(Z_{ij} - \frac{D_i^{\frac{1}{2}} D_j^{\frac{1}{2}}}{\sum_{p \in \mathcal{J}_1} D_p} \right)^2 \leq 0,$$

which implies that

$$Z_{ij} = \frac{D_i^{\frac{1}{2}} D_j^{\frac{1}{2}}}{\sum_{p \in \mathcal{J}_1} D_p}, \qquad i,\ j \in \mathcal{J}_1.$$

In summary, from the constraints in (3.1.8) we have found explicit expression for positive elements in i_1th row of Z, which we then used to establish the diagonal and later all other entries in the symmetric submatrix $Z_{\mathcal{J}_1\mathcal{J}_1}$, which can be described by

$$(Z_{\mathcal{J}_1\mathcal{J}_1})_{ij} = \frac{D_i^{\frac{1}{2}} D_j^{\frac{1}{2}}}{\sum_{p\in\mathcal{J}_1} D_p} \qquad i,j \in \mathcal{J}_1. \tag{3.2.10}$$

Recall that our goal was to use matrix $Z_{\mathcal{J}_1\mathcal{J}_1}$ as a first block in the block diagonal matrix Z_k obtained from Z by rearranging its entries. Thus, it is remains to show that

$$Z_{ij} = 0 \quad \text{for all } i \in \mathcal{J}_1,\ j \notin \mathcal{J}_1 \tag{3.2.11}$$

To arrive at the above result we first notice that for each $i \in \mathcal{J}_1$ we have

$$\sum_{j\in\mathcal{J}_1} D_j^{\frac{1}{2}} Z_{ij} = \sum_{j\in\mathcal{J}_1} D_j^{\frac{1}{2}} \frac{D_i^{\frac{1}{2}} D_j^{\frac{1}{2}}}{\sum_{p\in\mathcal{J}_1} D_p} = D_i^{\frac{1}{2}}.$$

On the other hand, the constraint $Zd = d$ indicates that for all $i \in \mathcal{J}_1$

$$\sum_{j=1}^{k} D_j^{\frac{1}{2}} Z_{ij} = D_i^{\frac{1}{2}},$$

From above equations along with the positivity of each demand value and non-negativity of Z we conclude that $Z_{ij} = 0$ for $i \in \mathcal{J}_1,\ j \notin \mathcal{J}_1$, as desired.

We will now point out some properties of the resulting submatrix $Z_{\mathcal{J}_1\mathcal{J}_1}$. First, from (3.2.10) we see that $Z_{\mathcal{J}_1\mathcal{J}_1}$ is a symmetric matrix. Second, its trace is equal to 1, indeed

$$\mathrm{Tr}(Z_{\mathcal{J}_1\mathcal{J}_1}) = \sum_{j\in\mathcal{J}_1} \frac{D_j^{\frac{1}{2}} D_j^{\frac{1}{2}}}{\sum_{p\in\mathcal{J}_1} D_p} = 1.$$

And finally by noting that

$$(Z_{\mathcal{J}_1\mathcal{J}_1})^2_{ij} = \sum_{k\in\mathcal{J}_1} (Z_{\mathcal{J}_1\mathcal{J}_1})_{ik}(Z_{\mathcal{J}_1\mathcal{J}_1})_{kj} = \sum_{j\in\mathcal{J}_1} \frac{D_i^{\frac{1}{2}} (D_{kk})^{\frac{1}{2}} (D_{kk})^{\frac{1}{2}} D_j^{\frac{1}{2}}}{\sum_{p\in\mathcal{J}_1} D_p \sum_{p\in\mathcal{J}_1} D_p} = (Z_{\mathcal{J}_1\mathcal{J}_1})_{ij},$$

we conclude that $(Z_{\mathcal{J}_1\mathcal{J}_1})^2 = Z_{\mathcal{J}_1\mathcal{J}_1}$. In other words we have shown that $Z_{\mathcal{J}_1\mathcal{J}_1}$ is an orthogonal projection matrix whose trace is equal to 1.

We now rearrange the elements in Z to obtain

$$Z_1 = \begin{pmatrix} Z_{\mathcal{J}_1\mathcal{J}_1} & 0 \\ 0 & Z_{\overline{\mathcal{J}}_1\overline{\mathcal{J}}_1} \end{pmatrix},$$

where $\overline{\mathcal{J}}_1 = \{j | j \notin \mathcal{J}_1\}$. Now setting aside $Z_{\mathcal{J}_1\mathcal{J}_1}$ and repeating this process with $Z_{\overline{\mathcal{J}}_1\overline{\mathcal{J}}_1}$, we form $Z_{\mathcal{J}_1}, \ldots, \mathcal{J}_k$ each with trace equal to one. Hence we obtain the following block diagonal matrix

$$Z_k = \mathrm{diag}\left(Z_{\mathcal{J}_1\mathcal{J}_1} \ldots Z_{\mathcal{J}_k\mathcal{J}_k}\right),$$

which completes the proof of the lemma. $\square$

Notice that in the process of proving Lemma 3.2 we have associated each nonzero entry of Z with some set $\mathcal{J}_k$. Thus, we are now able to provide a better description of the feasible solution to the PSDP (3.1.8), whose entries have the following form

$$Z_{pq} = \begin{cases} \frac{D_p^{\frac{1}{2}} D_q^{\frac{1}{2}}}{\sum_{m \in \mathcal{J}_{r^*}} D_m} & \text{if there exists } r^* \in \{1, \ldots, k\} \text{ such that } p, q \in \mathcal{J}_{r^*} \\ 0 & \text{otherwise} \end{cases}$$

Armed with the knowledge about the structure and properties of the feasible solution to the Projective Semidefinite Programming problem (3.1.8) we are ready to state the main theorem of this chapter, which will rely strongly on the previous lemma.

Theorem 1. *Given $C \in \mathbb{R}^{n \times 2}$, a matrix containing the coordinates of n customers in $\mathbb{R}^2$, and $D \in \mathbb{R}^{n \times n}$ a diagonal matrix containing demand values for each customer, solving the planar facility location problem* (3.0.1) *is equivalent to solving Projective Semidefinite Programming Problem* (3.1.8).

Proof. To establish equivalence between the two optimization problems we begin by showing that a feasible solution for (3.1.8) can be obtained from a feasible solution for (3.0.1), and vice versa.

It follow immediately from the construction of (3.1.8) that if X is an assignment matrix satisfying the constraints in (3.0.1), then by defining

$$Z := Y^T \left(YY^T\right)^{-1} Y, \quad \text{where } Y = XD^{\frac{1}{2}}, \tag{3.2.12}$$

we obtain a feasible solution to the PSDP (3.1.8). Moreover, it is important to point out that if X and Z are related by the relationship above, then the objective value of the clustering problem evaluated at X is equal to the objective value of the PSDP evaluated at Z, as can be seen from (3.1.5). Thus, to establish the equivalence between the two problems, it remains to show that given a feasible solution Z to (3.1.8) it is always possible to construct an feasible assignment matrix X satisfying (3.2.12) with matching objective values. Next we will describe the process of extracting such feasible X from matrix Z.

Let Z be a feasible solution to the PSDP. From Lemma 3.2 it follow that by rearranging rows and columns of Z we can obtain a block diagonal matrix

$$Z_k = \operatorname{diag}\left(Z_{\mathcal{J}_1\mathcal{J}_1} \ldots Z_{\mathcal{J}_k\mathcal{J}_k}\right).$$

Next we define the assignment matrix X as follows

$$X = \{x_{ij}\} = \begin{cases} 1 & \text{if } j \in \mathcal{J}_i \\ 0 & \text{othewise.} \end{cases} \tag{3.2.13}$$

To see that the resulting assignment matrix is feasible to (3.0.1) we can think of the resulting k sets $\mathcal{J}_i$, $i = 1, \ldots k$, as k facilities, and the indices they contain can be seen as customers serviced by that facility. Since after assigning an index

j_p to a set $\mathcal{J}_p$ it is removed and a smaller submatrix $Z_{\overline{\mathcal{J}}_p \overline{\mathcal{J}}_p}$ is considered, we are guaranteed that each customer will only be assigned to one facility, as required to satisfy the feasibility conditions in (3.0.1).

It remains to show that if we let $Y := XD^{\frac{1}{2}}$, where X is a feasible assignment matrix obtained from feasible Z, then $Z = Y^T \left(YY^T\right)^{-1} Y$. Indeed, we have that

$$\left(Y^T \left(YY^T\right)^{-1} Y\right)_{pq} = \sum_{r=1}^{k} \frac{x_{rp} D_p^{\frac{1}{2}} x_{rq} D_{qq}^{\frac{1}{2}}}{\sum_{m=1}^{n} x_{rm} D_m}. \tag{3.2.14}$$

To have a better idea what the above expression represents, we can think of the outer summation as going thought each set $\mathcal{J}_r$ for $r = 1, \ldots, k$. Using definition of X we see that (3.2.14) will be equal to zero, unless both p and q belong to some $\mathcal{J}_{r^*}$, in which case both x_{r^*p} and x_{r^*q} will be equal to one, indicating that customer c_p and c_q are assigned to facility f_{r^*}. From the discussion above we know that the resulting assignment matrix X is feasible, hence each customer is served by only one facility, this implies that all the other elements in the summation will be zero. Thus, we have

$$(Y^T \left(YY^T\right)^{-1} Y)_{pq} = \begin{cases} \frac{D_p^{\frac{1}{2}} D_{qq}^{\frac{1}{2}}}{\sum_{m \in \mathcal{J}_{r^*}} D_m} & \text{if there exists } r^* \in \{1, \ldots, k\} \text{ such that } p, q \in \mathcal{J}_{r^*} \\ 0 & \text{otherwise,} \end{cases}$$

which is precisely the definition of Z_{pq}, supporting the claim that the assignment matrix X obtained from Z satisfies (3.2.12).

In summary, from the construction of (3.1.8) we know that given a feasible solution X to the weighted clustering problem (3.0.1), we can obtain a feasible solution to (3.1.8) by letting $Z := Y^T(YY^T)^{-1}Y$, where $Y = XD^{\frac{1}{2}}$. Later we have shown that given a feasible solution Z for PSDP problem (3.1.8) we can obtain a feasible solution X for the clustering problem, satisfying $Z = Y^T(YY^T)^{-1}Y$, where Y is defined as above. In addition, it follows from (3.1.5) that at the corresponding feasible points Z and X the objective function values coincide. Combining these observations we conclude that solving Projective Semidefininte Programming problem is equivalent to solving the weighted K-means clustering problem, which completes the proof. □

Although the resulting PSDP problem (3.1.8) is difficult to solve exactly, various relaxations of the problem can be solved to obtain a lower bound on the optimal value. We will discuss a relaxation algorithm in a later section, but first we demonstrate a method of obtaining exact solutions to the above problem of a small size.

3.3 Solving SDP : $n = 3, k = 2$

In this section we demonstrate a way to solve a small instance of the problem discussed in the previous section. Although the method amounts to complete enumeration, it is motivated by the geometrical properties of the PSDP and highlights several interesting properties of the feasible solutions of the problem.

Let us take a closer look the constraints in (3.1.8). As was mentioned before, the constraints $Z^2 = Z,\ Z^T = Z$ indicate that Z is an orthogonal projection matrix, which implies that it has eigenvalues equal to 0 or 1. Moreover, the eigenspace of Z corresponding to zero eigenvalue is the null space of the projection, and eigenspace corresponding to 1 is the range of the projection. Hence, the constraints $\mathrm{Tr}(Z) = k,\ Zd = d$ together with the fact that trace of a matrix is equal to the sum of its eigenvalues imply that the range of the projection induced by Z is a subspace of R^n of dimension k containing vector $d = (eD^{\frac{1}{2}})^T$. And finally to satisfy feasibility we need to select projection matrices with nonnegative entries, as required by the constraint $Z \geq 0$.

Above observations suggest a possible approach for solving (3.1.8) numerically. The goal is to find an appropriate basis for the range of the projection. Since we know that it must contain vector d, as dictated by the feasibility requirement, it remains to find other $k-1$ linearly independent vectors, $v_1, \ldots, v_{k-1}$. We then let $A \in \mathbb{R}^{k\times n}$ be a matrix containing basis vectors $\{d, v_1, \ldots v_{k-1}\}$ as rows, and define the projection matrix $Z := A^T(AA^T)^{-1}A$. If all entries in Z are positive, we have constructed a feasible solution to (3.1.8). To illustrate the ideas discussed above consider the following example:

Let $n = 3,\ k = 2,\ D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 9 \end{pmatrix},\ C = \begin{pmatrix} 1 & 6 \\ 3 & 1 \\ 5 & 5 \end{pmatrix}.$

The problem becomes:

$$\begin{aligned}
&\text{minimize}\ \ \mathrm{Tr}\left(\begin{pmatrix} 37 & 90 & 105 \\ 90 & 1000 & 600 \\ 105 & 600 & 450 \end{pmatrix}(I - Z)\right) \\
&\text{subject to} \\
&\qquad\quad Z^2 = Z,\ Z^T = Z, \\
&\qquad\quad Z\begin{pmatrix} 1 \\ 10 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \\ 10 \\ 3 \end{pmatrix},\ \mathrm{Tr}(Z) = 2,\ Z \geq 0.
\end{aligned}$$

We will proceed to construct the globally optimal solution Z by the method discussed above. We will show that there is a small number of feasible solutions to the problem above, which will make the task of finding the global optimum trivial.

We begin by relaxing the nonnegativity constraint $Z \geq 0$. Choosing the basis for $\mathbb{R}^3$

$$B = \{(1\ 10\ 3), (0\ 1\ 0), (0\ 0\ 1)\}$$

we would like to find a basis for the two dimensional subspace (since k=2) containing the demand vector $d = (1\ 10\ 3)$, which will result in a feasible projection matrix Z. In other words, our goal is to find a vector of the form

$$v = a(0\ 1\ 0) + b(0\ 0\ 1)$$

to form a basis $\{d, v\}$ of the desired subspace. Hence, the problem is reduced to a two dimensional problem of finding feasible (a, b). Without any regard for the magnitude of v but encompassing all the possible directions we will consider all points (a, b) on the unit circle. Using the polar coordinates we obtain the standard parametrization of the unit circle

$$a = \cos\theta,\ b = \sin\theta \quad \theta \in [0, 2\pi]$$

and use it to write

$$v = (0\ \cos\theta\ \sin\theta),\ \theta \in [0, 2\pi].$$

We now consider the following matrix, containing the basis vectors as rows

$$A = \begin{pmatrix} 1 & 10 & 3 \\ 0 & \cos\theta & \sin\theta \end{pmatrix}$$

for all $\theta \in [0, 2\pi]$. The problem is further reduced to finding θ that will result in a feasible projection matrix $Z := A^T(AA^T)^{-1}A$. Note that from the definition of A we see that for any value of θ all but one constraint $(Z \geq 0)$ are satisfied. Hence it remains to find those values of θ which will make all the entries of Z nonnegative.

We start by observing that

$$AA^T = \begin{pmatrix} 110 & 10\cos\theta + 3\sin\theta \\ 10\cos\theta + 3\sin\theta & 1 \end{pmatrix}$$

which gives

$$(AA^T)^{-1} = \frac{1}{\det(AA^T)} \begin{pmatrix} 1 & -10\cos\theta - 3\sin\theta \\ -10\cos\theta - 3\sin\theta & -110 \end{pmatrix},$$

where $\det(AA^T) = 110 - 100\cos^2\theta - 60\cos\theta\sin\theta - 9\sin^2\theta$, which can be shown to be positive for all $\theta\ \in [0, 2\pi]$ Hence to simplify the calculations it suffices to find all values of θ for which each entry of the projection matrix Z modulo $\det(AA^T)$ is positive. We begin by computing the above matrix

$\det(AA^T)Z =$

$$\begin{pmatrix} 1 & 10 + \cos\theta(-10\cos\theta - 3\sin\theta) & 3 + \sin\theta(-10\cos\theta - 3\sin\theta) \\ 10 + \cos\theta(-10\cos\theta - 3\sin\theta) & 100 - 90\cos^2\theta - 60\sin\theta\cos\theta & \cos\theta\sin\theta \\ 3 + \sin\theta(-10\cos\theta - 3\sin\theta) & \cos\theta\sin\theta & 9 + 92\sin^2\theta - 60\cos\theta\sin\theta) \end{pmatrix}$$

and plot each entry to find the values of θ satisfying the positivity constraint.

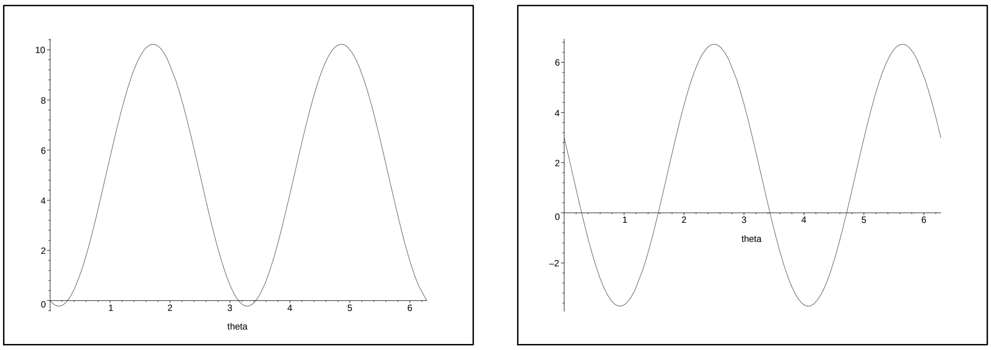

Figure 3.1: Left: $Z_{12} = Z_{21}$, Right: $Z_{13} = Z_{31}$

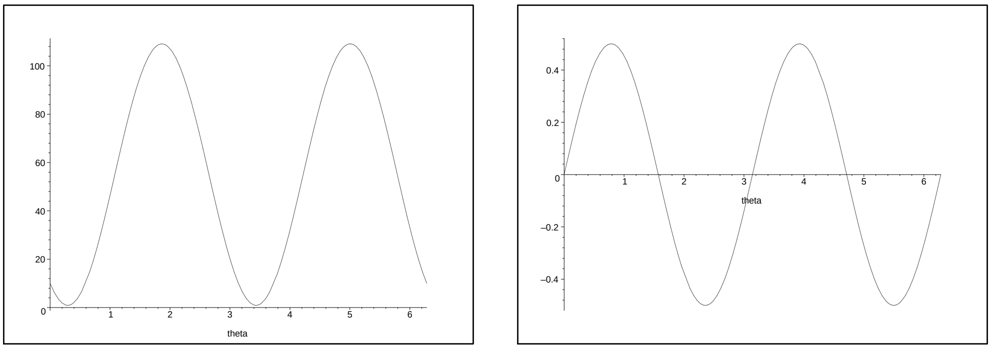

Figure 3.2: Left:Z_{22}, Right: $Z_{32} = Z_{23}$

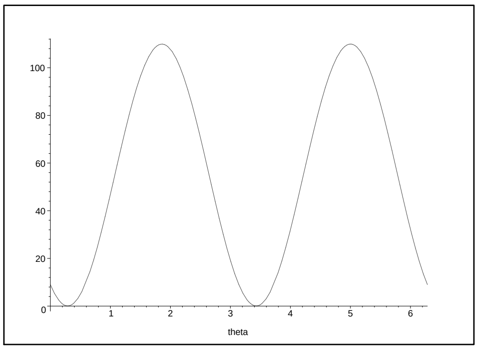

Figure 3.3: Z_{33}

By superimposing the above graphs we conclude that each entry in Z is nonnegative for the following values of θ

$$\theta_1 = 0,\ \theta_2 = \arctan\left(\frac{3}{10}\right),\ \theta_3 = \frac{\pi}{2},\ \theta_4 = \pi,\ \theta_5 = \arctan\left(\frac{3}{10}\right) + \pi,\ \theta_6 = \frac{3\pi}{2}.$$

Using above values we obtain the complete list of feasible solutions with corresponding objective values:

θ	Z	Objective value
0	$\begin{pmatrix} 0.1 & 0 & 0.3 \\ 0 & 1 & 0 \\ 0.3 & 0 & 0.9 \end{pmatrix}$	15.3
$\arctan(\frac{3}{10})$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.9174 & 0.2752 \\ 0 & 0.2752 & 0.0826 \end{pmatrix}$	165.1376
$\frac{\pi}{2}$	$\begin{pmatrix} 0.0099 & 0.0990 & 0 \\ 0.0990 & 0.9901 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	28.7129
π	$\begin{pmatrix} 0.1 & 0 & 0.3 \\ 0 & 1 & 0 \\ 0.3 & 0 & 0.9 \end{pmatrix}$	15.3
$\arctan(\frac{3}{10}) + \pi$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.9174 & 0.2752 \\ 0 & 0.2752 & 0.0826 \end{pmatrix}$	165.1376
$\frac{3\pi}{2}$	$\begin{pmatrix} 0.0099 & 0.0990 & 0 \\ 0.0990 & 0.9901 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	28.7129

Table 3.1: Angle θ, resulting feasible solution Z and corresponding objective values

From the above table we see that the global minimum 15.3 occurs at

$$Z = \begin{pmatrix} 0.1 & 0 & 0.3 \\ 0 & 1 & 0 \\ 0.3 & 0 & 0.9 \end{pmatrix}$$

Interestingly, there are only 6 values of θ satisfying the nonnegativity requirement, and surprisingly there is no interval satisfying the condition.

Although we have found the optimal solution to the PSDP problem, in order to emphasize the connection with the planar facility location problem we will proceed to extract the optimal facility locations using the method presented in the proof of Theorem 1. We begin by constructing the index sets $\mathcal{J}_1 = \{1,3\}$, $\mathcal{J}_2 = \{2\}$ and the resulting block diagonal matrix

$$Z_1 = \begin{pmatrix} 0.1 & 0.3 & 0 \\ 0.3 & 0.9 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Finally from (3.2.13) we obtain the following assignment matrix

$$X = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}. \tag{3.3.1}$$

As was discussed in the context of the weighted K-means algorithm, knowing the assignment matrix allows us to calculate the optimal facility location by simply defining them to be the centre of mass of each cluster. Thus, we obtain the following optimal facility locations:

$$f_1 = (4.6, 5.1), \;\; f_2 = (3, 1).$$

We illustrate the solution to the planar facility location problem below.

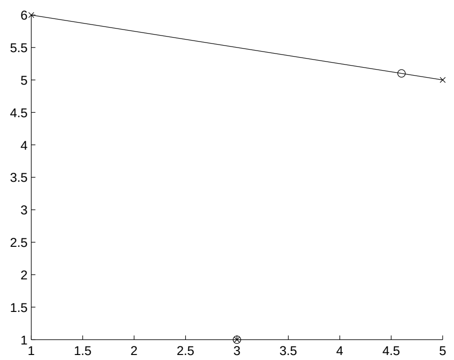

Figure 3.4: Customers : x; Facilities : o.

It is worth mentioning that the procedure for finding exact solution to the PSDP problem (3.1.8) discussed in this section can only be applied to very small problems. Consider increasing the number of customers from 3 to 4. The task is now to find a suitable two dimensional subspace of $\mathbb{R}^4$, assuming $k = 2$. Taking into account that the basis must contain the demand vector d and to guarantee linear independence we are to find

$$v = a(0\ 1\ 0\ 0) + b(0\ 0\ 1\ 0) + c(0\ 0\ 0\ 1)$$

which involves considering all points (a, b, c) on the unit sphere. As a result of using spherical coordinates the problem is reduced to finding θ and ϕ resulting in positive entries in the corresponding projection matrix Z. Although algebraically complicated, with enough effort the problem can be solved using procedure analogous to the one used for $n = 3$.

We conclude that although the method for obtaining exact solution to the Projective Semidefinite Programming problem (3.1.8) discussed in this section highlights geometrical aspect of the problem, it is not applicable to problems of higher dimension due to high computational difficulty. Next we discuss a method for obtaining a solution to a relaxation of the given problem.

3.4 Relaxation Algorithm for PSDP

In this section we briefly describe an algorithm for solving a relaxation of (3.1.8). It is worth pointing out, however, that even though the algorithm will not generally provide a solution to (3.1.8), it will certainly be useful in establishing a lower bound on the optimal value of the Projective Semidefinite Programming problem. It follows that if the algorithm produces a feasible solution to (3.1.8) it must be the optimal one. If the relaxation solution is not feasible, a rounding procedure can be used to extract a feasible solution, which will not be a part of our current discussion.

We begin by replacing the constraint $Z^2 = Z$ with $I \succeq Z \succeq 0$, where as before the relation $A \succeq B$ means that $A - B$ is positive semidefinite. Lastly, dropping the nonnegativity constraint $Z \geq 0$ we obtain the following relaxation of the original problem (3.1.8)

$$\begin{array}{ll} \text{minimize} & \operatorname{Tr}\left((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T(I - Z)\right) \\ \text{subject to} & \\ & \operatorname{Tr}(Z) = k,\ Zd = d \\ & Z^T = Z,\ I \succeq Z \succeq 0. \end{array} \tag{3.4.1}$$

We notice that any feasible solution for (3.4.1) satisfies

$$\frac{1}{\|d\|} Zd = \frac{1}{\|d\|} d$$

which simply implies that $\frac{1}{\|d\|}d$ is a unit eigenvector of Z corresponding to its largest eigenvalue 1. Next we will further reduce (3.4.1) to a simpler semidefinite programming problem by projecting it to the orthogonal complement of the unit eigenvector. The resulting formulation will be much easier to deal with than the original problem due to the absence of the constraint $Zd = d$.

We define the following orthogonal projection matrix

$$P^{\perp} = I - \frac{1}{\|d\|^2} dd^T$$

and recalling that $Zd = d$ observe that

$$Z^{\star} := P^{\perp} Z = Z - \frac{1}{\|d\|^2} dd^T Z = Z - \frac{1}{\|d\|^2} dd^T = P^{\perp} Z P^{\perp}. \tag{3.4.2}$$

The last equality is obtained using the fact that $P^{\perp} Z = Z - \frac{1}{\|d\|^2} dd^T$ is symmetric along with the properties of projection matrices in the following way:

$$P^{\perp} Z P^{\perp} = P^{\perp} (Z P^{\perp})^T = P^{\perp} P^{\perp} Z = P^{\perp} Z.$$

Remark 2. Above matrix $Z^{\star}$ represents an orthogonal projection of columns of Z onto the orthogonal complement of $\frac{d}{\|d\|}$. Indeed, denoting jth column of $Z^{\star}$ by $(Z^{\star})_j$ we see that

$$(Z^{\star})_j = Z_j - \frac{(d^T Z_j) d}{\|d\|^2},$$

supporting our observation.

Using (3.4.2) we notice that

$$\mathrm{Tr}(Z^{\star}) = \mathrm{Tr}(Z) - \mathrm{Tr}(\frac{1}{\|d\|^2} dd^T) = k - 1.$$

Finally, projecting data matrix $(D^{\frac{1}{2}} C)(D^{\frac{1}{2}} C)^T$ onto the orthogonal complement of d we define

$$((D^{\frac{1}{2}} C)(D^{\frac{1}{2}} C)^T)^{\star} := P^{\perp} (D^{\frac{1}{2}} C)(D^{\frac{1}{2}} C)^T$$

and reduce problem (3.4.1) to the following simpler semidefinite programming problem

$$\begin{array}{ll} \text{minimize} & \mathrm{Tr}\left(((D^{\frac{1}{2}} C)(D^{\frac{1}{2}} C)^T)^{\star} (I - Z^{\star})\right) \\ \text{subject to} & \\ & \mathrm{Tr}(Z^{\star}) = k - 1, \\ & I \succeq Z^{\star} \succeq 0. \end{array} \tag{3.4.3}$$

Remark 3. It is important to note that solving (3.4.3) is equivalent to solving (3.4.1). Indeed, let $Z^{\star}$ be the optimum solution to (3.4.3). It follows from (3.4.2) that $Z = Z^{\star} + \frac{1}{\|d\|^2} dd^T$ is a feasible solution to (3.4.1). Moreover, from (3.4.2) we observe that the objective functions of the two problems are the same, as can be seen from

$$
\begin{aligned}
&\operatorname{Tr}\left(((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)^\star(I-Z^\star)\right) \qquad (3.4.4)\\
&= \operatorname{Tr}\left(((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)^\star\right) - \operatorname{Tr}\left(((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)^\star Z^\star\right)\\
&= \operatorname{Tr}\left(((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)^\star\right) - \operatorname{Tr}\left((I-\frac{1}{\|d\|^2}dd^T)Z(I-\frac{1}{\|d\|^2}dd^T)(D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)\right)\\
&= \operatorname{Tr}\left(((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)^\star\right) - \operatorname{Tr}\left((I-\frac{1}{\|d\|^2}dd^T)Z(D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)\right)\\
&= \operatorname{Tr}\left(((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)^\star\right) - \operatorname{Tr}\left(Z(D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)\right) - \frac{1}{\|d\|^2}\operatorname{Tr}\left(dd^TZ(D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)\right)\\
&= \operatorname{Tr}\left((I-\frac{dd^T}{\|d\|^2})(D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T\right) - \operatorname{Tr}\left(Z(D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)\right) - \frac{1}{\|d\|^2}\operatorname{Tr}\left(dd^TZ(D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)\right)\\
&= \operatorname{Tr}\left((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T(I-Z)\right).
\end{aligned}
$$

Above observations suggest that $Z^\star$ is an optimal solution of (3.4.3) if and only if Z is an optimal solution of (3.4.1).

We have mentioned before that projecting the relaxed semidefinite programming problem onto the orthogonal complement of the unit eigenvector $\frac{1}{\|d\|^2}d$ will result in a more manageable problem. We will now state a theorem [9] which will allow us to obtain the solution for (3.4.3).

Theorem 2. *Let $\mathcal{S}_n$ denote the space of n by n symmetric matrices, and let $\Phi_{n,k} = \{U \in \mathcal{S}_n \mid 0 \preceq U \preceq I,\ Tr(U) = k\}$. Let $A \in \mathcal{S}_n$ have eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$. Then*

$$\max_{U\in\Phi_{n,k}} Tr(AU) = \sum_{i=1}^{k} \lambda_i.$$

At this point it is important to point out that adding the constraint $Z \succeq Z^2$ to (3.4.1) would improve the relaxation, but the resulting problem would be significantly more difficult to solve than (3.4.1), consequently, the tighter relaxation will not be considered in our discussion and will be left for future investigation.

We are now ready to apply the above result to our problem. Let $\lambda_1, \ldots, \lambda_{k-1}$ denote $k-1$ largest eigenvalues of $((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)^\star$ in decreasing order. Noting that minimizing $\operatorname{Tr}\left(((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)^\star(I-Z^\star)\right)$ is equivalent to maximizing $\operatorname{Tr}\left(((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)^\star Z^\star\right)$, and applying Theorem 2 we conclude that the optimal solution of (3.4.3) is achieved if and only if

$$\operatorname{Tr}\left(((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)^\star Z^\star\right) = \sum_{i=1}^{k-1} \lambda_i. \qquad (3.4.5)$$

It is easy to verify that the solution to the above equation is given by

$$Z^\star = \sum_{i=1}^{k-1} v_i v_i^T,$$

where v_i is the unit eigenvector corresponding to eigenvalue λ_i. Indeed,

$$\begin{aligned}
\operatorname{Tr}\left(((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)^\star \sum_{i=1}^{k-1} v_i v_i^T\right) &= \operatorname{Tr}\left(\sum_{i=1}^{k-1}((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)^\star v_i v_i^T\right) \\
&= \operatorname{Tr}\left(\sum_{i=1}^{k-1}\lambda_i v_i v_i^T\right) \\
&= \sum_{i=1}^{k-1}\lambda_i \|v_i\|^2 = \sum_{i=1}^{k-1}\lambda_i.
\end{aligned}$$

We can summarize the above discussion with the following algorithm for solving (3.4.3) and the relaxation of the semidefinite programming formulation of the planar facility location problem:

Relaxation Algorithm

1. Calculate $((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)^\star$.
2. Compute the first $k-1$ largest eigenvalues of $((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)^\star$ and corresponding unit eigenvectors $v_1, \ldots, v_{k-1}$.
3. Set $Z = \frac{1}{\|d\|^2} dd^T + \sum_{i=1}^{k-1} v_i v_i^T$.

We have mentioned before that the solution to the relaxed problem (3.4.1) provides a lower bound on the optimal value of (3.1.8). At this point it is important to note that the above relaxation algorithm will provide a meaningful lower bound only for $k \le 2$. This follows from the fact that the matrix $((D^{\frac{1}{2}}C)(D^{\frac{1}{2}}C)^T)^\star$ has rank equal to 2, since C does. This implies that it has 2 nonzero eigenvalues. Thus, for $k \ge 3$ the optimal Z will result in the objective value equal to 0, as can be seen from (3.4.4) and (3.4.5).

In the previous section we considered a method of solving (3.1.8) for $n = 3$ and $k = 2$, by constructing a set of feasible solutions from which the optimal solution was chosen. We pointed out that the resulting feasible set was small enough to make the above approach effective, and that the nonnegativity requirement $Z \ge 0$ proved to be the most restrictive. The results of the implementation of the above relaxation algorithm with $k = 2$ support our earlier observations. In majority of cases the optimal solutions to (3.4.1) obtained from the algorithm violated the feasibility requirements of (3.1.8) by containing negative entries. It is worth mentioning, however, that in some cases, especially for small problem sizes, the algorithm produced optimal solution to (3.1.8). We include several examples below.

Demand matrix D	Customer locations C	Optimal solution Z
$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 9 \end{pmatrix}$	$\begin{pmatrix} 3 & 10 \\ 5 & 6 \\ 1 & 5 \end{pmatrix}$	$\begin{pmatrix} 0.2 & 0.4 & 0 \\ 0.4 & 0.8 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 9 \end{pmatrix}$	$\begin{pmatrix} 4 & 5 \\ 10 & 8 \\ 8 & 3 \end{pmatrix}$	$\begin{pmatrix} 0.1 & 0 & 0.3 \\ 0 & 1 & 0 \\ 0.3 & 0 & 0.9 \end{pmatrix}$
$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 10 \end{pmatrix}$	$\begin{pmatrix} 1 & 8 \\ 1 & 3 \\ 7 & 4 \\ 1 & 4 \end{pmatrix}$	$\begin{pmatrix} 0.0667 & 0.1333 & 0 & 0.2108 \\ 0.1333 & 0.2667 & 0 & 0.4216 \\ 0 & 0 & 1 & 0 \\ 0.2108 & 0.4216 & 0 & 0.6667 \end{pmatrix}$
$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 0 & 25 \end{pmatrix}$	$\begin{pmatrix} 5 & 1 \\ 3 & 10 \\ 2 & 1 \\ 9 & 10 \\ 10 & 1 \end{pmatrix}$	$\begin{pmatrix} 0.0286 & 0 & 0.0857 & 0 & 0.1429 \\ 0 & 0.2 & 0 & 0.4 & 0 \\ 0.0857 & 0 & 0.2571 & 0 & 0.4286 \\ 0 & 0.4 & 0 & 0.8 & 0 \\ 0.1429 & 0 & 0.4286 & 0 & 0.7143 \end{pmatrix}$

Table 3.2: Examples of optimal solutions Z obtained from the relaxation algorithm with $k = 2$ and data matrices D and C, which are also optimal to the PSDP.

In conclusion, we will compare the performance of the mixed integer programming formulations discussed in the earlier chapter, weighted K-means heuristic and PSDP relaxation algorithm in solving the planar facility location problem with $k = 2$. We will consider best lower bound and best upper bound produced by the mixed integer program in 2 hour time limit, best objective value found by weighted K-means algorithm starting with 1000 random initializations, and the solution of the PSDP relaxation algorithm. We demonstrate the results in Table 3.3.

Number of Customers n	Weighted K-means	MIP best LB	MIP best UB	PSDP Relaxation
16	8953.26	6034.4	6034.4	4374.6
25	12434.6	11164.6	11164.6	8498.2
36	16309.1	16309.1	16309.1	11891
49	31257.5	23612.9	23614.4	19046
64	36025.2	32676.3	32679.5	25702
81	47571.9	34863	38587.2	31903
100	50579.8	30779.2	50557.3	38572

Table 3.3: Comparing optimal solutions to the planar facility location problem with $k = 2$. Objective values obtained by solving mixed integer program, applying weighted K-means algorithm and PSDP relaxation algorithm.

In this chapter we extended the classical K-means heuristic to take into account demands associated with each customer, which made it applicable to the planar facility location problem. We then showed that solving the planar facility location problem is equivalent to solving a projective semidefinite programming problem and briefly discussed a relaxation algorithm, which provides a nontrivial lower bound on the total costs for small values of k. This suggests the need to explore tighter relaxations, which may result in more meaningful lower bounds for $k \geq 3$, or investigate rounding techniques to extract a feasible solution from optimal Z, which can be used to improve the computational efficiency of the mixed-integer programming formulations of the planar facility location problems.

Chapter 4

A Preliminary Look at Continuous Demand in Facility Location Problem

In this chapter we extend the planar facility location problem discussed in the previous chapters. The new framework will be characterized by the assumption that customers are spread over some region $\Omega \in \mathbb{R}^2$, resulting in a continuous demand function $D : \Omega \to \mathbb{R}$ describing customer demand distribution.

As pointed out by the authors in [3], continuous facility location models are applicable to large scale problems, provide a more accurate closed form solution, and have lesser data requirements in contrast to the discrete models, where a demand value has to be obtained for each customer. As a result, when dealing with discrete models, an aggregation of customers into a smaller set is often performed to reduce the problem to a manageable size and reduce costs associated with data collection at the cost of accuracy. It is often the case, however, that discrete models are easier to solve, once all the necessary data has been obtained.

To avoid the confusion, recall that we have discussed two different discrete facility location models; one with discrete customer locations and continuous facility locations, which has been referred to as the planar facility location model, and the other where both customer locations and facility locations are discrete. The second formulation is referred to as the discrete formulation of the facility location model and it was briefly discussed in Section 1.1.

In this chapter, using an illustrative example, we will address some of the practical issues companies have to deal with when choosing optimal facility locations under the continuous customer distribution model. To combine the strengths of discrete and continuous approaches, we will begin by describing the process of obtaining an approximation of the continuous demand function. Then, to take advantage of the fast computational time of the discrete formulation of the facility location problem discussed in Section 1.1, we will discretize the customers in Ω by splitting the region into an $n \times n$ grid for increasingly large values of n in an attempt to approximate the continuous distribution model. Note that in our setup, the discrete formulation is a better candidate than the planar facility location problem due to its faster computation time and ability to deal with larger data sets. Even though the discrete model results in an approximate solution due to the fact that the facility locations are chosen from a predetermined set, the availability of the continuous demand function will allow us to increase number of customers and potential facility locations, thereby improving the accuracy while retaining fast computational time.

4.1 Gaussian Demand Function

To set up the stage, we suppose that company X is planning to expand its services into southern Ontario by opening 5 new facilities which will service customers in the area. In this section, we demonstrate the process of data aggregation as a means of acquiring an approximation of the continuous demand function $D : \Omega \to \mathbb{R}$, where the region of interest Ω contains Toronto, Mississauga, Scarborough, Brampton, Hamilton, Oshawa and Waterloo.

We begin by aggregating all the customers in Ω into 7 largest cities in that area. Due to the fact that information about customer demand is costly and difficult to obtain, our goal is to use population density as well as the area of of each city as an estimate of the demand for the product. To achieve this, we make a simplifying assumption that each city is circular and use the two-dimensional Gaussian function

$$D_j(x, y) = He^{\frac{-((x-p_1)^2+(y-p_2)^2)}{\sigma^2}} \tag{4.1.1}$$

where (p_1, p_2) is the centre of the distribution, σ is standard deviation and H is the height of the function at the centre. After constructing Gaussian demand function for each aggregate consumer, we will disaggregate by defining the total demand function to be the sum of the individual functions D_j.

We begin by listing the location of each customer enumerated by an integer $j = 1, \ldots, 7$, which will be the centre of the demand distribution for corresponding city.

City	Coordinates of Centre
1. Toronto	$(90, 50)$
2. Mississauga	$(74, 40)$
3. Scarborough	(132,59)
4. Brampton	(75,70)
5. Oshawa	(175,74)
6. Hamilton	(60,4)
7. Waterloo	(7,26)

Table 4.1: Aggregate customer locations in Ω.

The above coordinates were obtained using latitude and longitude measurements to calculate horizontal and vertical distances between the cities and by choosing an origin of convenience. Thus, we have the following domain Ω:

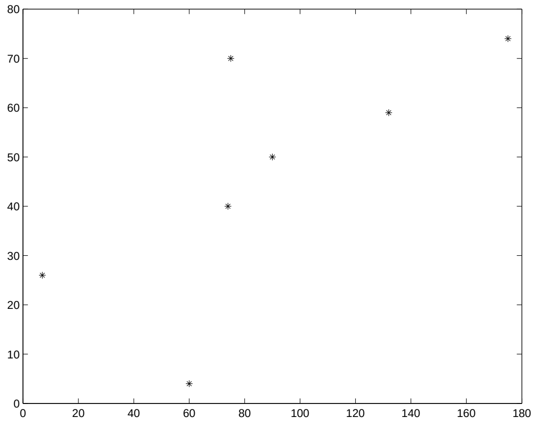

Figure 4.1: $\Omega = \{(x,y)\ |0 \leq x \leq 180,\ 0 \leq y \leq 80\}$ and customer locations.

We obtain population size and area of each city and display the data in the table below.

City	Population(P_j)	Area(A_j)
1. Toronto	2503281	630
2. Mississauga	668549	288
3. Scarborough	593297	188
4. Brampton	433806	267
5. Oshawa	14150	146
6. Hamilton	504559	138
7. Waterloo	97475	64

Table 4.2: Population and area of cities in Ω.

Next we will discuss how the data from Table 4.2 will be used to obtain approximation of standard deviation σ_j and height H_j for each city $j = 1, \ldots, 7$.

We begin by using the information about population size to obtain the height of the distribution H_j for city i. Since the Gaussian function describes the population distribution, we know that

$$\text{Total Population of city } j = P_j = \int He^{\frac{-((x-p_1)^2+(y-p_2)^2)}{\sigma^2}}$$

where (p_1, p_2) and σ are coordinates of the centre and radius of city i, respectively. Therefore we obtain the following equation for the height H_j:

$$P_j = H_j \sigma_j \sqrt{2\pi}. \tag{4.1.2}$$

We will use the area measure A_j to obtain an approximation of the standard deviation σ_j for each customer $j = 1, \ldots, 7$. We will apply so called "68-95-99.7 rule" or the "empirical rule", which states that about 99.7% of values drawn from a standard normal distribution are within 3 standard deviations away from the mean. Assuming that each city is circular, the rule simply states that 99.7% of people live within the radius of 3σ from the centre of the city. As a result we obtain the following approximation:

$$\sigma_j \approx \frac{1}{3} R_j. \tag{4.1.3}$$

Using the formula for the area of the circle $A_j = \pi R_j^2$, we obtain the value of the radius R_j for each city, then using (4.1.3) we find the values of σ_j, and finally using (4.1.2) and values of σ_j we find H_j for each i. We summarize our findings in the following table.

City	σ_j	H_j
1. Toronto	4.7203	211570
2. Mississauga	3.1915	83569
3. Scarborough	2.5786	91791
4. Brampton	3.0730	56318
5. Oshawa	2.2724	2484.2
6. Hamilton	2.2092	91113
7. Waterloo	1.5045	25847

Table 4.3: Standard deviation σ_j and height H_j for each customer $j = 1, \ldots, 7$.

Now using the data from Table 4.3 and equation (4.1.1) we obtain demand function for each city $j = i, \ldots, 7$, which we then sum together to obtain the continuous demand function:

$$\begin{aligned} D(x,y) =& 211570 e^{\frac{-((x-90)^2+(y-50)^2)}{4.7203^2}} + 83569 e^{\frac{-((x-74)^2+(y-40)^2)}{3.1915^2}} \\ &+ 91791 e^{\frac{-((x-132)^2+(y-59)^2)}{2.5786^2}} + 56318 e^{\frac{-((x-75)^2+(y-70)^2)}{3.0730^2}} \\ &+ 2484.2 e^{\frac{-((x-175)^2+(y-74)^2)}{2.2724^2}} + 91113 e^{\frac{-((x-60)^2+(y-4)^2)}{2.2092^2}} \\ &+ 25847 e^{\frac{-((x-7)^2+(y-26)^2)}{1.5045^2}}. \end{aligned}$$

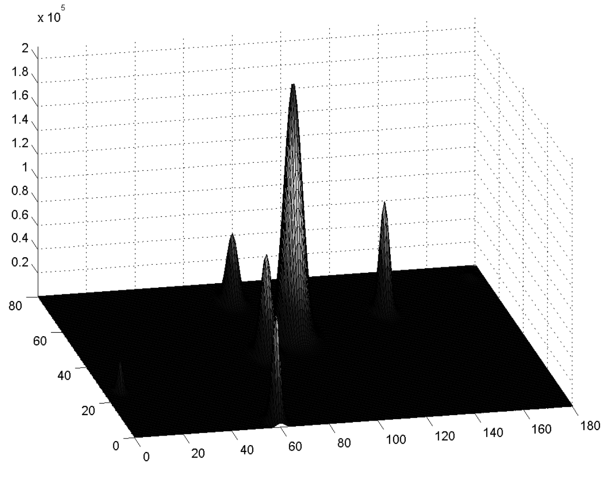

Figure 4.2: Gaussian demand function $D : \Omega \to \mathbb{R}$.

In the next section we will use the continuous demand function to solve the problem stated in the beginning of this section.

4.2 Discrete Facility Location Model

In the previous section, we pointed out the benefits and downsides of the continuous and discrete approaches to the facility location model. The continuous formulation requires one continuous demand function, in contrast to the discrete formulation, which in order to produce results of comparable accuracy relies on a large number of demand values.

Since the continuous facility location is very difficult to solve directly, we will obtain an approximate solution to the problem by solving the discrete model. Note that by increasing the number of aggregate customers and the set of potential facility locations we increase the accuracy of the approximate solution, while still having high computational efficiency.

Recall that in the discrete version of the facility location problem the set of customer locations is finite and optimal locations are chosen from a set of predetermined potential sites. Therefore we will aggregate the customers in Ω by dividing the region into $n \times n$ grid and placing a representative customer and a potential facility location into the centre of each square. We obtain the associated demand values for each aggregate customer in the grid by evaluating the integral of the continuous demand function $D : \Omega \to \mathbb{R}$ over the corresponding square of the grid, as a result keeping the total population constant.

Starting with $n = 20$ we refine the grid by increasing n to 30 and later 40, obtaining a more accurate solution to our problem. We illustrate the outcome in the figures below, where customers and potential outlet locations are represented by points, and cities are marked by an x, the lines indicate the assignment of customers with facilities. Customers with zero demand are not served by any facility. Along with each figure we display the computational time, total costs and list optimal facility locations.

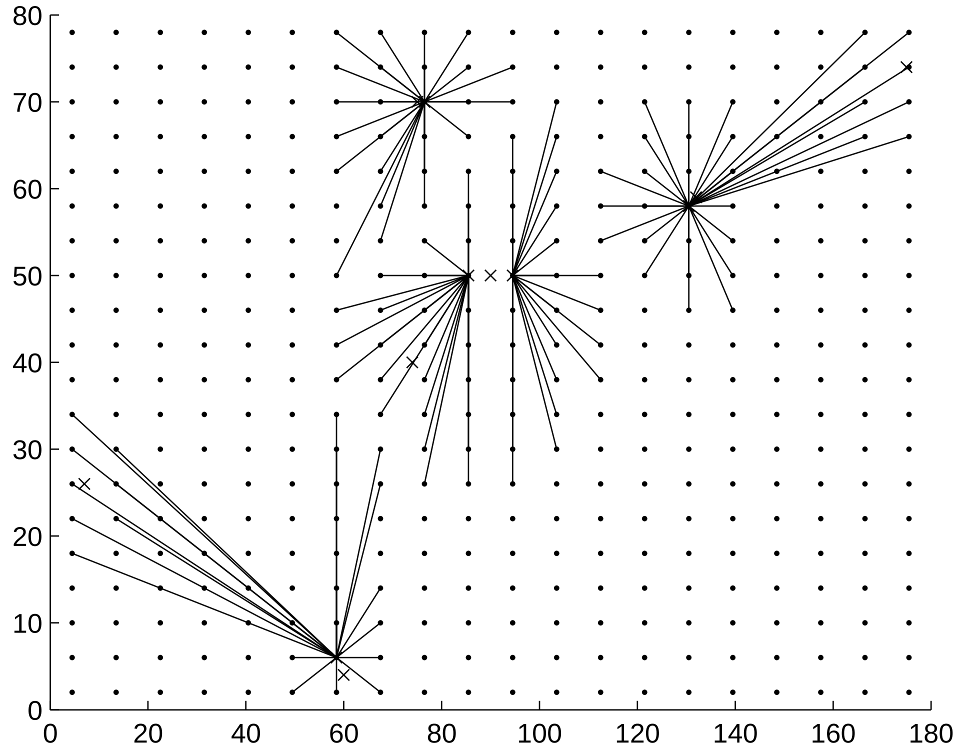

Figure 4.3: 400 customers, 5 Facilities

Computational time	45.9 seconds
Total costs	$C_{400} = 1.0008 \times 10^8$
Facility Locations	$\Phi_{400} = \{(58.5, 6), (85.5, 50), (94.5, 50), (130.5, 58), (76.5, 70)\}$

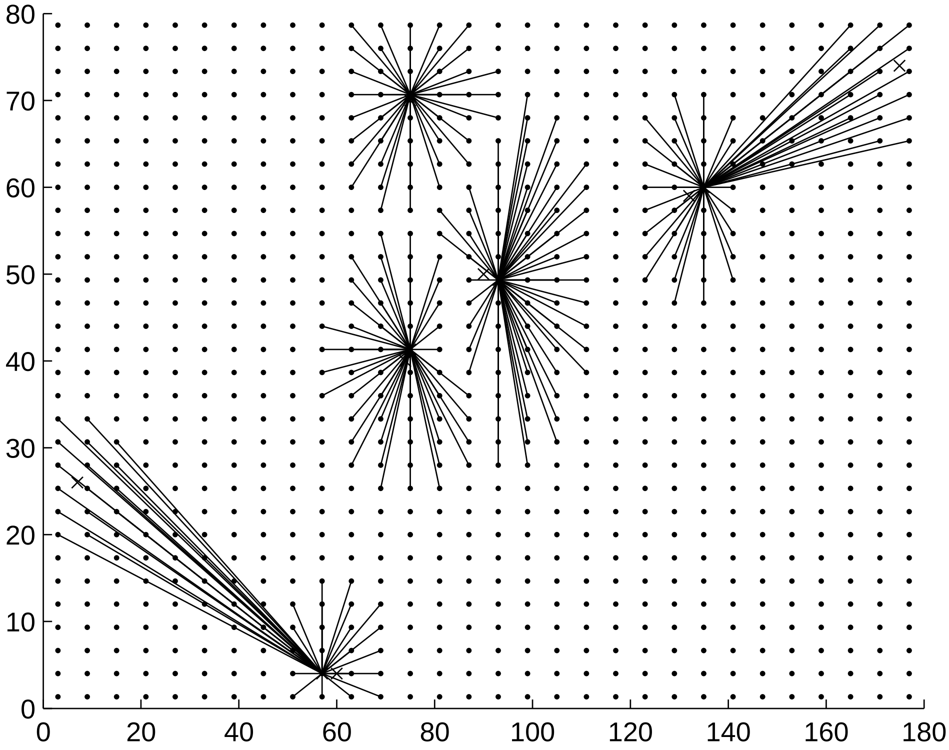

Figure 4.4: 900 customers, 5 Facilities

Computational time	647.9 seconds
Total costs	$C_{900} = 1.1114 \times 10^8$
Facility Locations	$\Phi_{900} = \{(57, 4), (75, 41.33), (93, 49.33), (135, 60), (75, 70.67)\}$

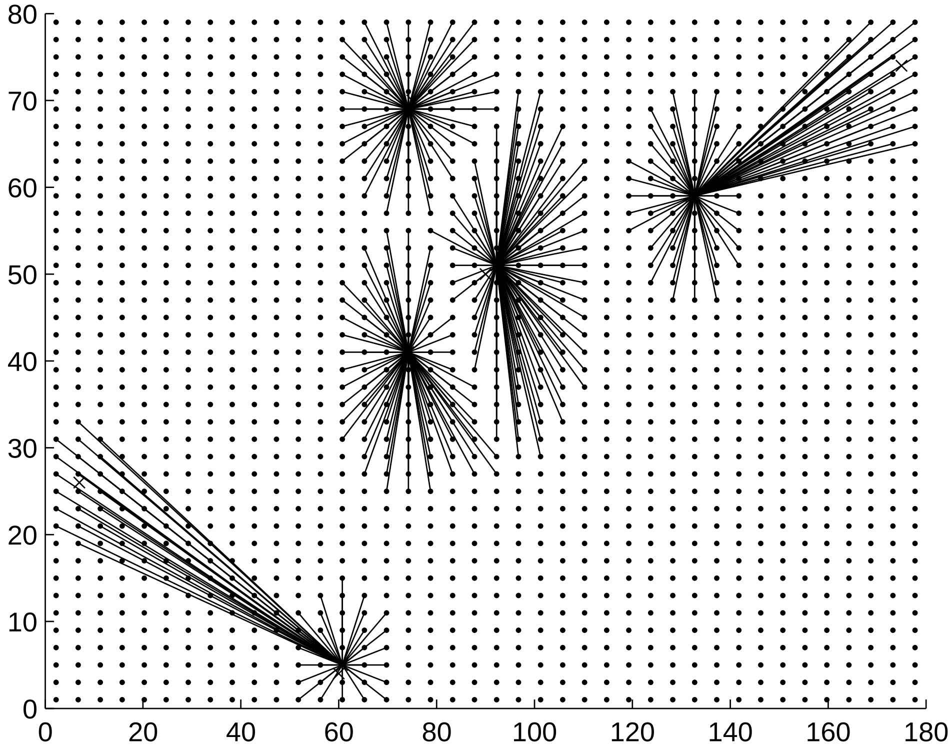

Figure 4.5: 1600 customers, 5 facilities

Computational time	5709.44 seconds
Total costs	$C_{1600} = 1.0313 \times 10^8$
Facility Locations	$\Phi_{1600} = (60.75, 5), (74.25, 41), (92.25, 51), (132.75, 59), (74.25, 69)$

It is important to point out that as the number of potential facility locations and the number of aggregate customers increase, there are several factors affecting the total cost.

First, as was mentioned by the authors in [13], one of the customer aggregation errors results in underestimation of the total costs. This follows from the fact that when the facility is located at an aggregate demand point, the resulting cost of transportation to the customers belonging to this aggregate point is zero, when in reality it is not. Thus, as number of aggregate data points increases, so do the total costs.

On the other hand, as the number of potential facility locations increases, the model produces a more accurate optimal solution due to the higher availability of location possibilities. As a result, as the number of potential facility sites increases, the total costs decrease.

Since in the discrete version of the facility location model the set of aggregate customers coincides with the set of potential facility locations, the two effects discussed above offset each other, as can be seen by considering total costs for the above solutions. Comparing the costs we see that in the case of 900 customers, the total costs are $C_{900} = 1.1114 \times 10^8$, and for 1600 customers the total costs are $C_{1600} = 1.0313 \times 10^8$. In this case, the improved accuracy of the optimal solution due to enlarged potential facility set had a larger effect on the total costs than the increase in number of customers.

To demonstrate the relationship between the two factors, we will first find the total cost of 900 customer model using optimal facility locations Φ_{1600}. Based on the discussion above, we expect it to be lower than C_{900}, due to higher accuracy of the solutions produced by the model with 1600 potential facility location. Using the new facility locations we find the total cost to be $1.0262 \times 10^8 < C_{900} = 1.1114 \times 10^8$, indicating the accuracy effect. Similarly, we observe that the new cost is lower than $C_{1600} = 1.0313 \times 10^8$ demonstrating the presence of the aggregation error.

To conclude the discussion we recall that in this chapter we addressed some of the practical issues related to data mining. By aggregating customers into cities we have constructed an approximate continuous demand function, reflecting population density and area of the major cities in the region. After obtaining the continuous demand function, we solved the problem of locating 5 facilities under the assumption that the customers are spread over the entire region, using the discrete version of the facility location problem.

Chapter 5

Concluding Remarks

In this thesis, we present various mathematical programming formulations of the planar facility location problem. In addition, we extended the classical K-means clustering algorithm to take into account the demand values associated with each customer, which made it applicable to the planar facility location problem and provided a fast way of obtaining approximate solutions. We shifted into an alternative framework by showing that solving the facility location problem is equivalent to finding a solution to a projective semidefinite programming problem and discussed a possible relaxation algorithm. In addition, we discussed some of the practical issues related to the facility location problem under the continuous customer distribution model. We first constructed an approximation of the continuous demand function, which we then discretized to obtain demand values for a finite number of aggregate consumers to apply the discrete formulation of the facility location model.

The preliminary computational results suggest we need to improve mixed integer formulations of the facility location problem. Introduction of strong valid inequalities can result in tighter linear programming relaxations and faster computational time. An alternative approach may use tailored branch and bound algorithms, which may prove to be effective in solving the problem.

Formulating the planar facility location as a projective semidefinite programming problem offers interesting directions in research. Investigating alternative relaxations of the PSDP may result in more meaningful lower bounds on the optimal value and developing new approximation algorithms has a potential to yield nontrivial bounds for problems of larger size.

Accurate closed form solutions and lesser reliance on data are just some of the benefits offered by the continuous facility location model, where customers are assumed to be spread over some region in the plane. The problem, however, is difficult to solve directly. We hope to explore approaches that deal directly with the continuous demand function and offer new insight into the facility location problem.

Bibliography

[1] Sanjeev Arora, Prabhakar Raghavan, and Satish Rao, *Approximation schemes for Euclidean k-medians and related problems*, STOC '98 (Dallas, TX), ACM, New York, 1999, pp. 106–113. MR MR1731569

[2] Burçin Bozkaya, Jianjun Zhang, and Erhan Erkut, *An efficient genetic algorithm for the p-median problem*, Facility location, Springer, Berlin, 2002, pp. 179–205. MR MR1933971

[3] Abdullah Dasci and Vedat Verter, *Erratum to: "A continuous model for production-distribution system design" [European J. Oper. Res.* **129** *(2001), no. 2, 287–298; 1806003]*, European J. Oper. Res. **137** (2002), no. 1, 229. MR MR1864061

[4] Zvi Drezner, Kathrin Klamroth, Anita Schöbel, and George O. Wesolowsky, *The Weber problem*, Facility location, Springer, Berlin, 2002, pp. 1–36. MR MR1933966

[5] R. L. Francis, T. J. Lowe, A. Tamir, and H. Emir-Farinas, *A framework for demand point and solution space aggregation analysis for location models*, European J. Oper. Res. **159** (2004), no. 3, 574–585. MR MR2078856 (2005b:90062)

[6] S. L. Hakimi, *Optimum distribution of switching centers in a communication network and some related graph theoretic problems*, Operations Res. **13** (1965), 462–475. MR MR0186497 (32 #3957)

[7] A.K. Jain, M.N. Murty, and P.J. Flynn, *Data clustering: A review*, ACM Computing Surveys **31** (1999).

[8] F. Margot, *Exploiting orbits in symmetric ilp*, Mathematical Programming, Ser. B **98** (2003), 321.

[9] M. L. Overton and R. S. Womersley, *Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices*, Math. Programming **62** (1993), no. 2, Ser. B, 321–357. MR MR1247620 (95b:49029)

[10] Jiming Peng and Yu Wei, *Approximating K-means-type clustering via semidefinite programming*, SIAM J. Optim. **18** (2007), no. 1, 186–205 (electronic). MR MR2299680

[11] C ReVelle and R. Swain, *Central facilities location*, Geographical Analysis **2** (1970), 30–42.

[12] H.F. Sherali and J.C. Smith, *Improving discrete model representations via symmetry considerations*, Management Science **47** (2001), no. 10, 1396–1407.

[13] Peiwu Zhao and Rajan Batta, *An aggregation approach to solving the network p-median problem with link demands*, Networks **36** (2000), no. 4, 233–241. MR MR1796240 (2001h:90012)