COVERING PROBLEM WITH MINIMUM RADIUS ENCLOSING CIRCLE

By

Eric Nartey Onyame

Lakmali Weerasena
Professor of Mathematics
(Committee Chair and Main Advisor)

Ebiefung Aniekan
Professor of Mathematics
(Committee Member)

Lani Gao
Professor of Statistics
(Committee Member)

Damitha Bandara
Professor at Business Department, UTC
(Committee Member)

COVERING PROBLEM WITH MINIMUM RADIUS ENCLOSING CIRCLE

By

Eric Nartey Onyame

A Thesis Submitted to the Faculty of the University of
Tennessee at Chattanooga in Partial
Fulfillment of the Requirements of the Degree
of Master of Science: Mathematics

The University of Tennessee at Chattanooga
Chattanooga, Tennessee

May 2023

ABSTRACT

This study extends the classical smallest enclosing circle problem in location science to optimize healthcare communication hubs. Given a set of demand points and potential groups, we identify the optimal number of subgroups to cover all points and the circle enclosing them with minimum radius. The center of this circle serves as the communication hub location, minimizing the distance between demand points and facilities subject to customer demand. We develop a nonconvex-nonlinear optimization model and propose a quadratic programming-based approximation algorithm to solve it. Tested on various hypothetical and real scenarios, our model effectively reduces the facility setup cost and identifies the optimal communication hub location.

DEDICATION

I would like to dedicate this thesis to my family. Specifically my parents, Emmanuel and Rebecca Onyame, and my dear uncles, Abraham Amewuda, Abednego Amedor, and Amankwah Kingston Anan. They have been a great source of love and encouragement. Thank you for always believing in me and giving me the confidence to work harder to achieve my dreams. I am the man I am today because of them.

ACKNOWLEDGEMENTS

This work was possible due to the support and guidance received from many people. I first want to express my sincere gratitude to the Graduate School and the Department of Mathematics at UTC for the opportunity offered to me to pursue a graduate degree in Mathematics. I especially want to thank my Mentor and Supervisor Dr. Lakmali Weerasena for her invaluable knowledge and advice to get this work done. I would also like to thank my committee members, Prof. Aniekan Ebiefung, Prof. Lani Gao, and Prof. Damitha Bandara for their critiques and advice. My sincere appreciation also goes to Chris Tompkins and Nyssa Hunt for their support with my GIS analysis in this work. Also, I will like to thank Israel Adikah and Gertrude Osei for their encouragement and support. I would like to express my utmost gratitude, with one final appreciation, to the National Science Foundation (NSF) for generously funding my graduate school fees, stipend, and health insurance. This funding has been critical in enabling me to complete my research successfully. Thank you, NSF, for your invaluable support.

TABLE OF CONTENTS

ABSTRACT iii

DEDICATION iv

ACKNOWLEDGEMENTS v

LIST OF FIGURES ix

LIST OF ABBREVIATIONS x

LIST OF TABLES xi

CHAPTER

1 INTRODUCTION 1

1.1. Literature Review 1

2 Background and Existing Methods for Mathematical Optimization 13

2.1. Basic concepts and optimization models 13

2.2. Convex optimization 14

Lines and line segments 15

Convex sets 16

Convex functions 17

2.3. Optimality 17

First-order conditions 18

Second-order conditions 18

2.4. Linear programming 19

2.5. Nonlinear programming 21

2.6. Least-squares problems 22

2.7. Global and local optimization 23

2.8. Related facility location problems-location science 24

2.9. Weber location problem 25

2.10. $P-$center problem 25

2.10.1. Approaches to analyzing the $P-$center problem 27

The absolute $P-$center problem 28

Vertex restricted $P-$center problem 28

Capacitated $P-$center problem 29

Discrete center problem 30

3 OPTIMIZATION MODELS AND ALGORITHMS 31

3.1. Optimization models 31

3.2. Computing an upper bound on R : SCP-allocate algorithm 34

4 COMPUTATIONAL WORK 44

4.1. Model evaluation- on more hypothetical test problems 44

4.2. Model evaluation- on real data test case from Dougherty county 52

Demand representation in GIS 54

Sensitivity Analysis 58

5 CONCLUSION AND FUTURE WORKS 62

5.1. Overview and summary 62

5.2. Future works 64

REFERENCES 65

APPENDIX 71

A. CODE FOR SELECTING BEST FACILITIES TO SETUP 71

B. CODE FOR FINDING THE RADIUS AND CENTER OF THE COMMUNICATION HUB (THE HEURISTIC APPROACH) 74

C. QUADRATIC APPROACH-ALL IN ONE METHOD(AIO) 77

D. DYNAMIC SELECTION OF BEST FACILITIES TO SET UP 80

E. CODE FOR GENERATING THE COVERAGE MATRIX 87

VITA 90

LIST OF FIGURES

2.1 Convex, strictly convex and non-convex sets 16

2.2 Convex and non-convex sets 17

3.1 An illustration of facilities covering a set of demand points 43

4.1 A set of clustered 50 demand points covered with 11 circles 48

4.2 Order of circle selections with Algorithm 3 with $m = 50$ and $n = 11$ 48

4.3 Final lay of produced by Algorithm 3 with $m = 50$ and $n = 11$ 49

4.4 A set of clustered 100 demand points covered with 15 circles 49

4.5 Order of circle selections where Algorithm 3 with $m = 100$ and $n = 15$ 50

4.6 Final lay of produced by Algorithm 3 with $m = 100$ and $n = 15$ 50

4.7 Communication hub covering 10 facilities with demand points 51

4.8 A map showing Dougherty county with fire stations and health centers 55

4.9 Fire stations and Health centers in Dougherty county with buffers 56

4.10 Fire stations and Health centers in Dougherty county with hexagonal lattice's 57

4.11 Fire stations and Health Centers in Dougherty county with big buffer 60

LIST OF ABBREVIATIONS

LPs, Linear Programming systems

FLP, Facility Location Problem

HLP, Hub Location Problem

MINLP, Mixed Integer Nonlinear Programming

NLP, Nonlinear Programming

SCP, Set Covering Problem

GIS, Geographic Information Systems

UCC, University of Cape Coast

EMS, Emergency Medical Services

QP-AiO, Quadratic Approach, All in One

LIST OF TABLES

1 Coordinates of centers and radius for circles in Figure 3.1a 40

2 Coordinates of the demand points in Figure 3.1a 40

3 Coverage matrix for the hypothetical test problem with $m = 15$ and $n = 6$ 41

4 Center and radius of communication hub 43

5 Results of more hypothetical test cases 47

6 Data showing the GIS coordinates for health centers in Dougherty county 53

7 The GIS coordinates for fire centers in Dougherty county 54

8 Sensitivity analysis for fire centers in Dougherty county 61

9 Sensitivity analysis for health centers in Dougherty county 61

CHAPTER 1

INTRODUCTION

1.1. Literature Review

Optimization is essential to decision-making in every society. Many significant areas of application in mathematical programming and control theory have benefited from its recent advancements to solve practical problems and this trend is expected to continue. Mathematical optimization is the science of identifying the optimal solutions to mathematically defined problems, which may represent models of physical reality or industrial and management systems (Snyman et al., 2005). It is widely used across all branches of applied mathematics, including economics, engineering, medicine, and other scientific fields. The three major components of the mathematical optimization model include the decision variables, objective functions, and constraints. Decision variables refer to the unknown quantities which are to be calculated to represent the output of the optimization solution. Constraints are the restrictions in the form of equations and inequalities on the variables used in optimization problems. The objective function is the real-valued function that is to be maximized or minimized subject to a set of constraints. Objective functions are always needed to solve optimization problems. The two main approaches used in solving optimization problems are the “exact approaches,” which ensure that an optimal solution will be discovered and the “heuristic approaches,” which are used when exact techniques are too slow or fail to obtain an exact answer (Hooker, 2015). It is preferable to use an exact optimization approach if it can find a solution to the optimization problem with an amount of work that increases polynomially with the size of the problem (Hooker, 2015). Analytical, numerical, and other optimization methods can be used to solve optimization problems. Analytical approaches only apply to problems with a high level of structure and for which the objective function can be determined with certainty. The numerical approach uses numerical techniques to calculate an objective function's maximum or minimum value. Examples of such numerical methods include the simplex and interior point methods, which are used in solving linear programming problems (LPs). The simplex method is a technique for finding the optimal solution to LPs through the use of slack variables, tableaus, and pivot variables. This approach for solving LPs is proposed by George B. Dantzig in 1947 (Dantzig, 1990). In the early 1960s, Anthony V. Fiacco and Garth P. McCormick pioneered the invention of the interior point method to solve both linear and non-linear convex optimization problems, which explores the interior of the feasible region in order to locate the optimal solution (Shanno, 2012) .

Before 1979, the simplex method was the main and most efficient approach to solving LPs since it could solve very big LPs effectively (Rothlauf, 2011). For almost all important real-world problems, the number of iterations needed to get to the best possible solution is a small multiple of the problem's dimension (n) when using the simplex method (Rothlauf, 2011). Lenid Khachian (Khachian, 1979) presented yet another approach to the interior point method in 1979 which is a subset of non-linear methods. This method inscribes a series of ellipsoids of decreasing volume in the viable region rather than searching across its convex hull. The interior point method does not search on the convex hull but approaches the optimal solution (which is a corner point) from within the feasible region (Khachian, 1979).

However, the interior-point method in most cases performs better than the simplex methods for larger problems when there is no available prior information about the optimal solution (Beasley, 1996). In contrast, when prior information about an already obtained solution is known, the simplex methods perform better than the interior-point methods (Beasley, 1996).

The branch and bound method for LPs with integer decision variables is another approach for solving optimization problems. This approach was introduced by Land and Doig in 1960. Its first practical implementation was presented by Dakin in the year 1965 (Rothlauf, 2011). Branch-and-bound algorithms, which can be applied to non-linear problems, work by recursively breaking down the original problem into smaller sub-problems (Rothlauf, 2011). The fundamental concept behind these algorithms is to recursively partition a problem into a set of sub-problems. Branching is the process of subsequently adding more constraints to the initial problem, which may be depicted using hierarchical tree structures. This process can be done in an iterative manner. The process of eliminating (also known as "killing" or "fathoming") sub-problems from the further examination is referred to as "bounding." Sub-problems that have been eliminated from consideration that are not further broken down into sub-problems are not taken into account. If a bound is lower than an existing lower bound, then the sub-problems are eliminated (maximization problem). Computationally developed algorithms are applied to solve complex problems in nonlinear optimization (convex and non-convex), multi-objective programming, optimal control, discrete optimization, and stochastic optimization among others.

In general, recognizing and identifying a problem, and building and solving models to assess and put solutions into action are the standard processes in optimization. In operations research, optimization is a critical tool for making effective decisions in a variety of fields, including in the area of facility location. FLP seek to determine the optimal placement of facilities in order to achieve specific objectives, such as reducing costs or maximizing benefits. These problems involve finding the best location for facilities by considering a range of factors, including transportation costs, availability of resources, and accessibility for customers. FLP is a classical optimization problem that is used to find the optimal solution for the ideal location of a warehouse and factories. The concept of FLP is used to effectively reduce the cost of facility setup and to also cover demand points by the established facilities to ensure the maximum satisfaction of customers.

FLP identifies the ideal location for a distribution center based on demands, costs, and travel distances. The primary goal of any FLP is to strategically locate a set of facilities, each of which serves a set of users so that a particular optimality criterion is satisfied. The spatial decision problem of facility location and its demand allocations has been studied widely with various real-world applications (Liao and Guo, 2008) which include emergency response system management (Ehrgott, 2002), telecommunication network design (Liao and Guo, 2008), power distribution system design (Haghifam and Shahabi, 2002), industrial logistics (Ceselli and Righini, 2005) among others. Obtaining solutions to these facility location problems and their related demand allocations must describe the coverage of each facility and its respective location (Liao and Guo, 2008). Some common studies about FLP include $p-$center problem (Suzuki and Drezner, 1996), the weber problem (Cooper and Katz, 1981), regionally constrained $p-$median problem (Murray and Gerrard, 1997), the capacitated single allocation hub location problem (Ernst and Krishnamoorthy, 1999), the single source capacitated plant location problem (Díaz and Fernández, 2002). As described by Hsu et al. (1995), location models can be grouped into three categories namely the

$p-$median models, $p-$ center models, and covering models. The $p-$center model aims to locate $p-$facilities and minimize the maximum distance between various demand points to the closest facility (Hakimi, 1964). The minimax model minimizes the distance between sited facilities to their farthest client for public facility design and emergency services management like emergency medical services and fire protection (Love et al., 1988). The provision of a specific amount of service coverage in response to demand is one of the most important goals of covering models. Customers are considered to be covered if the distance between the customer and the facility from which the customer receives service falls within a specified effective range of the facility (Wei, 2008). Covering problems also can be described using the smallest enclosing circle problem which aims to identify the circle with the least radius that encompasses all of the other circles in a euclidean plane. General applications of covering models and their specific reviews can be found in (ReVelle et al., 2002). Applications of covering problems are vital in every domain of engineering and the applied sciences. In most modeling situations, facilities and users are represented as points on a plane (ReVelle et al., 2002). Hospitals, schools, supermarkets, rubbish dumps, and chemical plants are examples of facilities that can be set up using covering problem analysis (Barahona and Anbil, 2000). The set of users can either be continuous, meaning that they cover an area in which every point is regarded to be a user, or discrete, meaning that it is made up of a finite number of points (Pereira et al., 2015). The location of the facility is a critical factor in the decisions made about logistics. Every day, a large number of businesses rely on quantitative approaches to determine the optimal or most cost-effective approach to satisfying the needs of their customers in terms of the provision of goods or services. In other instances, the availability of the service may be contingent on the amount of time required to reach an existing facility or the distance traveled. Decision makers will therefore need to identify the most advantageous site to set up facilities in order to fulfill the majority of the demand (Adeleke and Olukanni, 2020). One of the most well-liked FLP is the covering models which continue to hold a lot of appeal for research as a result of their applicability in real-world life, particularly in the context of service and emergency facilities (Farahani et al., 2012). FLP is very vital in planning and setting up emergency medical services to provide coverage to a group of demand points under consideration.

Emergency medical services (EMS), is a network of medical professionals ready to respond to medical emergencies. EMS is primarily concerned with the immediate medical needs of a patient triggered in response to a life-threatening situation. The healthcare industry has always had difficulties in allocating few resources to meet the unending demands of its three primary goals: patient care, service excellence, and financial success. Decision-makers in the healthcare sector are tasked with coming up with efficient means of allocating limited resources in a way that is both equitable and maximizes society's benefits. Within healthcare systems, EMS performs a critical role in stabilizing and transporting critically injured patients to hospitals. Call volume, traffic, infrastructure, and overhead are just some of the variables that might have an impact on emergency medical services. The objective of optimization strategies is to determine the optimal placement of emergency medical facilities and to allocate ambulances to those facilities in order to increase the number of patients who survive their injuries while minimizing the overall cost of the EMS system. Studies that cover topics like the location set covering problem (Toregas et al., 1971), maximal covering location problem (Church and ReVelle, 1974) among others seek to get the optimal locations of facilities that can lead to improved performance and improvement of the EMS system.

The stochastic and dynamic nature of EMS needs to be considered when modeling the EMS system to see its real conditions. The study by Aboueljinane et al. (2013) discusses a review of the various simulation models which are applied to emergency medical service operations. The effectiveness of emergency medical services has been improved by the application of a number of methodologies, including simulation, mathematical programming, and models based on queuing theory. The study by Aboueljinane et al. (2013) primarily focuses on computer simulation models which are used for analysis and improvement of EMS. Simulation is the process of designing and making a computerized model of a system to mimic its operations or properties in order to learn more about how that system works under a certain set of conditions (Aboueljinane et al., 2013). It is one of the most popular operations research methodologies for analyzing complex systems like military deployment, telecommunications systems, logistical networks, manufacturing systems, and healthcare delivery to spot opportunities for improvement and waste. Simulation has been utilized in numerous healthcare-related studies, such as those examining capacity and hospital bed planning (Holm et al., 2013), emergency room critical care situations, patient flow, and wait times (Aboueljinane et al., 2013). Simulation has been proven to be useful in the field of EMS in a variety of contexts due to its capacity to provide an in-depth description of the system and to take into account a variety of different sources of uncertainty. This is in contrast to other approaches, such as mathematical programming and queuing theory, which require simplifying assumptions in order to obtain performance measure predictions. In the work done by Henderson and Mason (2005), they employ simulation to explore potential improvements to New York's emergency ambulance service. Since then, several researchers have developed simulation models to predict how potential future modifications to EMS would affect current operations. McCormack and Coates (2015) also describes a simulation model that allows for the optimization of the allocation of ambulance fleets and the location of base stations in order to enhance the number of patients who survive. It makes use of a genetic algorithm (GA) in conjunction with an integrated EMS simulation model. Within this model, multiple patient classes are described, and survival functions are utilized in order to differentiate the required levels of service (McCormack and Coates, 2015). The aim was to achieve the greatest possible increase in the overall anticipated survival probability across all patient classifications. Models developed from the notion of set covering are used in much of the research that has to be done on EMS facility location (Farahani et al., 2012). These types of models attempt to locate EMS resources in such a way that they can cover a collection of demand nodes. At its most basic level, a node is considered to be covered if an EMS resource is within a certain distance from it or within a certain amount of time. The early covering models neglected the stochastic nature of EMS systems and focused on solving the location problem using a static and deterministic approach (Farahani et al., 2012). Since ambulances function as servers in a queuing system and are occasionally unavailable, probabilistic models based on queuing theory have been created to account for this phenomenon Bianchi and Church (1988). EMS is classified as an application to the hierarchical hub covering facility location problems (Korani and Eydi, 2014). It is the goal of these hub problems to decrease costs and establish an adequate condition in the distribution network by identifying the location of service providers' facilities at various levels and specifying the directions in which they are linked (Korani and Eydi, 2014). Hub location problems (HLP) consider hub placement and assign demand nods to newly constructed hub facilities.

Hubs are specialized facilities used in many-to-many distribution networks for switching, transshipping, and sorting. The HLP entails deciding where to put hubs and how to distribute demand nodes among them so that traffic can be efficiently routed from one origin to a particular destination (Alumur and Kara, 2008). HLP is a relatively new extension of classical facility location problems. Hub facilities consolidate flows rather than serving each origin-destination pair individually in order to capitalize on the advantages of economies of scale. During the hub process, flows that originate from the same source but are headed in separate directions are merged with flows that originate from various sources but will end up in the same place. Consolidation occurs between hubs and along the route from origin to hub and hub to destination. Hub networks can be divided into two broad categories classified as single allocation and multiple allocations. They are different in the way that non-hub nodes are assigned to hubs. Each demand center's incoming and outgoing traffic in a single allocation model is directed to a single hub, while in a multiple allocation model, demand centers may receive and send flow via numerous hubs. The telecommunications industry is one of the earliest adopters of the hub network concept. However, this idea is heavily utilized in logistical systems by the airline and postal industries. The concept of a hub can now be applied to a wide variety of different domains, such as the shipping sector, organizations that specialize in freight transportation, public transportation systems, and message delivery networks (Farahani et al., 2013).

HLP is used to solve diverse problems by different researchers. Notable ones include the hub location and routing problem (Aykin, 1995) where locations of hubs and the services provided on the routes connecting demand points are established together. Rather than calculating the total demand for the services as a whole, each flow from a single origin to a number of different destination points is analyzed independently. Another application that can be discussed is the hierarchical-hub model for airline networks (Chou, 1990). This model is significant because it does not call for an arbitrary number of hub facilities to be built. The ideal number can be derived from within the system. In application to health, HLP can be discussed when the question of how to provide medical care to a large population within a certain radius is examined. An example of such a situation is analyzed in the article which discusses the hub location problem during an epidemic outbreak with emphasis on COVID-19 (Marmolejo-Saucedo and Rojas-Arce, 2020). It is very important to provide access to essentials like food and water as well as medical supplies in a situation where there is an outbreak of an epidemic. It is therefore vital to establish strategic storage and distribution centers in order to guarantee and supply these products. These centers must be able to interact with the projected demand locations in the event of emergencies. For a network to function at its peak efficiency, key parameters including the number and position of distribution and collecting centers, unloading sites, placement of demand centers, and the choice of optimal distribution algorithms, must be defined. Hu and Zhao (2012) offers a multi-objective programming model for the selection of emergency facilities and the amounts of drugs to be transferred from the supply sources to the demand sites (Marmolejo-Saucedo and Rojas-Arce, 2020). They extend the idea of multi-objective programming into a stochastic model by utilizing genetic algorithms for its solution and incorporating the study of system dynamics to describe the dynamic behavior of the refueling, reception, and dispensing sources in the event of an anthrax attack. In their subsequent research, they put forward a dynamic optimization model that includes variable replacement and transport times and relies on heuristic methods to solve it (Liu and Zhao, 2012).

Motivated by existing research in various works of literature, we are interested in proposing a mathematical model to address the following research question. Suppose we are provided a set of demand points and some potential groups of those demand points. How do we identify a hub location and an optimal number of groups to cover all demand points with a minimum radius centering the hub location? If we remove the condition of some potential groups of those demand points, the problem can be reduced to a certain type of p-center problem. Also, if we remove identifying a hub location goal, the problem can be reduced to an excessively studied set covering problem. To our understanding based on the review of several works of literature, no work has been done on simultaneously selecting the best facilities to set up and also locating a communication hub in FLP. This thesis, therefore, aims to fill this research gap by selecting the best facilities to set up and also finding a communication hub that works simultaneously with the selected facilities to supply the health needs of the people. Reduction in the total number of facilities leads to efficient cost minimization. We also consider the minimum radius of the communication hub in other to minimize the total distance between the selected communities (demand points) and the selected facilities. We assume that customers are also grouped to get service from potential facility locations. Thus, the research goal is to find a compact shape that will provide a maximum interior space including potential sites which minimize the maximum Euclidean distance among potential sites and selected sites for satisfying customer demand. The best compact shape is the circular shape. We will propose an extension to existing theories by developing a new mathematical model and algorithm to solve both hypothetical and real-world problems.

The thesis is arranged as follows. In Chapter 2, we discuss the context and current techniques for mathematical optimization. The various principles in linear programming, non-linear programming, and convex optimization are defined, and we supplement our explanations with diagrams showcasing convex and non-convex sets. Moreover, we examine different optimality conditions. We introduced related facility location problems, the Weber problem, and the p-center problem briefly. In Chapter 3 of this thesis, we present an overview of the optimization models and algorithms utilized in our research analysis. We not only explore the formulation proposed by Xu et al. (2003), but also introduce a novel mathematical model that identifies optimal facility locations and minimizes communication hub radii. To assess the effectiveness of our approach, we apply the algorithm to a hypothetical test case and thoroughly analyze the resulting outcomes. In Chapter 4, we expand upon our computational analyses by conducting additional hypothetical test cases and applying our algorithms to a practical, real-world scenario in Dougherty County, Georgia, USA. Specifically, we examine the county's fire stations and health centers and evaluate the efficacy of our algorithms in determining the optimal location for a communication hub with the smallest feasible radius. Our findings shed light on the practical implications of our research and provide valuable insights that can be used by stakeholders and decision-makers in the field. Chapter 5 serves as the culminating chapter of our thesis, offering a comprehensive summary and conclusion of our research findings. In addition, we outline potential areas for future research and expansion of our work. The Appendix complements this discussion by providing a detailed record of the codes utilized in our study, allowing for transparency and reproducibility of our results.

CHAPTER 2

Background And Existing Methods For Mathematical Optimization

In this chapter, we define notations and review basic concepts of mathematical optimization. We also briefly review the various categories of mathematical optimization problems along with the solution approaches. Optimization problems are categorized into groups based on the characteristics of the objective function and constraint functions.

2.1. Basic concepts and optimization models

Mathematical optimization can be defined as the process of minimizing or maximizing an objective function by finding the best available values across a set of inputs. The three major components of an optimization model include the decision variables, objective function, and constraints. Based on the variables in the objective function and constraint function, optimization problems can be classified into continuous and discrete problems. Any general optimization problem can either be minimized or maximized. Without loss of generality, we consider minimization with non-negative decision variables to represent an optimization problem as shown in the model (2.1a - 2.1c).

$$z = \min \; f(\mathbf{x}) \tag{2.1a}$$

$$\text{s.t. } g_i(\mathbf{x}) \leq b_i \text{ for } i = 1, \ldots, m \tag{2.1b}$$

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \geq 0 \tag{2.1c}$$

where $\mathbf{x} = (x_1 \cdots x_n) \in \mathcal{R}^n$ is an $n-$dimensional decision vector, $f : \mathcal{R}^n \rightarrow \mathcal{R}$ is the objective function, $g_i(x)$ represent the constraint function for $i = 1, \ldots, m$, and b_i represents the various limits on the constraint for $i = 1, \ldots, m$.

Optimization problems also can be categorized into convex problems and nonconvex programming problems. Convex problems consist of all linear programming problems and nonlinear optimization problems which are convex while nonconvex programming problems consist of optimization problems that do not meet conditions of convex optimization. In the subsequent discussions, the basic definitions of linear and non-linear optimization, convex and non-convex optimization, some linear models, and the formulation of optimization problems will be explained.

2.2. Convex optimization

Convex optimization can be defined as an area in the field of mathematical optimization that aims to minimize convex functions over convex sets or maximize concave functions over convex sets. Referring to the generic model (2.1a - 2.1c), for convex optimization problems, the objective function f is a convex function, and the constraint functions g_i are also convex functions. The objective function is usually restricted with inequality and equality constraints that show if the given set of optimal solutions achieved should lie within a range or should exactly lie at a point. The convex optimization problems can be further categorized as constrained convex optimization problems and unconstrained convex optimization problems. Unconstrained convex optimization is not subject to any convex constraints. Thus the convex function to be optimized does not have convex constraints in its restrictions as compared to constrained convex optimization problems. Convex optimization problems can be solved using different methods such as the projected gradient methods, interior point methods using convex conjugate gradients and barrier function, modified optimization methods, and sequential convex programming among others. In practical applications, convex optimization can be applied in diverse fields. Examples include the solving of facility location problems which are used to optimize the use of resources within a facility. Scheduling flights to finding various flight times which minimizes costs while maximizing the total number of passengers. Routing phone calls for general network design problems. Logistics for the transportation of commodities from a group of supplies to a group of end users. Inventory management to minimize the total costs of ordering, holding, and shortage and also maintaining stocks within a certain desired point. Some important concepts used in describing convex optimization problems are briefly described below:

Lines and line segments: A line segment is a portion of a straight line bordered by two unique endpoints and containing every point on the line between its endpoints. Let $y_1 \neq y_2$ represent points in $\mathcal{R}^n$ then points of the form $x = \theta y_1 + (1-\theta)y_2$ for $\theta \in \mathcal{R}$ represents the line that passes through y_1 and y_2. If $\theta = 0$ then $x = y_2$ and $\theta = 1$ then $x = y_1$. The (closed) line segment between y_1 and y_2 corresponds to values of the parameter θ that fall within the range of 0 to 1. If we let $x = y_2 + \theta(y_1 - y_2)$ then x is equal to the sum of the base point y_2 (which indicates that $\theta = 0$) and the direction y_1, y_2 (which indicates that y_2 is pointing in the direction of y_1) where the direction is scaled by the parameter θ. Therefore, the value of θ indicates the percentage of the distance between y_2 and y_1 at which point x can be found. The point x moves from y_2 to y_1 as θ increases from 0 to 1; for θ values that are larger than 1, the point x is located on the line beyond y_1 in the equation.

Convex sets: For convex sets, the subset contains the entire line segment that connects any two points in it. It is a set that intersects every line into a single line segment. The term "convex hull of A" refers to the intersection of all of the convex sets in euclidean space that contain a particular subset denoted by the letter "A". A set Q is said to be convex if the line segment that connects any two points in the set Q lies somewhere within Q. Thus for any $a_1, a_2 \in Q$ and any θ where $0 \leq \theta \leq 1$ we will have $\theta a_1 + (1-\theta)a_2 \in Q$. Each and every affine set is also convex due to the fact that it includes the complete line that runs between any two different points that are included in it, as well as the line segment that runs between those points. In Figure (2.1- 2.2) below, we show some examples of convex and non-convex sets:

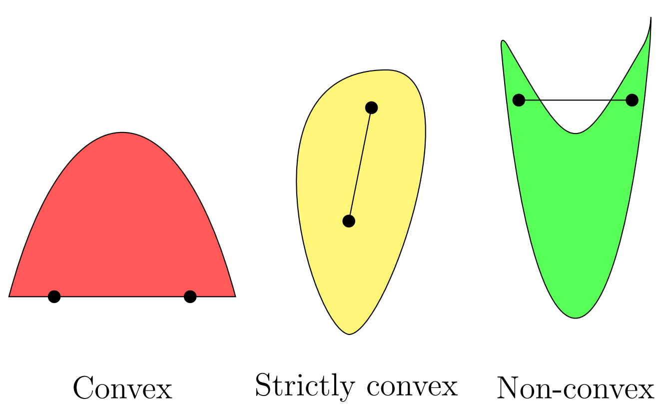


FIGURE 2.1 Convex, strictly convex and non-convex sets

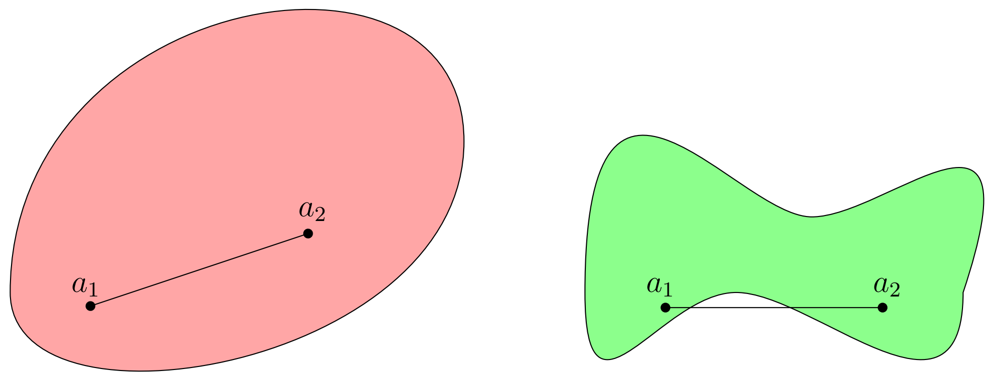


FIGURE 2.2 Convex and non-convex sets

Convex functions: If the domain of a function is convex and satisfies the condition 2.2

$$h(\theta x + (1-\theta)y) \leq \theta h(x) + (1-\theta)h(y) \tag{2.2}$$

where $x, y \in \mathbf{dom}(h)$ and $\theta \in [0,1]$ then the function can be classified as a convex function. The line segment that geometrically connects $(x, h(x))$ to $(y, h(y))$ must be above the graph of h. The function becomes strictly convex if the inequality is strict. Examples of convex functions include quadratic form, least squares, quadratic-over-linear, and geometric mean among others. Some convex functions in $\mathcal{R}$ and $\mathcal{R}^n$ includes exponential: $e^{\beta x}, \forall \beta \in \mathcal{R}$, powers of absolute value: $|y|^q$, $\forall\ q \geq 1$, and affine: $b^T x + m, \forall\ b \in \mathcal{R}^n$, $m \in \mathcal{R}$.

2.3. Optimality

An optimality criterion is a set of requirements that a function's minimum value must satisfy. Optimization techniques that aim to satisfy the optimality conditions by some means (perhaps numerically) are commonly known as optimality criterion techniques (Arora, 2004). To analyze the optimality condition for a convex optimization problem, we consider the optimization problem below (2.3a-2.3b)

$$z = \min \; f(\mathbf{x}) \tag{2.3a}$$

$$\text{s.t. } x \in \Omega \tag{2.3b}$$

where the function $f : R^n \longrightarrow R$ is convex and differentiable, and Ω is convex. We consider a point x as optimal if and only if $x \in \Omega$ and satisfies the condition below 2.4

$$\nabla f(x)^T(y - x) \geq 0, \; \forall \; y \in \Omega \tag{2.4}$$

Thus, f is increased locally whenever we move towards any feasible y from x.

First-order conditions: Consider the gradient function h stated below 2.5

$$\nabla h(x) = \left(\frac{\partial h}{\partial x_1}, \frac{\partial h}{\partial x_2}, \frac{\partial h}{\partial x_3}, \cdots, \frac{\partial h}{\partial x_n} \right) \tag{2.5}$$

If the function exists for all $x \in$ **dom** h and the domain is open then the function h is differentiable. From the definition above, assume that h is differentiable, thus its gradient can be found at some point in **dom** h which is open, then h is convex if and only if **dom** h is convex and $h(x) \geq f(y) + \nabla h(y)^T(x - y)$ holds for all $x, y \in$ **dom** h.

Second-order conditions: Consider the function f given in 2.6. If the domain of the function is open and the hessian exists $\forall \; x \in$ **dom**f then the function is twice differentiable.

$$\nabla^2 f(x) \in S^n, \nabla^2 f(x)_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j} \; i, j = 1, \cdots, n. \tag{2.6}$$

From the above definition, the second-order condition for convexity can be explained as follows. Suppose that f is twice differentiable thus its second derivative ($\nabla^2 f$) can be found at each point in **dom** f, which is open, then the function f is convex if and only if the domain function f is convex and the hessian of the function f is positive semidefinite.

2.4. Linear programming

Linear programming is a subset of convex optimization, also known as linear optimization, is a technique for optimizing a mathematical model provided in (2.1a - 2.1c) which obey a linear relationship for the objective function and constraint functions. In general, a linear programming problem can be written as (2.7a- 2.7e):

$$\min \ f(x) = c_1x_1 + c_2x_2 + c_3x_3 + \cdots + c_nx_n \tag{2.7a}$$

$$\text{s.t.} \quad a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n \leq b_1 \tag{2.7b}$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n \leq b_2 \tag{2.7c}$$

$$\vdots \qquad \vdots \qquad \vdots \quad + \cdots + \quad \vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \cdots + a_{mn}x_n \leq b_m \tag{2.7d}$$

$$x_1, x_2, x_3, \ldots x_n \ \geq 0 \tag{2.7e}$$

where $f(x) = c_1x_1 + c_2x_2 + c_3x_3 + \cdots + c_nx_n$ represents the objective function to minimized.The coefficients of the objective function are denoted by $c_1, c_2, c_3, \cdots, c_n$, while the restriction on the objective function also known as the constraint is stated as $\sum_{k=1}^{n} a_{jk}x_k \leq bj$ where a_{jk} represent the coefficient of the various constraints for $j = 1, \cdots p$ and $k = 1, \cdots n$. The nonnegativity constraint is represented as $x_1, x_2, x_3, x_n \geq 0$.

Linear programming is useful in operations research for modeling, formulating, and solving real-world application problems with linear objective functions and linear constraints together with continuous decision variables. Transportation, energy, telecommunications, and manufacturing are examples of disciplines where linear programming can be implemented. It has been effective for simulating a variety of planning, routing, assignment, and design problems. The problems can be easily changed from one equivalent form to another (minimization to maximization and vice versa) with some basic modifications. In the practical application of linear programming, most problems represent physical quantities and are usually non-negative. The Simplex method and its variants, such as the dual-simplex method, are successfully used in solving linear programs. This method is used through the introduction of slack or surplus variables, pivot variables, and tableaus as a way of finding the optimal solution to optimization problems. Having the problem in standard form is essential because it provides a solid foundation for addressing optimization problems using the simplex method and other approaches. While the exact number of arithmetic operations used in solving linear programming problems cannot be determined, rigorous bounds can be established on the number of operations required to solve a given problem to a particular accuracy using the interior point method. The two most basic forms of linear program representations are the standard form and the canonical form. A linear programming problem is said to be in standard form if all variables are nonnegative and all constraints are equality constraints. The simplex method is applied in solving problems in their standard form. Any linear program can be transformed to this form using slack or surplus variables. The **standard** form of a linear program in the matrix form can be represented as in (2.8a - 2.8c).

$$\min f(x) = c^T x \quad (2.8a)$$

$$\text{s.t. } Ax = b \quad (2.8b)$$

$$x \geq 0 \quad (2.8c)$$

where $c^T = (c_1, c_2, \ldots, c_n) \in \mathcal{R}^n$ represents the transpose of the objective coefficients, $A = [a_{ki}]$ denotes the $m \times n$ matrix in $\mathcal{R}^{m \times n}$ whose element in the i^{th} row and k^{th} column is denoted by a_{ki}, $b^T = (b_1, b_2, \ldots, b_m) \in \mathcal{R}^m$ and $x^T = (x_1, x_2, \ldots, x_n) \in \mathcal{R}^n$.

2.5. Nonlinear programming

Nonlinear programming (NLP) refers to the process of finding solutions to an optimization problem with some of the objective functions or constraints being nonlinear. In this type of optimization problem, the feasible region is determined by the nonlinear constraints. The standard form for a nonlinear optimization problem with both equality and inequality constraints is given below (2.9a - 2.9d):

$$z = \min\ f(\mathbf{x}) \quad (2.9a)$$

$$\text{s.t. } g_i(\mathbf{x}) = 0 \text{ for } i = 1, \ldots, m \quad (2.9b)$$

$$h_j(\mathbf{x}) \leq 0 \text{ for } j = 1, \ldots, n \quad (2.9c)$$

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \geq 0 \quad (2.9d)$$

where x is a vector of decision variables, $f(x)$ is the objective function to be minimized, $g(x)$ is a set of equality constraints that must be satisfied, and $h(x)$ is a set of inequality constraints that must be satisfied. In this type of problem, either the objective function or the constraints are nonlinear.

NLP algorithms can be categorized into two main groups: gradient-based methods and derivative-free methods. Gradient-based methods rely on computing the gradient of the objective function, which is the vector of its partial derivatives with respect to the decision variables. Derivative-free methods, on the other hand, do not rely on gradient information and can be useful when the gradient is difficult to compute. NLP has many real-world applications, such as in finance, where it can be used to optimize investment portfolios, or in engineering, where it can be used to optimize the design of complex systems such as aircraft or automobiles. NLP can also be used in machine learning, where it can be used to train models with nonlinear activation functions.

2.6. Least-squares problems

In regression analysis, the method of least squares is a commonly used methodology for approximating the solution of overdetermined systems by the minimization of the sum of squares of residuals produced by the solution of each equation. Least-squares problems are defined as optimization problems whose objective function is represented by the sum of squares but without a constraint. An example of a least square problem is stated in 2.10 below:

$$\min \; f_0(x) = ||Ax - b||_2^2 = \sum_{i=1}^{k} (a_i^T x - b_i)^2 \tag{2.10}$$

where the total number of rows for A is represented by $k \geq n$ and a_i^T for $M \in \mathcal{R}^{k \times n}$. The vector $x \in \mathcal{R}^n$ denotes the optimization variable.

It is trivial to recognize an optimization problem as a least-squares problem, provided that the objective function is quadratic and that the corresponding quadratic form is positive semi-definite. Ordinary least squares and nonlinear least squares are the two classes within which least square problems can be grouped. A problem can be classified as ordinary least squares or nonlinear least squares based on the linearity of the residuals in all unknowns. Iterative refinement is a common method for solving a nonlinear problem, as the system is transformed into a linear approximation at each step. The linear least square can often be seen in statistical regression analysis and can be used to fit data in a model if the idealized value supplied by the model for any data point can be stated linearly in terms of the unknown parameters of the model. In solving least squares problems, the solution can be reduced to a set of linear equations in the form

$$(A^T A)x = A^T b.$$

Analytically, the solution for this system is simply written as

$$x = (A^T A)^{-1} A^T b.$$

We have good techniques using software implementations for solving least-squares problems with very high accuracy and reliability. Algorithms and software for solving least-squares problems are reliable enough for embedded optimization. Existing methods for solving optimization least-squares problems are very effective and extremely reliable.

2.7. Global and local optimization

Local optimization refers to algorithms that are used to find the local optima. Thus, it locates the best possible optimal solution for a particular region of the search space. The main idea behind local optimization is to find a point that is locally optimal or get a point that optimizes the objective function among feasible points that are near it but not assured of having a lower objective value than all other feasible points. One disadvantage to the local optimization method is that it requires an initial guess for the optimization variable. This initial guess can greatly affect the objective value of the local solution obtained since it is very critical. This method involves experimenting with the choice of algorithm, adjusting algorithm parameters, and finding a good enough initial guess when solving problems. Algorithms that perform local searches iteratively make small modifications to each solution in the search space until an optimal solution is identified or a certain amount of time has elapsed. In machine learning and deep learning applications, local optimization is employed to find solutions to complex problems. Using local optimization can boost the functionality of a design generated using manual or other design processes in an engineering design program.

The goal of global optimization is to identify the optimal solution for a set of (usually nonlinear) models when several optimal solutions exist locally. The solution to nonlinear models in many applied areas such as data analysis, financial planning, environmental management, risk management, and scientific modeling among others usually requires a global search approach.

2.8. Related facility location problems-location science

Different kinds of FLP have been studied extensively in different articles Drezner et al. (2002). The earliest one to be studied is the weber problem which aims to locate a facility in the euclidean plane in order to minimize the sum of its (weighted) distances to the locations of a given set of demand points $X = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ on the plane. The $p-$center problem is also another form of FLP that aims to find the minimum coverage distance such that all demands are covered. It is also called the minimax problem since it aims at minimizing the greatest possible distance between demand and the nearest facility. In effect, the $p-$center problem is minimizing the maximum distance from any customer to the nearest facility.

2.9. Weber location problem

The problem stated below is well-known in the continuous location theory. It is called the Weber problem which is defined as (2.11a-2.11b)

$$\min_{\bar{x}} \phi(\bar{x}) = \sum_{j=1}^{n} w_j ||\bar{x} - \bar{x}_j|| \tag{2.11a}$$

$$\text{s.t. } \bar{x} = (x, y) \text{ and } \bar{x}_j = (x_j, y_j) \tag{2.11b}$$

where the parameters w_j represents the various weights that are known, $\bar{x}_j$ represents points in E^2, and $\bar{x}$ is to be determined (Cooper and Katz, 1981).

The problem stated above (2.11a-2.11b) is an unconstrained optimization where $\phi(\bar{x})$ is a convex function. Newton-Raphson method and the weiszfeld algorithm are some approaches that have been used to find a solution to the weber problem (Cooper and Katz, 1981). Some of these approaches are found to be more effective than others.

2.10. $P-$center problem:

The main objective of the p-center problem is to arrange a certain number of facilities (say p facilities) on a network in such a way as to reduce the greatest distance between a demand point and the facility that is geographically closest to it. A p-center can therefore be simply referred to as an approach that seeks to find a minimax solution that consists of a collection of p points that minimize the greatest possible distance between a demand point and the point in the collection that is closest to the demand point. The p-center problem in an area generalizes the euclidean p-center problem by requiring that the minimum-radius circles centered at the p depot sites enclose the entire area, rather than only a finite collection of demand points. It's far more challenging than the discrete counterpart. The application of this problem is most commonly used when deciding where to place emergency services like ambulances, fire stations, and police stations.

The problem is referred to as the "vertex restricted p- center problem" when the placement of the facilities is limited to the vertices of the network, however, if the facility's placement can be located anywhere on the network, we refer to such instances as the "absolute p- center problem". In some instances, it can be called a "capacitated p- center problem". This is when on the facilities, there are capacity restrictions where the demand nodes can be distributed to the facilities using a single allocation strategy or multiple allocations Calık (2013).

Given a finite set of demand points, the $p-$ center problem is to find p depots to minimize the maximum distance from any demand point to its respective nearest depot. Specifically, given a set $X = \{(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)\}$ of demand points on the plane, the problem is to find p points with coordinates $\{(s_1, t_1), (s_2, t_2), \ldots, (s_p, t_p)\}$ to minimize $\max_{1 \leq i \leq n} \min_{1 \leq j \leq p} \{(a_i - s_j)^2 + (b_i - t_j)^2\}$.

The absolute center problem was first proposed by Hakimi (1964) for the purpose of determining where a police station or a hospital should be situated so as to minimize the maximum distance between the facility and a group of communities that are linked together by a network of highways. The literature work of Hakimi (1964) described an analysis as follows:

Assume $G = (Q, E)$ be a graph which has the node set $Q = \{q_1, \cdots, q_n\}$ and f_j is used to represent the weight of the node $q_j \in Q$. Also, assume that l_{ij} is used to represent the length for the edge $\{i, j\} \in E$ that connects nodes q_i and q_j.

The goal of the absolute center problem is to locate a point x on the nodes such that $\max\limits_{j=1,\cdots,n} f_j d(q_j, x)$ is minimized, where $d(q_j, x)$ represent the length of the shortest distance between node q_j and point x.

When the problem is solved, then the optimal value that gives the results in the best possible way is referred to as the absolute radius of graph G. If we restrict x to the nodes of G, we will locate the center of the graph G, and the value that gives us the best results will be the so-called radius of G. There is no guarantee that there is a unique "center of G." In other words, the radius is not necessarily equal to the absolute radius. The generalization of the absolute center problem to the p-center problem is something that Hakimi (1965) discusses in the conclusion of his subsequent research on median and covering problems.

If a set of m points in G known as $X_m = \{x_1, \cdots, x_m\}$ is provided, then the distance $d(X_m, q_j)$ between X_m and node q_j can be determined using the formula

$$\min_{i=1,\cdots,m} d(x_i, q_j).$$

The objective of the p-center issue is to locate a set X_m of m points in G such that the maximum value of

$$\max_{j=1,\cdots,n} f_j d(q_j, X_m)$$

is minimized.

2.10.1. Approaches to analyzing the $P-$center problem: In this section, the various approaches or variants which are used in analyzing the p-center problem are discussed.

Firstly, the absolute and vertex $p-$center problem will be discussed. Capacitated $p-$center Problem as well as the discrete approach, will also be analyzed.

The absolute $P-$center problem : In this problem, the aim is to locate a set $Y' \subset G$ which has p number of points so that $f(Y') \leq f(Y)$ where $Y \subset G$ for any p number of points. To get the optimal value of the absolute p-center problem, the condition $r'_A = f(Y') = \min_{Y \subset G:|Y|=p} f(Y)$ must hold. Hakimi (1964) first developed and defined the absolute $p-$center problem by using graphical methods and also using the idea of the piecewise linearity of a function. A point on the edge (y), is considered an intersection point in this problem if there exist two distinct vertices such that $d(i, y) = d(y, j)$. Note that i, j are considered as the two distinct vertices and the point y on the edge $\{t, h\} \in E$. To find $d_{iy} = d_{yj}$, the condition $d_{it} + d_{ty} = d_{yh} + d_{hj}$ or $d_{jt} + d_{ty} = d_{yh} + d_{hi}$ must hold where $d(i, y)$ becomes the relative radius of y. According to Minieka (1970), Y' is an optimal solution of the absolute $p-$center problem where $Y' \subset (P \cup N)$. There are no more than $O(n^2)$ intersecting points on any one particular edge, and there are $O(n^2|E|)$ points over the entire network. Given this, it is possible for us to suppose that the potential facilities at the absolute p-center come from a finite collection (Calık, 2013). A more in-depth examination than just enumerating the vertex set is required in order to solve the absolute center problem, which necessitates an endless search through a continuous set of points on edges.

Vertex restricted $P-$center problem : The vertex-restricted $p-$center problem was first proposed by Daskin (1997), however, he opts, instead to do a set-covering-based bisection search within a region bounded by lower and upper bounds on r_v^* that has already been computed. This approach is enhanced by Daskin (2000), who solves a maximal covering location problem using lagrangean relaxation, where the goal is to maximize the number of covered vertices inside a region of radius r, with a maximum of p open facilities allowed. In this problem, the idea is to locate a set $Y' \subseteq N$ which has p number of vertices so that $f(Y') \leq f(Y)$ where $Y \subseteq N$ for any p number of vertices. In an approach to find the optimal value of the vertex-restricted $p-$ center problem, the condition $r'_V = f(Y') = \min_{Y \subset Y:|N|=p} f(Y)$ must hold.

Capacitated P-center problem : Barilan et al. (1993) carried out the very first research investigation on the capacitated $p-$center problem. They investigate the p-center problem, which is the situation in which a facility can satisfy at most L demand nodes. As a subset of the problem studied about the $p-$center, they coin the term "balanced $p-$ center problem" to describe situations in which every node requires exactly one unit but every facility can produce a maximum of L units. A simplified version of this capacity problem was also considered by Barilan et al. (1993) in which more than one center can be placed and located on a node and this was referred to as a multi p-center problem. Albareda-Sambola et al. (2010) also provides a solution approach to the capacitated $p-$ center problem. They use two auxiliary problems known as the maximum demand coverage within a defined radius and minimal necessary centers within a fixed radius to derive lower bounds from the lagrangean duals. By employing a binary search approach, their precise algorithm not only resolves the second auxiliary problem but also chooses the radius value that should be used to resolve this problem from the set of possible radius values. The lower and higher boundaries that they obtain place restrictions on the range of possible radius values. Let B_k represent the capacity node for $k \in K$ and c_i be the demand node where $i \in I$. Then to get the effective capacity Aardal et al. (1995) of the node $k \in K$ of the radius r, we use the condition

$$B_k^r = \min\left\{B_k, \sum_{k\in I_r(i)} c_k\right\}$$

Discrete center problem: Locating one or more facilities on a network to service a set of demand points at known locations is the focus of this class of problems. The goal is to ensure that all demand points are serviced by the facility that are geographically closest to them, while also minimizing the maximum distance between any two demand points and the facilities servicing them. While the title suggests a finite number of demand points, there are also continuous variants of center location problems that arise when the collection of demand points to be served represents a continuous number of points on the network. Hakimi (1964) first investigated the discrete center problem. This is a foundational publication in the sense that it paved the way for an entirely new field of research that we now know as network location as a result of its influence. In his research article, Hakimi (1964) discusses two issues that he refers to as the absolute median problem and the absolute center problem. Both of these issues assume that the weights are positive and that there are no addends. Both problems require a continuous set of points to represent the edges of a network. Minimizing the greatest possible distance between the facility and any vertex is the goal of the absolute center problem, whereas minimizing the weighted total of such distances is the goal of the absolute median problem.

CHAPTER 3

OPTIMIZATION MODELS AND ALGORITHMS

In this section, we develop an optimization framework for identifying the best location of a communication hub and circles such that all demand points are covered by at least one circle such that the radius of the largest circle containing all selected circles is minimized.

3.1. Optimization models

The problem consists of m fixed demand points with the index set $I = \{1, 2, \ldots, m\}$ that should be covered by a collection of subsets of n potential facilities (hospitals) with the index set $J = \{1, 2, \ldots, n\}$. Assume that these subsets correspond with circles with given the centers $(a_j, b_j) \in \mathcal{R}^2$ and radii r_j (maximum distance from the center) for $j \in J$ on the plane. The objective of the optimization problem is to identify the optimal coordinates $(x, y) \in \mathcal{R}^2$ for a new landmark and a subset of circles, such that all m demand points are encompassed by at least one circle. The goal is to minimize the radius of the largest circle that encloses all selected circles while meeting the coverage requirements. In addition to decision variables (x, y, R), the decision variable $x_j,\ j \in J$ is defined to indicate which polygon j is selected to cover demand point $i \in I$.

$$x_j = \begin{cases} 1 & \text{if the circle with the center } (a_j, b_j) \text{ is selected} \\ 0 & \text{otherwise.} \end{cases}$$

The problem can be formulated into a mixed integer nonlinear programming model and the required distance R can be formulated by the optimization model provided by (3.1a-3.1d).

$$MA(I,J): \min R \tag{3.1a}$$

$$\text{s.t.} \sum_{i \in J_i} x_j \geq 1, \text{ for all } i \in I \tag{3.1b}$$

$$\sqrt{(a_j - x)^2 + (b_j - y)^2} + r_j \leq R + (1 - x_j)M, \text{ for all } j \in J \tag{3.1c}$$

$$x \geq 0, y \geq 0, x_j \in \{0,1\}, \text{ for all } j \in J \tag{3.1d}$$

The objective function (3.1a) minimizes the radius of the largest circle that contains all selected polygons. Constraint (3.1b) serves as the coverage constraint. It allows respective demand points to be covered by a selected circle and Constraint (3.1c) is used to minimize the maximum distance from any demand point to its respective nearest polygon. Now, considering (3.1c), if $x_j = 1$ we obtain $\sqrt{(a_j - x)^2 + (b_j - y)^2} + r_j \leq R$. This condition ensures that all selected circles are included in the largest circle with the radius R. If $x_j = 0$, we just obtain $r \geq 0$. Constraint (3.1d) provides the domains of decision variables x, y, x_j for $j \in J$, respectively.

We outline three algorithms to obtain a bound on R, respectively. We divide the constraints (3.1c) into two sets of constraints. Assume that we partition the set of polygons into two groups such that for all $j \in \bar{J},\ x_j = 1$ and for all $j \in J \setminus \bar{J},\ x_j = 0$. If we set $x_j = 1$ then obtain $\sqrt{(a_j - x)^2 + (b_j - y)^2} + r_j \leq R$. With this binary partition, we obtain two sets of convex constraints and the following theoretical result.

THEOREM 3.1. *Let $x = (x_1, x_2, \ldots, x_n)$ be set to any binary assignment $\{0,1\}^n$. Then the MINLP defined above is reduced to a convex optimization problem*

PROOF. Let $\bar{J} = \{j \in J : x_j = 1\}$. The optimization model (3.1a-3.1d) can be represented as

$$f = \min R \tag{3.2a}$$

$$\text{s.t.} \sqrt{(a_j - x)^2 + (b_j - y)^2} + r_j \leq R, \text{ for all } j \in \bar{J} \tag{3.2b}$$

$$\sqrt{(a_j - x)^2 + (b_j - y)^2} + r_j \leq R + M, \text{ for all } j \in J \setminus \bar{J} \tag{3.2c}$$

$$x \geq 0, y \geq 0 \tag{3.2d}$$

The right side of constraint (3.2b) and (3.2c) are convex functions of the variables $(x, y) \in \mathcal{R}^2$ because each term $\sqrt{(a_j - x)^2 + (b_j - y)^2} = ||(x, y) - (a_j, b_j)||_2$ is convex due to l_2 norm. This yield the terms $\sqrt{(a_j - x)^2 + (b_j - y)^2} + r_j$ and $\sqrt{(a_j - x)^2 + (b_j - y)^2} + r_j$ which are convex for $j \in \bar{J}$ and $j \in J \setminus \bar{J}$. The epigraphs of constraint (3.2b) are given by the sets $\{(x_j, y_j, R) : ||(x, y) - (a_j, b_j)||_2 + r_j \leq R\}$ for $j \in J \subseteq \mathcal{R}^3$. Similarly, the epigraphs of constraint (3.2c) are given by the sets $\{(x_j, y_j, R) : ||(x, y) - (a_j, b_j)||_2 + r_j - M \leq R\}$ for $j \in J \subseteq \mathcal{R}^3$. Therefore, the inequalities given in constraint (3.2b) and (3.2c) describe the epigraph of the convex function for $j \in \bar{J}$ and $j \in J \setminus \bar{J}$ which are convex Boyd et al. (2004). The intersection of convex sets forms a convex set. Hence the feasible region of the relaxed problem forms a convex set on $\mathcal{R}^3$. We obtain a convex optimization problem by minimizing a linear function R over the intersection of n convex sets. This completes the proof. $\square$

3.2. Computing an upper bound on R: SCP-allocate algorithm

We determine an assignment $\bar{J} = \{j \in J : x_j = 1\}$ satisfying the constraint $\sum_{i \in J_i} x_j \geq 1$, for all $i \in I$. Such an assignment can be obtained by solving the classical SCP which is an optimization/search version of the set cover is NP-hard. Thus, our first algorithm is constructed based on solving the classical SCP. We first solve the classical SCP using the model (3.3a-3.3c)

$$f_{scp} = \min \sum_{j \in J} x_j \tag{3.3a}$$

$$\text{s.t.} \sum_{i \in J_i} x_j \geq 1, \text{ for all } i \in I \tag{3.3b}$$

$$x_j \in \{0,1\}, \text{ for all } j \in J \tag{3.3c}$$

We may solve the LP-relaxed version of the SCP in which the integrity gap of this relaxation is at most $\log m$. Hence, its relaxation gives a factor-$\log m$ approximation algorithm for the minimum set cover problem when solving the exact version is computationally challenging. Now we propose the iterative SCP-allocate algorithm for approximately solving the MINLP.

Let x^* be the optimal solution of the SCP. Let $SE(x^*) = \{j \in J : x_j^* = 1\}$ and $NS(x^*) = \{j \in J : x^* = 0\}$. Then the set $\{\bar{J}, J \setminus \bar{J}\}$ partition the index set J. Let $\bar{J} \subseteq J$ denote the index set of the selected set to optimal solution x^*. With this binary assignment $\{0,1\}^n$, we then define the model (3.4a-3.4c) using indices in $\bar{J}$.

$$f_R = \min R \tag{3.4a}$$

$$\text{s.t.} \left(\sqrt{(a_j - x)^2 + (b_j - y)^2} + r_j \right) \leq R, \text{ for all } j \in \bar{J} \tag{3.4b}$$

$$x \geq 0, y \geq 0 \tag{3.4c}$$

The model is nonlinear model and a quadratic programming approach can be used to find the upper bound R. This model can be solved globally using the solution approaches described in (Xu et al., 2003). We review the complete solution approach proposed by (Xu et al., 2003) below.

Formulation 1: First, we reformulate $\sqrt{(a_j - x)^2 + (b_j - y)^2} + r_j \leq R$, for all $j \in \bar{J}$ by the introduction of new auxiliary variables $x_j = a_j - x, y_j = b_j - y$, and $z_j = R - r_j, j \in \bar{J}$ the model (3.4a-3.4c) is reformulated as (3.5b - 3.5e):

$$\min R \tag{3.5a}$$

$$\text{s.t. } x_j + x = a_j,\ j \in \bar{J} \tag{3.5b}$$

$$y_j + y = b_j,\ j \in \bar{J} \tag{3.5c}$$

$$z_j + r_j = R,\ j \in \bar{J} \tag{3.5d}$$

$$z_j \geq \sqrt{x_j^2 + y_j^2},\ j \in \bar{J} \tag{3.5e}$$

This problem can be solved using any global nonlinear solver such as Ipopt.

Formulation 2: Second, we use iterative means to solve a series of convex quadratic programming problems to enable us to find the smallest enclosing circle. In order to accomplish this, we will first construct a new variable θ and consider a fixed $R \geq \max\{r_j,\ j \in \bar{J}\}$ for the problem can be written as in model (3.6a-3.6b).

$$\min \theta \tag{3.6a}$$

$$\text{s.t. } (x - a_j)^2 + (y - b_j)^2 - \theta \leq (R - r_j)^2, \ j \in \bar{J} \tag{3.6b}$$

Let $z = x^2 + y^2 - \theta$. We formulate model (3.6a-3.6b) as in model (3.7a-3.7b).

$$\min x^2 + y^2 - z \tag{3.7a}$$

$$\text{s.t } -2a_j x - 2b_j y + z \leq (R - r_j)^2 - a_j^2 - b_j^2, \ j \in \bar{J}. \tag{3.7b}$$

The model (3.7a-3.7b) is a linearly constrained quadratic programming problem if R is fixed and sufficiently large. We solve this model using the heuristic approach as described below (steps I-III). Let ϵ be a very small number.

Step I: start with $(x, y) = (0, 0)$ and then compute the value of R using

$$R = \max_{j \in \bar{J}} \left\{ \sqrt{(a_j - x)^2 + (b_j - y)^2} + r_j \right\}$$

Step II: solve the model (3.7a-3.7b) to find the values of x, y and z

Step III: if $|x^2 + y^2 - z| \leq \epsilon$, then stop, otherwise we will continue the computation for $R = \max_{j \in \bar{J}} \left\{ \sqrt{(a_j - x)^2 + (b_j - y)^2} + r_j \right\}$ with new (x, y) and then proceed to **Step II**.

We summarize our solution approaches in the following pseudocodes. We provide three algorithms. SCP-allocate Algorithm 1 solves the model with Formulation 1, while SCP-allocate Algorithm 2 solves the model with Formulation 2. The codes for these algorithms are provided in Appendix 5.2. The implemented algorithms using Julia and Python programming languages.

Algorithm 1 SCP-allocate Algorithm 1

1: **Input:** an instance of $MA(I, J)$
2: **Step I: Initialization** Solve SCP and identify initial potential locations $\bar{J} = \{j \in J : x_j^* = 1\} \subseteq J$
3: **Step II: Obtain Upper Bound**
4: Apply Formulation 1
5: Return: (R, x, y)

The pseudocode in Algorithm 1 above explains the process involved in identifying the best location for the communication hub and which facilities are to be opened in order to minimize the radius of the enclosing circle but maximize the needs of the people. It also shows the direct quadratic approach also known as the all-in-one (AIO) algorithm used to obtain the center and minimum radius of the communication hub. Line 1 gives reference to the model (3.3a-3.3c) applied in solving the test case. In Line 2, we input the coverage matrix together with the objective function and constraint functions in the code. We then apply the Mosek optimizer to solve the SCP. SCP can be solved with a linear solver which ensures the selection of the best facilities $j \in J$ to set up that covers all demand points $i \in I$. In other to obtain the upper bound as shown in Line 3, we applied Formulation 1 in Line 4 to get values for the center and minimum radius of the communication hub. Note that any global nonlinear solver can be used in solving the problem. In this work, we used the "Interior Point Optimizer" (Ipopt). Ipopt is a software package that is available under an open-source license and is designed for large-scale nonlinear optimization. The goal of the Ipopt algorithm is to discover a local solution to nonlinear problems using a filtering mechanism that searches along interior point lines. Line 5 returns the values of x, y and R

Algorithm 2 SCP-allocate Algorithm 2

1: **Input:** an instance of $MA(I, J)$, $\epsilon = 0.001$
2: **Step I: Initialization** Solve SCP and identify initial potential locations $\bar{J} = \{j \in J : x_j^* = 1\} \subseteq J$
3: **Step II: Obtain Upper Bound**
4: Apply Formulation 2
5: Set $(x, y) = (0, 0)$ and compute $R = \max_{j \in \bar{J}} \left\{ \sqrt{(a_j - x)^2 + (b_j - y)^2} + r_j \right\}$
6: **while** $|x^2 + y^2 - z| \leq \epsilon$ **do**
7: Apply $QP(\bar{J}, R)$ procedure to find new (x, y, z)
8: Compute $R = \max_{j \in \bar{J}} \left\{ \sqrt{(a_j - x)^2 + (b_j - y)^2} + r_j \right\}$ using new (x, y)
9: **end while**
10: Return: (R, x, y)

The pseudocode in Algorithm 2 explains the iterative approach to finding the center and minimum radius of the communication hub as proposed by Xu et al. (2003). Lines 1 and 2 solve the classical SCP to enable the selection of the best facilities to be opened for the maximum satisfaction of customers as explained in Algorithm 1. Line 3 obtains the upper bound by solving a series of convex quadratic programming problems using iterative means as described in Formulation 2 in Line 4. Lines 5 to 10 further explain the idea behind formulation 2. In Line 5, we set the initial values of (x, y) as $(0, 0)$. We input the center and radius of the selected facilities to be opened in our code together with the objective function and constraints. We find the value of R by selecting the maximum distance between the selected facilities and the initial (x, y). Note that on each iteration for finding the distance, we add the corresponding radius of the circle before we select the maximum value as R. We then solve the model (3.7a-3.7b) to find the optimal values of x, y and z. In Line 6, we test a condition with respect to ϵ where ϵ is taken as 0.001. If the condition is met, the iteration stops. If the condition is not met as shown in Lines 7 and 8, we run Formulation 2 again using the new values of (x, y) obtained from the previous iteration. This continues until the condition in Line 6 is met to end the iteration as shown in Line 9. Line 10 returns the optimal values of R, x, y. where (x, y) and R show the center and minimum radius of the communication hub.

Algorithm 3 Dynamic Set Selection

1: **Input:** an instance of $MA(I, J)$
2: **Step I: Initialization** Find initial x_{j^*} with largest coverage. Let $\bar{J} = \{j^*\}$
3: **Step II: Obtain Upper Bound**
4: Set $count$ = covered demand points by x_{j^*}
5: Set $(x, y) = (a_{j^*}, b_{j^*})$
6: **while** $count \neq m$ **do**
7: find next best $j^* \in J \setminus \bar{J}$ that covers the highest uncovered demand points
8: if only one set is eligible update $\bar{J} = \bar{J} \cup j^*$
9: if more than one set is eligible
10: apply $QP(\bar{J}, R)$ procedure to find new (x, y, z) for each candidate
11: compute $R = \max_{j \in \bar{J}} \left\{ \sqrt{(a_j - x)^2 + (b_j - y)^2} + r_j \right\}$ using new (x, y) for each candidate
12: select the set that produces small R and update $\bar{J} = \bar{J} \cup j^*$
13: update covered demand points
14: update $count = count$+covered demand points by new x_{j^*}
15: **end while**
16: Return: (R, x, y)

The newly developed algorithm is displayed in Algorithm 3 above. From the algorithm, Lines 1 and 2 explain the model that solves the classical SCP as explained in Algorithm 1. It locates the best facilities to be opened and allows respective demand points to be covered by the selected circles. This algorithm uses Ipopt to solve for the minimum radius which represents the upper bound in Line 3. Lines 4 to 16 explain the iterative approach used in selecting facility $j \in J$ that covers demand point $i \in I$. In Line 4, the circle that covers the maximum number of demand points is selected. The center of the selected circle (a_{j^*}, b_{j^*}) is assigned to (x, y) as shown in Line 5 to illustrate the selection of the first facility. Lines 6 to 9 represent the selection of the next best facility $j \in J$ that covers the highest uncovered demand point by the previously selected circle. If only one of the circles covers a maximum number of demand points, that circle is selected as the next best facility but if more than one set is eligible, we apply Lines 10 to 16 which illustrates the solving of the quadratic approach as illustrated in Algorithm 2 using Ipopt to find the minimum radius of those circles to select the one with the smallest radius as the next best circle to be selected. This process is repeated until all demand points are covered.

We demonstrate the application of Algorithms 1, 2 and 3 using a hypothetical test problem. This test case consists of 15 different communities to represent our demand points and 6 health facilities to set up. The aim of this experiment is to provide adequate health coverage for 15 demand points using 6 health facilities. Circles are used to represent the facilities with points within the circles to illustrate the various demand points used in this experiment. The coordinates for the center of the circle and radius are shown in Table 1. As shown below, we also give the x and y coordinates for the data set used to represent the demand point in Table 2.

TABLE 1 Coordinates of centers and radius for circles in Figure 3.1a

$x-$cordinates	2.11	2.59	6.25	9.19	1.37	3.59
$y-$cordinates	8.78	8.60	5.57	1.19	7.49	5.55
radius	4.5	4.7	5	2	4	3

TABLE 2 Coordinates of the demand points in Figure 3.1a

$x-$cordinates	3.55	5.25	1.74	7.49	4.65	9.42	5.63	4.75	10.82	5.25	3.04	7.31	3.14	5.58	8.35
$y-$cordinates	5.53	5.78	10.94	5.27	1.88	8.69	10.59	5.40	6.32	1.40	12.81	8.53	4.17	8.17	1.80

For this test case, $I = \{1, 2, \ldots, 15\}$ and $J = \{1, 2, \ldots, 6\}$. In order to select a cost-effective subset of facilities to be opened at a given set of potential locations, we first cover all demand points with the various circles. Demand points are considered covered when facilities provide maximum coverage within a certain area closer to the service point. We, therefore, generate a coverage matrix by considering facilities that serve demand points within the shortest distance possible. Table 3 below shows the coverage matrix used for this experiment where the columns corresponding to $H1-H6$ denote the demand points that the corresponding facility can cover. From the coverage matrix, 1 indicates if a facility located in $j \in J$ can cover a demand point $i \in I$; otherwise is 0. For example, demand point 3 can be covered by facilities 1,2,5,6 but, not by facilities 3 and 4.

TABLE 3 Coverage matrix for the hypothetical test problem with $m = 15$ and $n = 6$

	Facility Coverage					
	$H1$	$H2$	$H3$	$H4$	$H5$	$H6$
Coverage of demand point 1	1	1	1	0	1	1
Coverage of demand point 2	1	1	1	0	1	1
Coverage of demand point 3	1	1	0	0	1	1
Coverage of demand point 4	0	0	1	1	0	1
Coverage of demand point 5	0	0	1	1	0	1
Coverage of demand point 6	0	0	1	0	0	0
Coverage of demand point 7	1	1	0	0	0	0
Coverage of demand point 8	1	1	1	0	1	1
Coverage of demand point 9	0	0	1	0	0	0
Coverage of demand point 10	0	0	1	1	0	1
Coverage of demand point 11	1	1	0	0	0	0
Coverage of demand point 12	0	1	1	0	0	1
Coverage of demand point 13	1	1	1	0	1	1
Coverage of demand point 14	1	1	1	0	1	1
Coverage of demand point 15	0	0	1	1	0	0

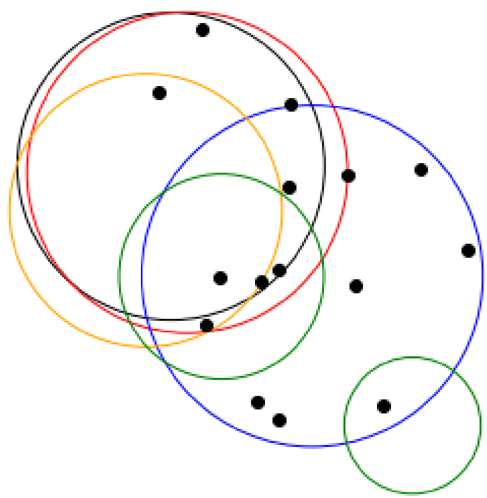
The geographical illustration of this test problem is given in Figure 3.1a where all 15 communities are covered by the 6 health facilities (hospitals). Thus the problem consists of nine decision variables $(x_1, x_2, x_3, x_4, x_5, x_6, x, y, R)$. As described in Algorithm 1, we first solve the SCP. The results obtained show that $x_1 = x_3 = 1$ and $x_2 = x_4 = x_5 = x_6 = 0$. Thus $\bar{J} = \{1, 3\}$ which illustrates the selection of Hospitals 1 and 3 to cover all demand points while Hospitals $2, 4, 5,$ and 6 have not been selected. Formulation 1 (the direct quadratic approach) is then applied using the selected circles to find the center and radius of the communication hub. Algorithm 1 produces a radius of 7.3714 with center $(4.3841, 7.0298)$.

We also applied Algorithm 2 which uses Formulation 2 to solve the same test problem. Algorithm 2 produces the radius of the communication hub and this also gives a radius of 7.3714 with center $(4.3841, 7.0298)$. Thus the center and radius of the communication hub using Algorithm 1 and Algorithm 2 produce the same results for this test case.

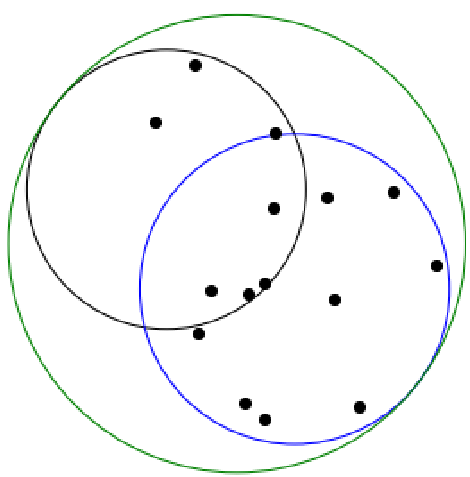
We similarly applied Algorithm 3 for this test problem. The main idea behind Algorithm 3 is to first select the circle that covers the maximum number of demand points. Circle 1 has been selected on the first iteration since it covers the maximum number of demand points. The next circle that covers the maximum number of demand points is circle 3. Algorithm 3, therefore, selects Circle 3 as the next best facility representation to set up. All demand points were covered after the selection of Circles 1 and 3 which stops the algorithm from further selections. This implies that 2 facilities were selected out of a total of 6 to supply the needs of the various demand points. It, therefore, reduces the total cost of the facility setup since every customer will equally have access to maximum health care through the 2 selected facilities instead of 6 facilities. We then applied Formulation 1 (the direct quadratic approach) and Formulation 2 (iterative means) to find the center and radius of the communication hub. Both Formulations produced a radius of (7.3709) with center $(4.3841, 7.0298)$ for the communication hub. In comparing Algorithm 3 to Algorithms 1 and 2, it can be observed that Algorithm 3 performs better in finding the radius of the communication hub since it gives the smallest radius as compared to Algorithms 1 and 2. More hypothetical test cases are explained in chapter 4 using Algorithm 3 to show its effectiveness in selecting the best facilities that produce the minimum radius for the communication hub. The results discussed above are depicted in Figure 3.1b showing the various hospitals selected that cover the 15 communities under study. The center and minimum radius for the communication hub are summarized in Table 4.

TABLE 4 Center and radius of communication hub

	Center (x, y)	Radius (R)
Algorithm 1	$(4.3841, 7.0298)$	(7.3714)
Algorithm 2	$(4.3841, 7.0298)$	(7.3714)
Algorithm 3	$(4.3841, 7.0298)$	(7.3709)



(A) Coverage of 6 facilities with 15 demand points



(B) A selection of 2 facilities selected out of 6

FIGURE 3.1: An illustration of facilities covering a set of demand points

CHAPTER 4

COMPUTATIONAL WORK

Hypothetical Problems and Real Data Analysis using Geographic Information Systems (GIS)

In this section, we first discuss the quality and effectiveness of Algorithm 3 using more hypothetical problems. We also provide a table that compares Algorithm 3 to Algorithms 1 and 2. A real-world test scenario using GIS has also been discussed to effectively analyze the importance of the various algorithms introduced in Chapter 3. For the hypothetical problems, we considered 11 facilities with 50 demand points and 15 facilities with 100 demand points in our discussion. We choose Dougherty County in Georgia as the study area for the real test problem. Dougherty County, Georgia, has some characteristics regarding its boundary shape and demand representation as discussed in section 4.2.

4.1. Model evaluation- on more hypothetical test problems

We discuss the effectiveness of algorithms using various hypothetical test cases in this section. The test cases discussed include 11 circles with 50 demand points, 15 circles with 100 demand points, and 10 circles with 50 demand points. As discussed previously, the circles represent the various facilities to be opened while the demand points represent the customers or communities that will be covered by the various facilities. Demand points are covered when facilities are able to provide maximum service to supply their needs within a short distance. The data set that represents the demand points for the 50 demand points with 11 circles and 100 demand points with 15 circles is obtained from the literature work of Canbolat and von Massow (2009). The centers of the circles were randomly generated in Python using the inbuilt uniform random number generation function. The radius of the circles was manually entered and increased to ensure all demand points are covered by at least one circle. The data set that represents the demand points for the 10 circles and 50 demand points is randomly generated using Python. The center for the 10 circles was obtained using an idea from the literature work of Liu (2022). The radius for this test case was also entered manually and increased to ensure that all demand points are covered by at least a circle. In generating the coverage matrix for the various test cases, we used 1 for demand points covered by a particular circle and 0 otherwise. The process involved in selecting the best facilities to be opened and finding the communication hub for the various test cases is described below.

We first consider the 11 circles with 50 demand point test case. We demonstrate the use of Algorithm 3 using this test case. Algorithm 3 is used to find the best facilities to be opened. As previously discussed, the main goal of Algorithm 3 is to select the best facilities to be opened and allow respective demand points to be covered by the selected circles. This is implemented by first selecting the circle that covers the maximum number of demand points. It then selects the next best circle with the maximum coverage until all demand points are covered. An illustration of this test case is shown in Figure 4.1. On the first iteration, Algorithm 3 selects circle 3 since it has the maximum number of demand coverage as shown in 4.2a. On the next iteration, circle 5 was selected followed by circle 9 to ensure the coverage of all demand points as shown in (4.2b-4.2c). The iteration stops since all demand points are covered. This shows that 3 facilities were selected to provide service to the 50 communities used in this experiment. The results produced for the minimum radius and center of the communication hub are 40.9598 and $(23.3924, 28.3349)$. As shown in Figure (4.3a-4.3c), the communication hub covers all selected circles with demand points effectively at a reduced distance between demand points and facilities. The reduction in the total number of facilities based on our test case illustrates cost-effectiveness. We also use Ipopt with Formulation 1 (Algorithm 1) and Formulation 2 (Algorithm 2) together with the selected circles from the SCP to find the center and radius of the communication hub.

We also consider a similar analysis used for the 11 circles and 50 demand points test for the 15 circles and 100 demand point test case as shown in Figure 4.4. From the results produced, 3 circles were selected as shown in 4.5. The communication hub as shown in Figure 4.6 has a radius of 42.8782 with a center of $(15.1734, 33.3349)$. The star located in the circle shows where the center of the communication hub can be located.

The results for more test problems are included in Table 5. We compare the results for the hypothetical test cases discussed above using Algorithms 1 to 4 as summarized in Table 5. In our observation with the test cases analyzed, Algorithm 3 works better than the other algorithms since it produces the smallest minimum radius. Our newly developed algorithm is, therefore, more effective in locating the best facilities to set up based on this analysis. Further discussion for Table 5 is discussed as follows.

Table 5 below, shows the results of the number of circles selected, as well as the radius and center for the communication hub from Algorithms $1, 2, 3$ and 4. From the table, n represents the total facilities, and m shows the total demand points. Considering problem 2 with 10 facilities and 25 demand points, the results produced by Algorithms 1 and 2 show a radius of 41.8685 and 45.9702. The radius for Algorithm 3 is 37.3539. For problem, 5 with 20 facilities and 25 demand points, The radius produced for Algorithms 1 and 2 are 40.2546 and 41.3795 while Algorithm 3 produces a result of 34.388. Subsequent results from Table 5 show that Algorithm 3 always produces a smaller radius as compared to the other algorithms. This shows that the distance between demand points and selected facilities is effectively minimized with the newly developed algorithm as compared to the others. Algorithm 3, therefore, works better to find the minimum radius.

TABLE 5 Results of more hypothetical test cases

	Algorithm 1				Algorithm 2				Algorithm 3 (Dynamic selection)				All selected			
Test Cases	$\#circles$	R	(x,y)	Time (s)	$\#x$	R	(x,y)	Time	$\#x$	R	(x,y)	Time	$\#x$	R	(x,y)	Time
Problem 1 (n,m)=(6,15)	2	7.3714	(4.3841,7.0298)	0.5319	2	7.3714	(4.3841,7.0298)	0.5319	2	7.3709	(4.38411,7.0298)	0.0803	6	8.4407	(4.8000, 5.9042)	0.1310
Problem 2 (n,m)=(10,25)	4	41.8685	(30.6000,19.6200)	0.1250	4	45.9702	(28.02746,20.8935)	0.3627	3	37.3539	(29.0809,19.2393)	0.1244	10	46.7459	(27.2466, 18.7470)	0.0690
Problem 3 (n,m)=(10,50)	4	41.7278	(24.5821,25.4136)	0.1580	4	43.0296	(36.9368,21.6637)	0.3456	4	39.8714	(26.0413,18.8307)	0.1406	10	75.1710	(49.9999,49.9999)	0.0839
Problem 4 (n,m)=(11,50)	3	44.3955	(21.6903,33.7499)	0.2650	3	44.3955	(21.6903,33.7499)	0.3706	3	40.9598	(23.3924,28.3349)	0.1992	11	53.0638	(37.0063, 37.3469)	0.14000
Problem 5 (n,m)=(10,100)	4	41.1066	(26.8690,27.8298)	0.2030	4	46.7449	(27.3982,18.9173)	0.2206	7	39.8714	(26.0413,18.8307)	0.1702	10	46.7459	(27.2466,18.7470)	0.1319
Problem 6 (n,m)=(15,100)	3	44.6503	(15.1734,33.3349)	0.4671	3	44.6503	(15.1734,33.3349)	0.3749	3	42.8782	(22.0175,32.7089)	0.1992	15	58.0638	(37.0063, 37.3469)	0.1849
Problem 7 (n,m)=(20,25)	3	40.2546	(30.9201, 36.5458)	0.0780	3	41.3795	(34.9719, 29.0329)	0.4883	4	34.388	(27.0506,23.7417)	0.1343	20	48.427	(26.2739,28.2879)	0.5367
Problem 8 (n,m)=(20,50)	4	40.7161	(26.5004, 27.8694)	0.1510	4	40.3998	(23.0827,19.3910)	0.2107	5	39.5973	(21.5515,30.4707)	0.1849	20	48.4274	(26.2739, 28.2879)	0.1879
Problem 9 (n,m)=(20,100)	4	44.6251	(18.3273,21.0272)	0.0460	4	44.62510	(18.3272,21.0271)	0.5489	5	42.5527	(24.4190, 32.5760)	0.0959	20	49.2395	(22.1753, 23.7359)	0.1309
Problem 10 (n,m)=(40,50)	11	34.4755	(23.9670, 27.6688)	0.1810	11	34.6124	(23.9470, 29.7689)	0.1895	7	31.3465	(25.5376, 32.5760)	0.0876	40	45.7886	(23.1875, 26.7550)	0.1429

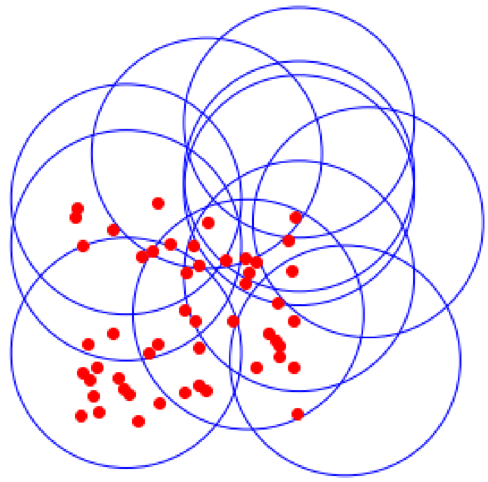
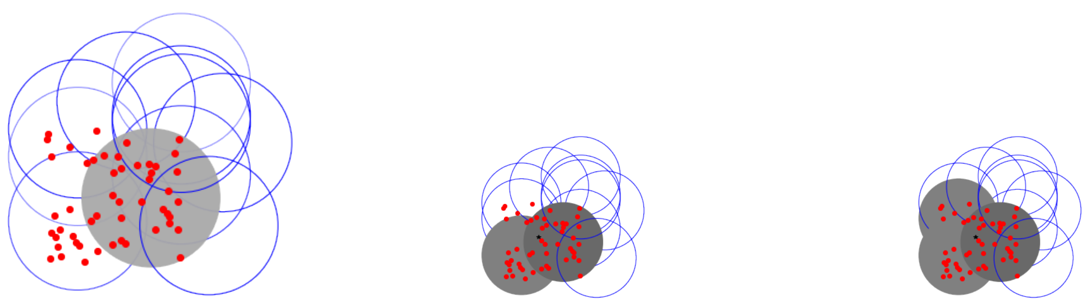


Figure 4.1: A set of clustered 50 demand points covered with 11 circles

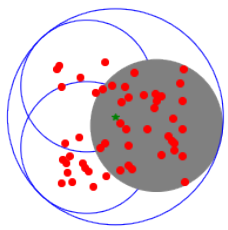


(a) Iteration 1: first highest density coverage circle

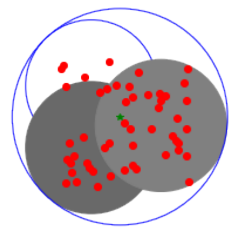
(b) Iteration 2: two highest density coverage circles

(c) Iteration 3: three highest density coverage circles

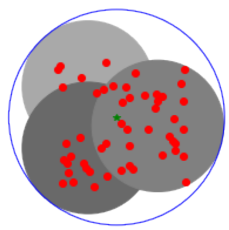
Figure 4.2: Order of circle selections with Algorithm 3 with $m = 50$ and $n = 11$



(A) First selected health facility with communication hub



(B) First two selected circles with communication hub



(C) Three selected circles with communication hub

FIGURE 4.3: Final lay of produced by Algorithm 3 with $m = 50$ and $n = 11$

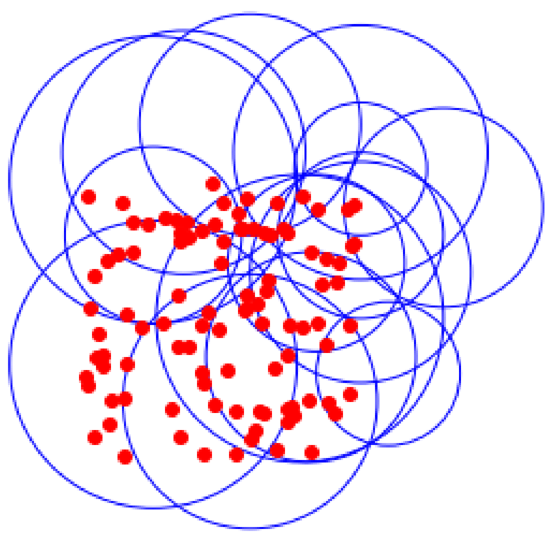


FIGURE 4.4: A set of clustered 100 demand points covered with 15 circles

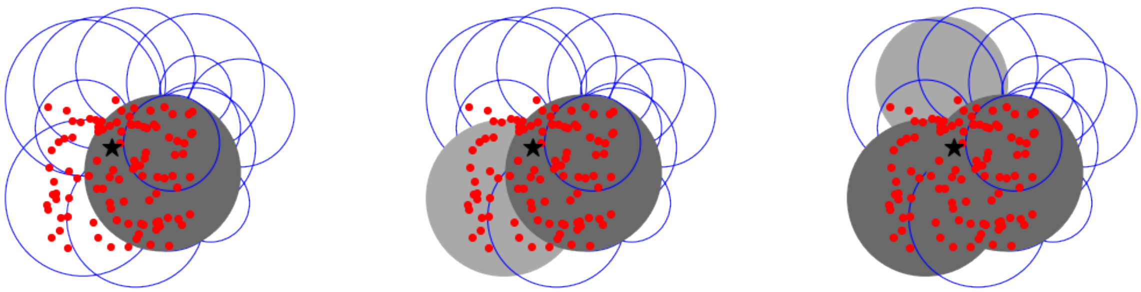


Figure 4.5: Order of circle selections where Algorithm 3 with $m = 100$ and $n = 15$

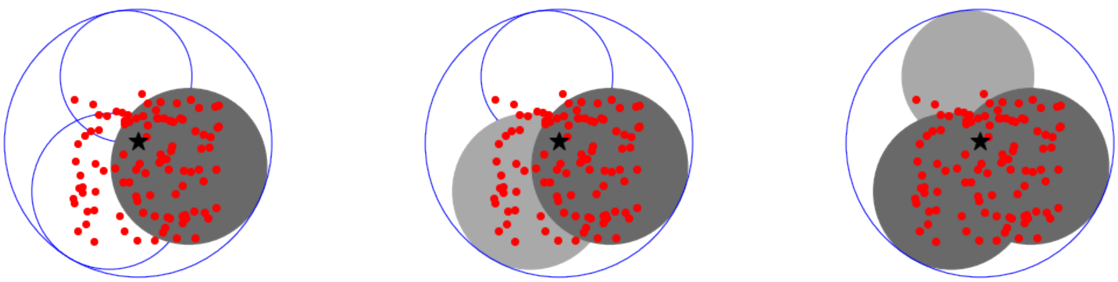


Figure 4.6: Final lay of produced by Algorithm 3 with $m = 100$ and $n = 15$

Using a similar analysis with Algorithm 3, we now consider the test case with 10 circles and 50 demand points as shown in Figure 4.7. With this test case, we consider all facilities to provide coverage to the various demand points. Thus all 10 facilities have been selected. The center and radius for the communication hub are $(49.9999, 49.9999)$ and 75.1710, respectively. From Table 5, Algorithm 4 shows some test cases conducted with all facilities considered as being selected. It can be observed that the radius of the communication hub for all test cases in Algorithm 4 produced a higher value than all algorithms. It is therefore important for the SCP model to be used to first select the best facilities to be opened in order to achieve the smallest radius possible for the communication hub.

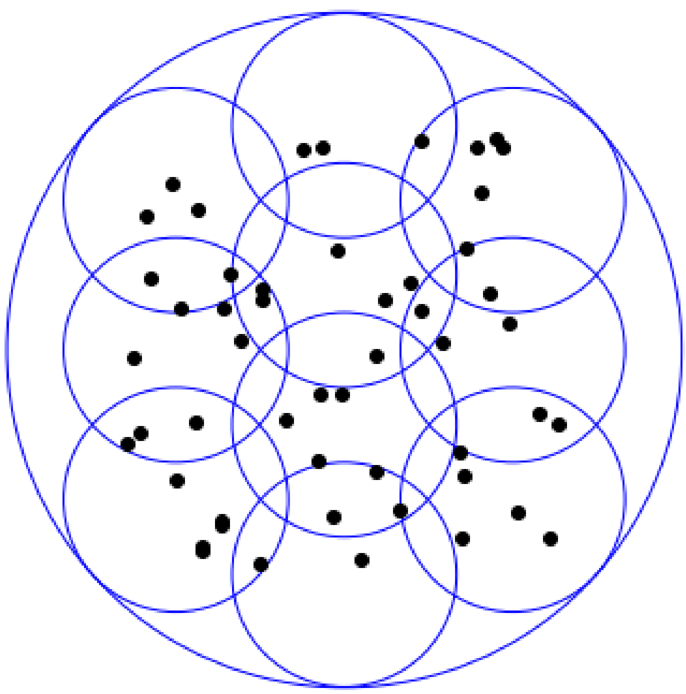


FIGURE 4.7: Communication hub covering 10 facilities with demand points

4.2. Model evaluation- on real data test case from Dougherty county

Dougherty county can be located in the southwestern (SW) part of Georgia (USA). According to the 2020 census, The SW part of Georgia has a total population of 496,433. Dougherty County has a population of about 85,790 according to the 2020 census and can be located in the city of Albany. The total area of the county according to the U.S. Census Bureau is 335 square miles, thus ($870km^2$). 329 square miles ($850km^2$) of the total is land and the remaining 5.9 square miles ($15km^2$) or (1.8%) is covered by water. Figure (4.8a-4.8b) displays the map of Dougherty county with 4.8a showing where the fire stations are located on the map and 4.8b showing the locations of the health centers. Our aim is to make an analysis with the application of FLP considering the highly populated areas in this county such that all communities in this county that are located in the highly populated areas have adequate access to health centers and fire stations within the shortest distance and also obtaining a maximum coverage from this facilities.

Our application of the facility location problem extends to solving real-world problems by studying demands from geographic information systems (GIS) in Dougherty county as shown in figure (4.8a - 4.8b). Our aim is to maximize the demand coverage and get the smallest radius that reduces the distance between the various facilities and demand points in Dougherty county, Georgia. In our analysis, we assumed that demand will always be served so far as it is within the highly populated area of the selected county under study. We, therefore, applied our model to maximize the demand coverage by finding a communication hub that provides additional coverage to some communities. We used circles to illustrate the facilities to set up and considered a radius of $2km$, $4km$, and $6km$ for each circle in this analysis. The buffer tool in ArcGIS is used to draw the $2km$, $4km$, and $6km$ buffers which cover all demand points, especially in highly populated areas. We give a geographical representation as shown below in Figure (4.9a - 4.9c) for fire stations and Figure (4.9d - 4.9f) for health centers. Note that Figure 4.9 represents both the fire stations and the health centers.

We applied various types of demand representations before conducting the sensitivity analysis as we aim to find solutions to the computational complexities in representing demand in GIS. To make an effective analysis, we used ArcGIS to visualize the various demand representations and applied the buffer tool to draw the circles that covers the various demand points. All analyses have been carried out using data from GIS for Dougherty county and Julia programming language is used to compute the various models. The x and y coordinates for the fire stations and health centers are shown in Tables 6 and 7. The coordinates were obtained based on the positions of the fire and health centers on the map shown in Figures (4.9a - 4.9c) and (4.9d - 4.9f). We manually generated the coverage matrix by allocating 1 to demand points covered by a particular circle and 0 otherwise. Note that the highly populated area is highly considered in generating the coverage matrix. All rows of zeros were removed using a code written in python.

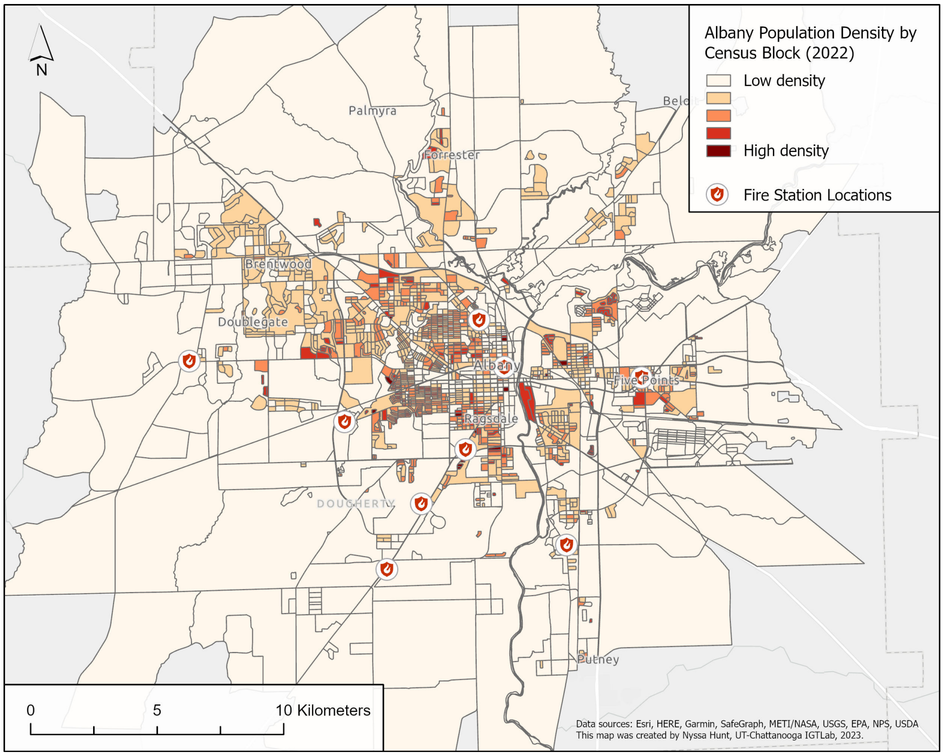
TABLE 6 Data showing the GIS coordinates for health centers in Dougherty county

$x-$cordinates	31.597	31.590	31.575	31.610	31.517
$y-$cordinates	84.163	84.158	84.105	84.218	84.115

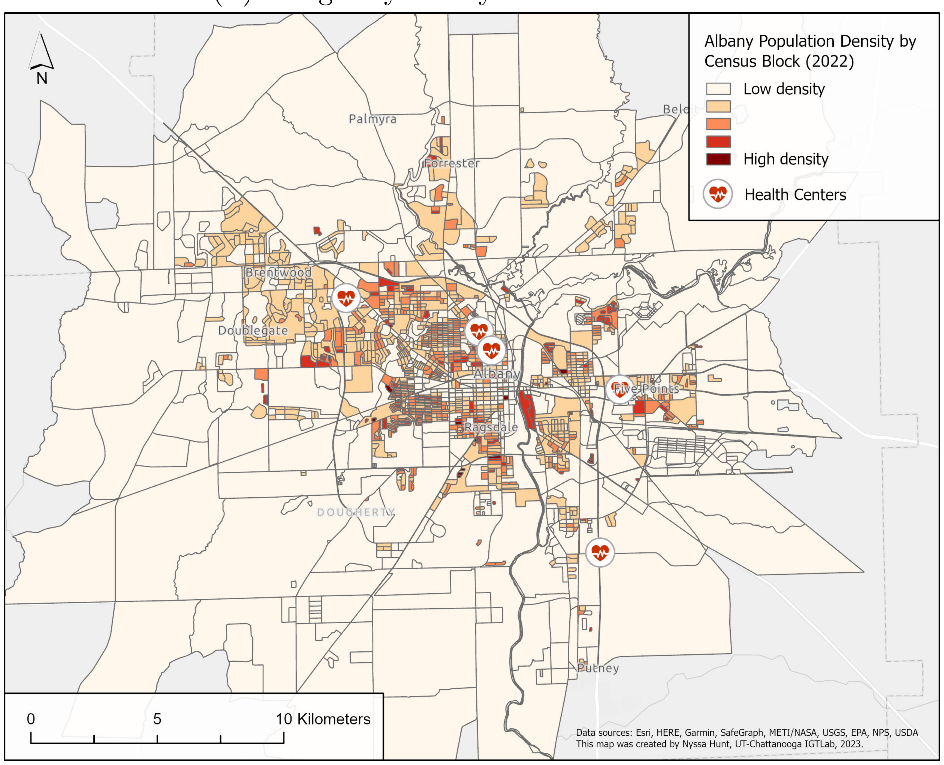
TABLE 7 The GIS coordinates for fire centers in Dougherty county

$x-$cordinates	31.576	31.581	31.598	31.517	31.563	31.552	31.586	31.510	31.533
$y-$cordinates	84.096	84.153	84.163	84.129	84.220	84.170	84.284	84.204	84.189

4.2.1. Demand representation in GIS:. Representing demand in GIS-based implementation requires the creation of demand objects, estimating the total demand for each of these created demand objects, and showing a relationship of coverage between these demand objects and potential service areas. In the literature work of Straitiff and Cromley (2010), the most efficient way to cover a particular region using circles is to apply equilateral triangle representation or polygon lattice-based representation. To create circular objects to cover all demand areas for potential facility sites, the buffer tool is used to create circles to represent the facilities. A hexagonal lattice is created with the center of the hexagons representing the various demand points in the study area, the various service sites are overlaid with the 'Identity' overlay tool. Each polygonal record in the resulting overlay represents a demand object and a prospective facility site whose service could accommodate that demand object. Figure 4.10 shows the fire stations and health centers in Dougherty county with buffers of $2km, 4km$, and $6km$ with hexagonal lattices. The circles in the figures show the various areas of coverage in the county under study. The center of the hexagons within the coverage areas has been labeled to represent the various demand points mostly in the highly populated areas. The coverage matrix used for our analysis in this research is generated based on the coverage areas for the fire stations and health centers as shown in Figure 4.10.

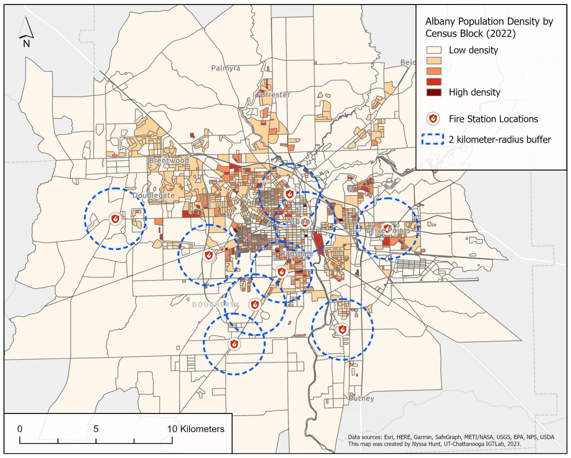


(A) Dougherty county with 9 fire stations

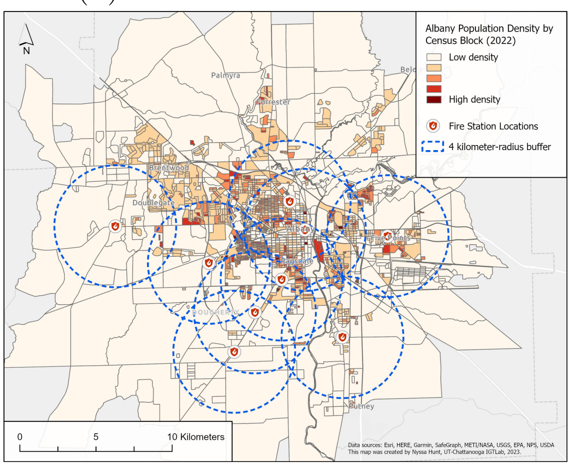


(B) Dougherty county with 5 health centers

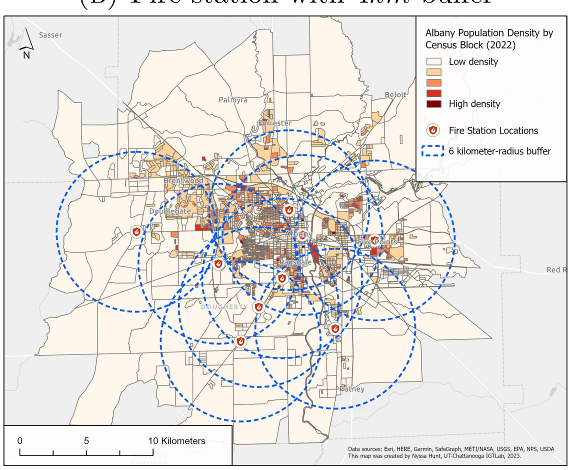
FIGURE 4.8: A map showing Dougherty county with fire stations and health centers



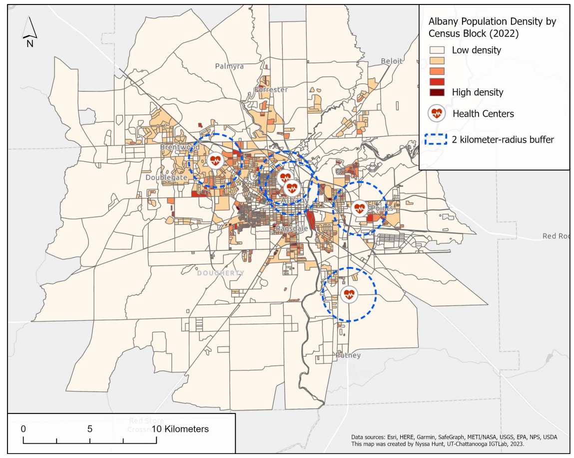
(A) Fire station with $2km$ buffer



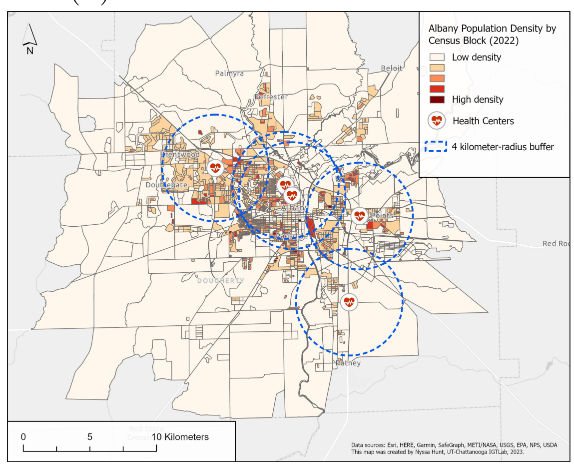
(B) Fire station with $4km$ buffer



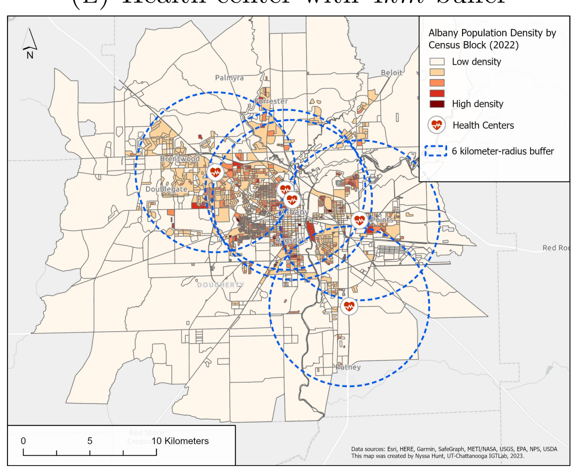
(C) Fire station with $6km$ buffer



(D) Health center with $2km$ buffer

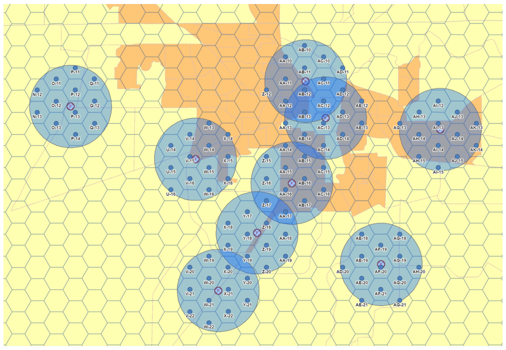


(E) Health center with $4km$ buffer

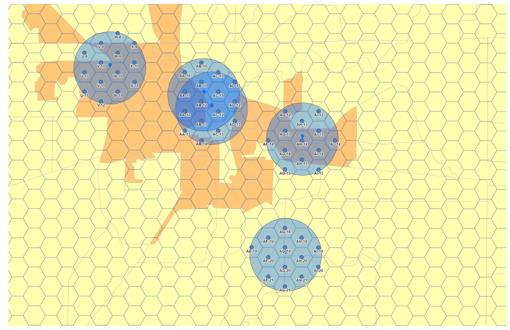


(F) Health center with $6km$ buffer

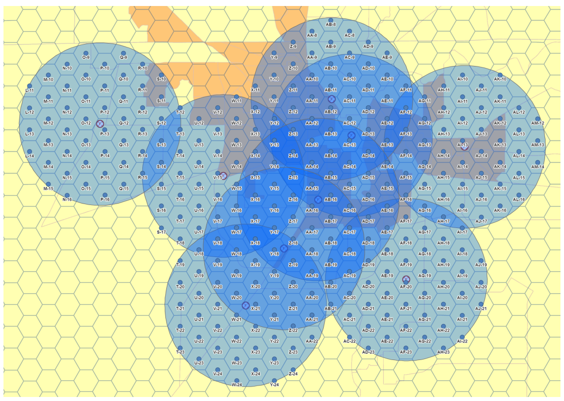
FIGURE 4.9: Fire stations and Health centers in Dougherty county with buffers



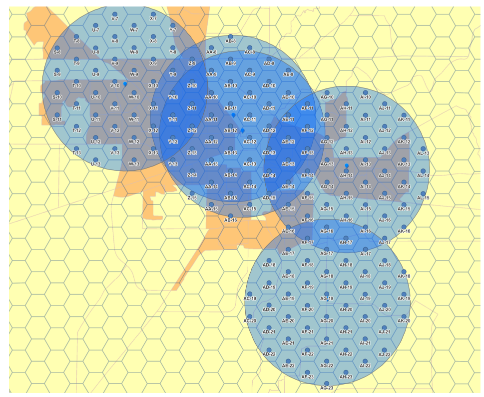
(A) Fire station with $2km$ buffer and hexagonal lattice



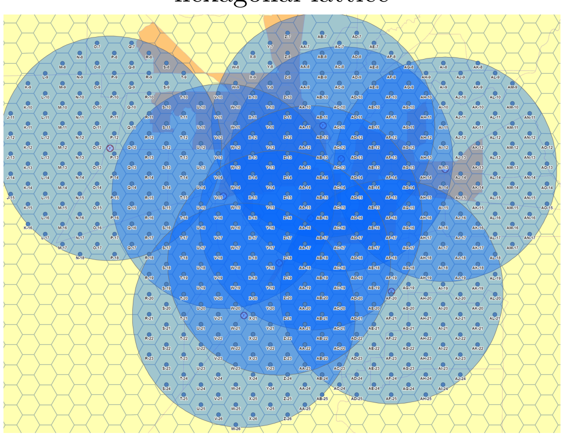
(D) Health center with $2km$ buffer and hexagonal lattice



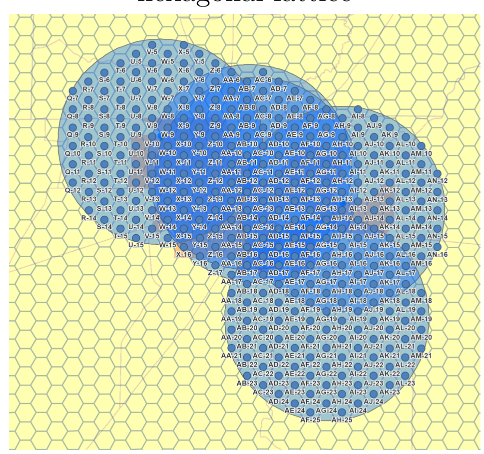
(B) Fire station with $4km$ buffer and hexagonal lattice



(E) Health center with $4km$ buffer and hexagonal lattice



(C) Fire station with $6km$ buffer and hexagonal lattice



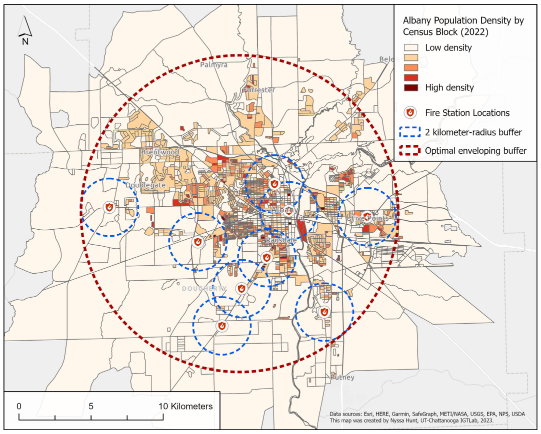
(F) Health center with $6km$ buffer and hexagonal lattice

FIGURE 4.10: Fire stations and Health centers in Dougherty county with hexagonal lattice's

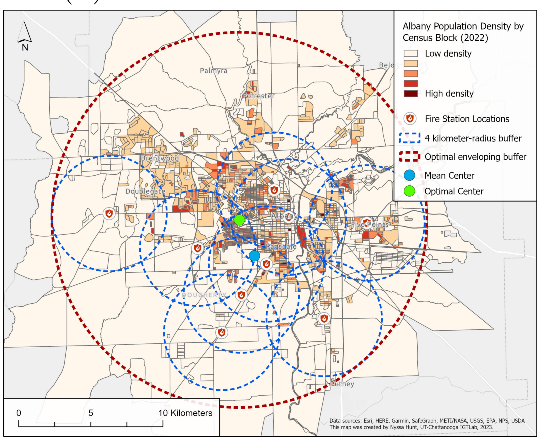
4.2.2. Sensitivity Analysis: In this section, we discuss the output of the selected facilities to be opened and the results obtained for the communication hub in the county under study. We applied our algorithms for the $2km$, $4km$, and $6km$ buffer test cases for both fire stations and health centers. Algorithms 1 and 3 are specifically applied for the real-world application of the FLP since Algorithm 2 produce the same results as Algorithm 1 after computing the test cases discussed in this experiment. Note that the results for Algorithm 1 in Table 8 and 9 are the same for Algorithm 2. A further discussion of the output produced is given below.

Considering the fire stations, we first used the centers generated from the map as discussed earlier with a $2km$ radius to cover all demand points, particularly at the highly populated region of the map. Algorithm 1 is applied on this test case, the results produced show that all 9 facilities were selected with the radius and center of the communication hub obtained as 2.094 and $(31.580, 84.189)$. Using Algorithm 3 for the $2km$ test case also, 9 circles were selected with a radius of 2.094 and center $(31.580, 84.189)$ for the communication hub. The communication hub gives additional coverage for some communities in the highly populated areas. The geographical representation of the $2km$ test case with the communication hub is shown in Figure 4.11a. A similar analysis as described for the $2km$ test case is used for the $4km$ and $6km$ test cases respectively and the results are shown in Table 8. The geographical representation for $4km$ and $6km$ is shown in Figure (4.11b - 4.11c). From Table 8, $Ar, R, (x, y)$ and $\#cir$ represents the total percentage coverage for the circles, the radius of the communication hub, the center of the communication hub, and the number of facilities selected respectively.

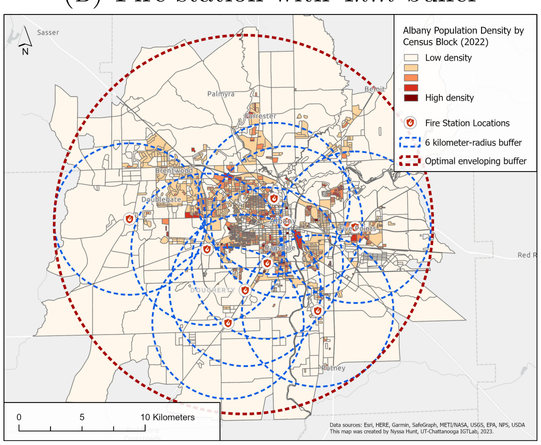
Considering the health centers, Algorithm 1 selects all 5 facilities for the $2km$ test case with the radius and center of the communication hub obtained as 2.069 and $(31.563, 84.166)$ respectively. The average coverage is 2.704%. We further used Algorithm 3 for the $2km$ test case and obtained 2.069 as the radius and $(31.563, 84.166)$ as the center. The average coverage with this algorithm is 2.704%. A similar analysis of Algorithms 1 and 3 is used for the $4km$ and $6km$ test case. The results produced is shown in Table 9. A geographical representation of the various buffers and the big buffer for the health centers is shown in Figure (4.11d- 4.11f).



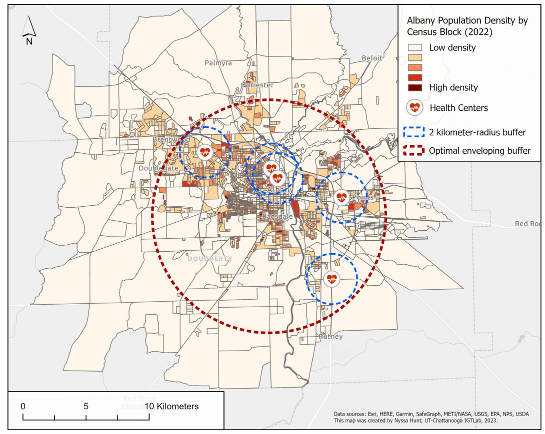
(A) Fire station with 2km buffer



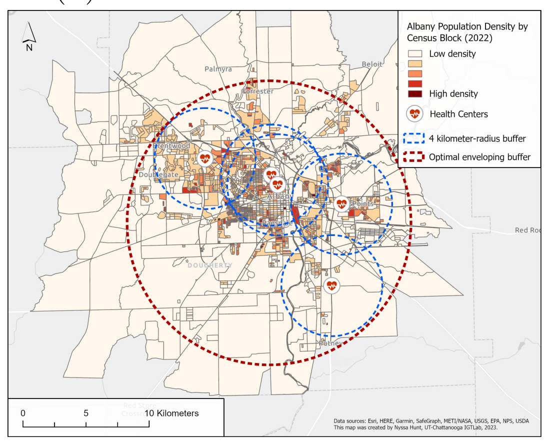
(B) Fire station with 4km buffer



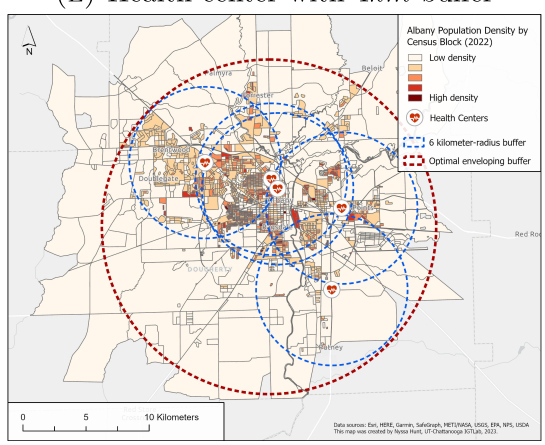
(C) Fire station with 6km buffer



(D) Health center with 2km buffer



(E) Health center with 4km buffer



(F) Health center with 6km buffer

FIGURE 4.11: Fire stations and Health Centers in Dougherty county with big buffer

TABLE 8 Sensitivity analysis for fire centers in Dougherty county

$r = 2$ for Algorithm 1				$r = 2$ for Algorithm 3			
$\#cir$	(x, y)	R	Ar	$\#cir$	(x, y)	R	Ar
9	$(31.580, 84.189)$	2.094	4.339%	9	$(31.580, 84.189)$	2.094	4.339%
$r = 4$ for Algorithm 1				$r = 4$ for Algorithm 3			
$\#cir$	(x, y)	R	Ar	$\#cir$	(x, y)	R	Ar
7	$(31.553, 84.156)$	4.064	18.742%	7	$(31.553, 84.156)$	4.064	18.742%
$r = 6$ for Algorithm 1				$r = 6$ for Algorithm 3			
$\#cir$	(x, y)	R	Ar	$\#cir$	(x, y)	R	Ar
6	$(31.553, 84.156)$	6.064	29.056%	6	$(31.569, 84.157)$	6.062	29.056%

TABLE 9 Sensitivity analysis for health centers in Dougherty county

$r = 2$ for Algorithm 1				$r = 2$ for Algorithm 3			
$\#cir$	(x, y)	R	Ar	$\#cir$	(x, y)	R	Ar
5	$(31.563, 84.166)$	2.069	2.704%	5	$(31.563, 84.166)$	2.069	2.704%
$r = 4$ for Algorithm 1				$r = 4$ for Algorithm 3			
$\#cir$	(x, y)	R	Ar	$\#cir$	(x, y)	R	Ar
5	$(31.563, 84.166)$	4.069	11.509%	5	$(31.563, 84.166)$	4.069	11.509%
$r = 6$ for Algorithm 1				$r = 6$ for Algorithm 3			
$\#cir$	(x, y)	R	Ar	$\#cir$	(x, y)	R	Ar
4	$(31.563, 84.166)$	6.069	20.062%	5	$(31.563, 84.166)$	6.069	20.062%

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1. Overview and summary

In this study, we examined the practical application of FLP in solving some hypothetical and real-world problems. Based on transportation costs, geographical demands, and the cost of facility setup, the idea of FLP is used by decision-makers to find the best location to set up a facility or warehouse. FLP can simply be defined as a place to set up facilities for the production of goods and services. Nonetheless, we need to note that FLP is not only important for companies when starting a business or when an existing business needs to be expanded. It is also important for healthcare systems to provide adequate services for various communities. Improper healthcare facility site selections have far-reaching effects on the community beyond simple cost and service measures but can lead to a high increase in morbidity and mortality rates. This highlights the importance of healthcare facility location modeling over that of other sectors (Ahmadi-Javid et al., 2017). FLP is therefore important because it provides cost benefits to organizations and identifies proximity to the sources of raw materials and transportation facilities. It is significant to also realize the importance of the hierarchical hub covering facility location problems in the distribution network in providing adequate services to customers. During the hub process, flows that originate from the same source but are headed in separate directions are merged with flows that originate from various sources but will end up in the same place. Consolidation occurs between hubs and along the route from origin to hub and hub to destination.

Therefore, this study provides a step-by-step approach to using different algorithms for the best location of facilities and finding a communication hub that works with the selected facilities to provide maximum service to the various demand points under study. We validate our newly developed mathematical model based on the test cases provided in this work and we observed that it works better than the other algorithms. We summarize the various chapters as follows.

In Chapter 1, we provide the literature review on the various concepts used in this research and also stated the motivation of our study. In Chapter 2, we discuss the background and existing methods for mathematical optimization. We define the various concepts in linear programming, non-linear programming, and convex optimization. We also provided some diagrams to show convex and non-convex sets and explained the various optimality conditions. This chapter also briefly discusses FLP, the Weber problem, and the $p-$center problem. The various approaches used in analyzing the $p-$center problem include the absolute $p-$center problem, vertex restricted $p-$center problem, capacitated $p-$center problem, and discrete $p-$center problem. If the facilities can only be placed at the network's vertices, then the problem is known as the vertex-restricted p-center problem. The absolute $p-$center problem occurs when facilities can be placed anywhere on the network. The goal of the capacitated p-center problem is to assign demand points to facility centers by choosing p locations such that the greatest distance between a demand point and its nearest center is reduced. The discrete $p-$center problem also aims to locate a set of $p-$ facilities so that demand points have the shortest possible travel times to the facilities that can best meet their needs.

In Chapter 3, we introduced the optimization models and the various algorithms used in making analysis in this research. We discussed the formulation as proposed by Xu et al. (2003) and also introduced a newly developed mathematical model to find the best facilities to set up and also get the minimum radius for our communication hub. We tested this algorithm on a hypothetical test case and discussed the results. In Chapter 4, we did more computational work. Thus, we further discussed more hypothetical test cases and applied our algorithms to a practical real-world application using Dougherty county in Georgia, USA as our case study. We considered the fire stations and health centers in this county and discussed the effectiveness of our algorithms in locating the best place to set up the communication hub using the minimum radius possible. Based on the various test cases analyzed in this research, we concluded that our newly developed algorithm works better than the already existing ones.

5.2. Future works

Moving forward, we will apply our algorithms to solve different test cases to verify their effectiveness. We aim to compare the newly developed algorithm (Algorithm 3) to the existing ones. We will do this by working on more hypothetical and real-world problems. One of the major problems we encountered in this work is how to convert the radius of the communication hub to a scale that will be accepted in ArcGIS for our real-world test problems. We aim to find a solution to this challenge. We also aim to incorporate the cost of the facility instead of uni-cost for all facilities and apply the multi-objective approach to minimize the radius and distance simultaneously. We will also develop a sub-gradient algorithm using Lagrange dual function to solve the proposed model.

In collaborating with my supervisor, Dr. Lakmali Weerasena, we are aiming to produce a very high and advanced research paper to be submitted to top peer-reviewed Operations Research/Mathematical journals for publication. We aim to use different polygons to represent our facilities and the communication hub and apply our algorithms to find optimal locations to set up facilities that will supply all demand points.

REFERENCES

Aardal, K., Pochet, Y., and Wolsey, L. A. (1995). Capacitated facility location: valid inequalities and facets. *Mathematics of Operations Research*, 20(3):562–582.

Aboueljinane, L., Sahin, E., and Jemai, Z. (2013). A review on simulation models applied to emergency medical service operations. *Computers & Industrial Engineering*, 66(4):734–750.

Adeleke, O. J. and Olukanni, D. O. (2020). Facility location problems: models, techniques, and applications in waste management. *Recycling*, 5(2):10.

Ahmadi-Javid, A., Seyedi, P., and Syam, S. S. (2017). A survey of healthcare facility location. *Computers & Operations Research*, 79:223–263.

Albareda-Sambola, M., Díaz, J. A., and Fernández, E. (2010). Lagrangean duals and exact solution to the capacitated p-center problem. *European Journal of Operational Research*, 201(1):71–81.

Alumur, S. and Kara, B. Y. (2008). Network hub location problems: The state of the art. *European journal of operational research*, 190(1):1–21.

Arora, J. S. (2004). Optimum design concepts: optimality conditions. *Introduction to optimum design*.

Aykin, T. (1995). The hub location and routing problem. *European Journal of Operational Research*, 83(1):200–219.

Barahona, F. and Anbil, R. (2000). The volume algorithm: producing primal solutions with a subgradient method. *Mathematical Programming*, 87(3):385–399.

Barilan, J., Kortsarz, G., and Peleg, D. (1993). How to allocate network centers. *Journal of Algorithms*, 15(3):385–415.

Beasley, J. E. (1996). *Advances in linear and integer programming*. Oxford University Press, Inc.

Bianchi, G. and Church, R. L. (1988). A hybrid fleet model for emergency medical service system design. *Social science & medicine*, 26(1):163–171.

Boyd, S., Boyd, S. P., and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Calık, H. (2013). *Exact solution methodologies for the p-center problem under single and multiple allocation strategies*. PhD thesis, Bilkent Universitesi (Turkey).

Canbolat, M. S. and von Massow, M. (2009). Planar maximal covering with ellipses. *Computers & Industrial Engineering*, 57(1):201–208.

Ceselli, A. and Righini, G. (2005). A branch-and-price algorithm for the capacitated p-median problem. *Networks: An International Journal*, 45(3):125–142.

Chou, Y.-H. (1990). The hierarchical-hub model for airline networks. *Transportation Planning and Technology*, 14(4):243–258.

Church, R. and ReVelle, C. (1974). The maximal covering location problem. In *Papers of the regional science association*, volume 32, pages 101–118. Springer-Verlag.

Cooper, L. and Katz, I. N. (1981). The weber problem revisited. *Computers & Mathematics with Applications*, 7(3):225–234.

Dantzig, G. B. (1990). Origins of the simplex method. In *A history of scientific computing*, pages 141–151.

Daskin, M. (1997). Network and discrete location: models, algorithms and applications. *Journal of the Operational Research Society*, 48(7):763–764.

Daskin, M. S. (2000). A new approach to solving the vertex p-center problem to optimality: Algorithm and computational results. *Communications of the Operations Research Society of Japan*, 45(9):428–436.

Díaz, J. A. and Fernández, E. (2002). A branch-and-price algorithm for the single source capacitated plant location problem. *Journal of the Operational Research Society*, 53(7):728–740.

Drezner, Z., Klamroth, K., Schöbel, A., and Wesolowsky, G. O. (2002). The weber problem. *Facility location: Applications and theory*, pages 1–36.

Ehrgott, M. (2002). Location of rescue helicopters in south tyrol. *International Journal of Industrial Engineering: Theory, Applications and Practice*, 9(1):16–22.

Ernst, A. T. and Krishnamoorthy, M. (1999). Solution algorithms for the capacitated single allocation hub location problem. *Annals of operations Research*, 86:141–159.

Farahani, R. Z., Asgari, N., Heidari, N., Hosseininia, M., and Goh, M. (2012). Covering problems in facility location: A review. *Computers & Industrial Engineering*, 62(1):368–407.

Farahani, R. Z., Hekmatfar, M., Arabani, A. B., and Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & industrial engineering*, 64(4):1096–1109.

Haghifam, M. and Shahabi, M. (2002). Optimal location and sizing of hv/mv substations in uncertainty load environment using genetic algorithm. *Electric Power Systems Research*, 63(1):37–50.

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

Hakimi, S. L. (1965). Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations research*, 13(3):462–475.

Henderson, S. G. and Mason, A. J. (2005). Ambulance service planning: simulation and data visualisation. In *Operations research and health care*, pages 77–102. Springer.

Holm, L. B., Lurås, H., and Dahl, F. A. (2013). Improving hospital bed utilisation through simulation and optimisation: with application to a 40% increase in patient volume in a norwegian general hospital. *International journal of medical informatics*, 82(2):80–89.

Hooker, J. N. (2015). Toward unification of exact and heuristic optimization methods. *International Transactions in Operational Research*, 22(1):19–48.

Hsu, V. N., Daskin, M., Jones, P. C., and Lowe, T. J. (1995). Tool selection for optimal part production: a lagrangian relaxation approach. *IIE transactions*, 27(4):417–426.

Hu, J. and Zhao, L. (2012). Emergency logistics network based on integrated supply chain response to public health emergency. *ICIC Express Letters*, 6(1):113–118.

Khachian, L. (1979). A polynomial time algorithm for linear programming. *Doklady Akademii Nauk SSSR*, 244:1093–1096.

Korani, E. and Eydi, A. R. (2014). Modeling of the capacitated single allocation hub location problem with a hierarchical approch. *International Journal of Engineering*, 27(4):573–586.

Liao, K. and Guo, D. (2008). A clustering-based approach to the capacitated facility location problem 1. *Transactions in GIS*, 12(3):323–339.

Liu, M. and Zhao, L. (2012). An integrated and dynamic optimisation model for the multi-level emergency logistics network in anti-bioterrorism system. *International Journal of*

Systems Science, 43(8):1464–1478.

Liu, Y. (2022). Two lower-bounding algorithms for the p-center problem in an area. *Computational Urban Science*, 2(1):1–19.

Love, R. F., Morris, J. G., and Wesolowsky, G. O. (1988). Facilities location.

Marmolejo-Saucedo, J. A. and Rojas-Arce, J. (2020). Hub location problem for medical care during epidemic outbreaks. In *International Conference on Computer Science and Health Engineering*, pages 117–129. Springer.

McCormack, R. and Coates, G. (2015). A simulation model to enable the optimization of ambulance fleet allocation and base station location for increased patient survival. *European Journal of Operational Research*, 247(1):294–309.

Minieka, E. (1970). The m-center problem. *Siam Review*, 12(1):138–139.

Murray, A. T. and Gerrard, R. A. (1997). Capacitated service and regional constraints in location-allocation modeling. *Location science*, 5(2):103–118.

Pereira, M. A., Coelho, L. C., Lorena, L. A., and De Souza, L. C. (2015). A hybrid method for the probabilistic maximal covering location–allocation problem. *Computers & Operations Research*, 57:51–59.

ReVelle, C. S., Williams, J. C., and Boland, J. J. (2002). Counterpart models in facility location science and reserve selection science. *Environmental Modeling & Assessment*, 7(2):71–80.

Rothlauf, F. (2011). Optimization methods. In *Design of Modern Heuristics*, pages 45–102. Springer.

Shanno, D. (2012). Who invented the interior-point method? *Optimization Stories, DOCUMENTA MATHEMATICA, Journal der Deutschen Mathematiker-Vereinigung, S*, pages 55–64.

Snyman, J. A., Wilke, D. N., et al. (2005). *Practical mathematical optimization*. Springer.

Straitiff, S. L. and Cromley, R. G. (2010). Using gis and k= 3 central place lattices for efficient solutions to the location set-covering problem in a bounded plane. *Transactions in GIS*, 14(3):331–349.

Suzuki, A. and Drezner, Z. (1996). The p-center location problem in an area. *Location science*, 4(1-2):69–82.

Toregas, C., Swain, R., ReVelle, C., and Bergman, L. (1971). The location of emergency service facilities. *Operations research*, 19(6):1363–1373.

Wei, H. (2008). *Solving continuous space location problems.* PhD thesis, The Ohio State University.

Xu, S., Freund, R. M., and Sun, J. (2003). Solution methodologies for the smallest enclosing circle problem. *Computational Optimization and Applications*, 25(1):283–292.

APPENDIX A

CODE FOR SELECTING BEST FACILITIES TO SETUP

In this appendix, we provide the code used to implement the various algorithms discussed in this thesis. The code is written and presented using Julia programming language, which can be run using a Julia interpreter. The codes are designed to take input data in a specific format and output the results of the algorithm in a structured format that can be easily analyzed and interpreted. The various codes provided generate the coverage matrix, select the best facilities to set up, and find the communication hub's radius and center. The code for Algorithm 3 which selects the best circles dynamically is also added in this Appendix.

```
##In this code, we consider 10 facilities with 25 customers.
The code runs to select the best facilities to set up that will
cover all demand points.


using JuMP
using Mosek
using MosekTools
model= Model(Mosek.Optimizer)
#Let parameter
m=25 # num customers
NumVar=10 # num hospitals
ObjCoefficient=[1 1 1 1 1 1 1 1 1 1] # Objective co-efficient representing
the 10 different hospitals

len=50
using Random
Random.seed!(1234)
using Distributions
xc=rand(Uniform(1,len),NumVar)
yc=rand(Uniform(1,len),NumVar)

r= [20, 20 , 20 , 20 ,20, 20, 20, 20, 20, 20] # Radiuus of circle

using DelimitedFiles, CSV, DataFrames
dlm = readdlm("coveragen10p25.txt") ### COVERAGE MATRIX

###Declare the variables###
```

```
@variable(model, x[i=1:NumVar], Bin)

###Define the objective function###
@objective(model, Min, sum(ObjCoefficient[i]*x[i] for i=1:NumVar))

###Define your constraints###

@constraint(model, constr[i=1:m],    sum(dlm[i,j]*x[j] for j=1:NumVar)
>= 1)


##Showing and solving the model##

@show model
print(model)
optimize!(model)
@show termination_status(model)
@show primal_status(model)
@show dual_status(model)
@show objective_value(model)
for i in 1:NumVar
    println("x[$i] = ", value(x[i]))
end
```

APPENDIX B

CODE FOR FINDING THE RADIUS AND CENTER OF THE COMMUNICATION HUB (THE HEURISTIC APPROACH)

```
## This code provides the heuristic approach used in finding the center
and radius of the communication hub. The code consists of 5 selected
facilities that provide coverage for some demand points.


using JuMP
import Juniper
import Ipopt
 NumVar = 5
    a = [46.89, 18.36, 24.81, 14.31, 45.81] #x-cordinates of selected circle
    b = [34.21, 34.21, 15.03, 5.03, 17.23] #y-cordinates of selected circle
    r = [20, 20, 20, 20, 20 ] #Radius of selected circles

function inner_loop( R)

  model = Model(
        optimizer_with_attributes(
            Juniper.Optimizer,
            "nl_solver" => optimizer_with_attributes(
                Ipopt.Optimizer,
                MOI.Silent() => true,
            ),
        ),
    )

    @variable(model, x >= 0)
    @variable(model, y >= 0)
    @variable(model, z >= 0)

    @objective(model, Min, x^2 + y^2 - z)
    @constraint(
        model,
        [i in 1:NumVar],
        -2 * x * a[i] - 2 * b[i] * y + z <= (R - r[i])^2 - a[i]^2 - b[i]^2)


    #print(model)
    optimize!(model)

    return value(x), value(y), value(z)
end
```

```
function main()
    count=0;
    x, y, z, R = 0.0, 0.0, Inf,0
    while abs(x^2 + y^2 - z) >= 0.01
        R = 0.0 ### Added this line
        for i in 1:length(a)
            R = max(R, sqrt((a[i] - x)^2 + (b[i] - y)^2) + r[i])
        end
        x, y, z = inner_loop(R)
        count=count+1;
        @show(count)
        println("The abs value is: abs(x^2+y^2-z) = " , abs(x^2+y^2-z))
    end
    @show(x)
    @show(y)
    @show(R)
end
main()
```

APPENDIX C

QUADRATIC APPROACH-ALL IN ONE METHOD (AIO)

```
## This code is used to find the radius and center of the communication
hub using the direct method also known as the all-in-one approach.
The code consists of 4 selected facilities used to cover 25 demand points.


using JuMP
import Juniper
import Ipopt

a = [46.89, 18.36, 24.81, 14.31, 45.81] #x-cordinates of selected circle
b = [34.21, 34.21, 15.03, 5.03, 17.23] #y-cordinates of selected circle
r = [20, 20, 20, 20, 20 ] #Radius of selected circles
NumVar = size(r)[1][1]


function solveAiO( )

  model = Model(
        optimizer_with_attributes(
            Juniper.Optimizer,
            "nl_solver" => optimizer_with_attributes(
                Ipopt.Optimizer,
                MOI.Silent() => true,
            ),
        ),
    )

    @variable(model, X)
    @variable(model, Y)
    @variable(model, R >= 0)

    @variable(model, x[i in 1:NumVar] )
    @variable(model, y[i in 1:NumVar] )
    @variable(model, z[i in 1:NumVar] >= 0)

    @objective(model, Min, R)
    @constraint( model, xeq[i in 1:NumVar], x[i] + X == a[i])
    @constraint( model, yeq[i in 1:NumVar], y[i] + Y == b[i])
    @constraint( model, zeq[i in 1:NumVar], z[i] + r[i] == R)

    @constraint( model, xyz[i in 1:NumVar], x[i]^2 + y[i]^2 <= z[i]^2)
```

```
    #print(model)
    optimize!(model)

    return value(X), value(Y), value(R)
end

function main()
    X, Y, R = solveAiO()
    @show(X)
    @show(Y)
    @show(R)
end
main()
```

APPENDIX D

DYNAMIC SELECTION OF BEST FACILITIES TO SET UP

```
## This code is used to select the best facilities to set up dynamically.


using JuMP
import Juniper
import Ipopt

NumVar=10
len=50
using Random
Random.seed!(100)
using Distributions
a1=rand(Uniform(1,len),NumVar) # X-cordinates of Circle
b1=rand(Uniform(1,len),NumVar) # Y-cordinates of Circle
print(a1,b1)

r1= [20, 20 , 20 , 20 ,20, 20, 20, 20, 20, 20] # Radiuus of circle

using DelimitedFiles, CSV, DataFrames
A = readdlm("coveragen10p25.txt")



m,n=size(A)
CoveredItems=zeros(Int64,m)
Coverage=zeros(Int64,n)
for i in 1:n
    sumval=sum(A[:,i])
    Coverage[i]=sumval
end
println("Intial Coverage is ", Coverage )
SelectedSolution=zeros(Int64,n)
maxval,maxpos=findmax(Coverage)
InitialBarJ=Vector{Int64}()
InitialR=Vector{Int64}()
  for q in 1:n
      if (maxval==Coverage[q])
      push!(InitialBarJ, q)
      push!(InitialR, r1[q])
      end
  end
println("InitialBar is ", InitialBarJ)
println("InitialR is ", InitialR)
```

```
count=maxval

if (length(InitialBarJ)==1)
CoveredItems=A[:,maxpos]
SelectedSolution[maxpos]=1
     SelectedSet=maxpos
     println("Initial Set", maxpos)
     println("Initial Cover", CoveredItems)
else
     minR,minRpos=findmin(InitialR)
     CoveredItems=A[:,InitialBarJ[minRpos]]
     SelectedSolution[InitialBarJ[minRpos]]=1
     println("Initial Set ", InitialBarJ[minRpos])
     println("Initial Cover ", CoveredItems)
     SelectedSet=InitialBarJ[minRpos]
end
#Step II
UncoveredItems=zeros(Int64,m)
for i in 1:m
    if (CoveredItems[i]==0)
        UncoveredItems[i]=1
    else
       A[i, :] .= 0

    end
end
A # print A for checking


println("Initial Solution is", SelectedSolution)


println(A)
a = Vector{Float64}()
b = Vector{Float64}()
r = Vector{Float64}()
push!(a, a1[SelectedSet])
push!(b, b1[SelectedSet])
push!(r, r1[SelectedSet])
println(a)
println(b)
println(r)
```

```
BarJ = Vector{Int64}()
push!(BarJ, SelectedSet)
println("BarJ", BarJ)

function main()
#for p in 1:m
    while count<m
        for i in 1:m
            if(CoveredItems[i]==0)
                PossibleR = Vector{Float64}()
                BarJ = Vector{Int64}()
                IndiCover= Vector{Int64}()
                IndiCoverSet = Vector{Int64}()
                for p in 1:n
                    if(A[i,p]==1 && SelectedSolution[p]==0)
                        push!(IndiCover, sum(A[:,p]))
                        push!(IndiCoverSet, p)
                    end
                end
                println("IndiCover is ", IndiCover)
                println("IndiCoverSet is ", IndiCoverSet)
               # println("checking point")
               maxcover,maxcoverset=findmax(IndiCover)
                println(findmax(IndiCover))
                uniquebar=Vector{Int64}()
                for q in 1:length(IndiCover)
                    if (maxcover==IndiCover[q])
                        push!(uniquebar, IndiCoverSet[q])
                    end
                end
                println("uniquebar is ", uniquebar)
                println("length(IndiCover) is ", length(IndiCover))
                if (length(IndiCover)==1)
                        push!(a, a1[uniquebar[1]])
                        push!(b, b1[uniquebar[1]])
                        push!(r, r1[uniquebar[1]])
                        println(a)
                        println(b)
                        println(r)
                        SelectedSolution[uniquebar[1]]=1
                        R,X,Y=solveAiO(a,b,r,sum(SelectedSolution) )
                        println(R)
```

```
            println(X)
            println(Y)
        for r in 1:m
            if(A[r,uniquebar[1]]==1 && CoveredItems[r]==0)
            CoveredItems[r]=1;
            A[r,:] .=0
                #zeros(1,m);
            end
        end
       println(" Solution is ", SelectedSolution)
       println(" Updated A ", A)
       #break
    else
        for j in 1:length(uniquebar)

            SelectedSolution[uniquebar[j]]=1
            NumVar=sum(SelectedSolution)
            println(NumVar)
            push!(a, a1[uniquebar[j]])
            push!(b, b1[uniquebar[j]])
            push!(r, r1[uniquebar[j]])
            println(a)
            println(b)
            println(r)
            R,X,Y=solveAiO(a,b,r,NumVar )
            println(R)
            println(X)
            println(Y)

            push!(PossibleR, R)
            push!(BarJ , uniquebar[j])
            SelectedSolution[uniquebar[j]]=0
            deleteat!(a,NumVar)
            deleteat!(b,NumVar)
            deleteat!(r,NumVar)
            println(a)
            println(b)
            println(r)
        end

    println("PossibleR ",PossibleR)
    BestR,BestPos=findmin(PossibleR);
```

```
                println("BestPos ",BestPos)
                println("BarJ ",BarJ)
                BestSet=BarJ[BestPos]
                println(BestSet)

                SelectedSolution[BestSet]=1
                        push!(a, a1[BestSet])
                        push!(b, b1[BestSet])
                        push!(r, r1[BestSet])
                        println(a)
                        println(b)
                        println(r)
                    for k in 1:m
                        if(A[k,BestSet]==1 && CoveredItems[k]==0)
                        CoveredItems[k]=1;
                        A[k,:].=0;
                        end
                    end
                end
            end

        end
            println("CoveredItems ",CoveredItems)
            println("SelectedSolution ",SelectedSolution)
            count=sum(CoveredItems)
            println("count",count)
            if (count==m)

            break
             end

    end #BigForLoop/while loop ends

end # file Ends


function solveAiO(a,b,r,NumVar )

  model = Model(
        optimizer_with_attributes(
            Juniper.Optimizer,
            "nl_solver" => optimizer_with_attributes(
```

```
                    Ipopt.Optimizer,
                    MOI.Silent() => true,
                ),
            ),
    )

    @variable(model, X)
    @variable(model, Y)
    @variable(model, R >= 0)

    @variable(model, x[i in 1:NumVar] )
    @variable(model, y[i in 1:NumVar] )
    @variable(model, z[i in 1:NumVar] >= 0)

    @objective(model, Min, R)
    @constraint( model, xeq[i in 1:NumVar], x[i] + X == a[i])
    @constraint( model, yeq[i in 1:NumVar], y[i] + Y == b[i])

    @constraint( model, zeq[i in 1:NumVar], z[i] + r[i] == R)

    @constraint( model, xyz[i in 1:NumVar], x[i]^2 + y[i]^2 >= z[i]^2)

    #@show model

    #print(model)
   optimize!(model)
    println("PossibleR ",R)
    println("PossibleX ",X)
    println("PossibleY ",Y)
    @show(R)
    @show(X)
    @show(Y)
    return value(R), value(X), value(Y)
end

main()
```

APPENDIX E

CODE FOR GENERATING THE COVERAGE MATRIX

```
## This code is used in generating the coverage matrix. It consists
of 20 facilities with 50 demand points where we considered a radius of 20.

a=[19.7, 30.2, 44.3, 1.7, 43.3, 13.0, 2.6, 11.8, 15.0, 28.1, 19.2, 45.6,
21.9, 48.8, 49.6, 47.6, 35.5, 48.7, 15.7, 18.2, 4.6, 4.1, 24.9, 19.3, 3.0,
1.2, 5.9, 39.0, 40.6, 5.2, 38.1, 9.5, 28.1, 28.2, 48.5, 26.9, 40.5, 9.7,
6.6, 38.3, 49.3, 45.3, 25.5, 25.0, 2.9, 33.9, 29.6, 44.7, 10.8, 17.4]

b=[5.9, 45.1, 48.0, 21.0, 7.8, 3.2, 9.0, 1.8, 17.8, 18.5, 18.4, 23.8, 15.9,
40.5, 13.6, 3.6, 41.0, 23.8, 37.8, 39.0, 10.7, 18.6, 26.0, 49.3, 12.3, 46.4,
13.6, 34.3, 36.5, 7.5, 31.7, 20.8, 35.9, 9.9, 34.7, 40.0, 13.8, 43.5, 3.9,
37.2, 46.4, 27.7, 34.1, 8.1, 40.0, 36.9, 8.5, 19.5, 11.1, 16.9]

n=20
m=50
r=20
len=50
using Random
Random.seed!(45)
using Distributions
xs=rand(Uniform(1,len),n)
ys=rand(Uniform(1,len),n)
print(xs,ys)

function distance(xs, ys, a, b,n,m, r)
#      n = length(xs)
#      m = length(a)
    coverage = zeros(m, n)  # create the matrix and fill it with zeros
for j in 1:n
           for i in 1:m
               if sqrt( (xs[j]-a[i])^2 + (ys[j]-b[i])^2 ) <= r
                   coverage[i,j] = 1
                   @show i j coverage
               end
           end
       end
    return coverage
end
coverage=distance(xs, ys, a, b,n,m,r)

#Print the coverage matrix as a text file
#importing the module
```

```
using DelimitedFiles
writedlm("coveragen20p50.txt", coverage)
```

VITA

Eric Nartey Onyame was born in Accra, Ghana, to Emmanuel and Rebecca Onyame. He attended Dodowa Presby Primary School and Shai D/A Junior High School in Ghana. He then continued to Ghanata Senior High School also in Ghana where he majored in the study of General Arts. With his excellent performance in the West African Certificate Examination after his high school education, he was admitted into the University of Cape Coast (UCC) in Ghana where he studied for his first degree (BSc) in Mathematics with Economics. While at UCC, Eric received the Dean's award three consecutive times for his exceptional performance in Mathematics and Economics. In July 2020, he graduated with first-class honors and was among the top 5% of excellent graduates from his department.

Afterward, he served as a teaching assistant at the Mathematics department of the University where he was involved in teaching Calculus and Algebra. In August 2021, he gained admission into the University of Tennessee at Chattanooga with a graduate teaching assistantship award. Eric is currently working with Professor Lakmali Weerasena in the research study of facility location science in the area of Mathematical optimization.

Eric Nartey Onyame will graduate with a Masters of Science degree in Applied Mathematics in May 2023. He plans to begin his Ph.D. at University of Virginia in August 2023.