A Multi-Objective Optimisation Approach for the Location of a Public Primary Healthcare Facility

Lehlohonolo Moche

(708451)

School of Mechanical, Industrial and Aeronautical Engineering
University of the Witwatersrand
Johannesburg, South Africa

Supervisor: Dr. Joke Bührmann

A research report submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, in partial fulfilment of the requirements of the degree of Master of Science in Engineering (Industrial)

6 December 2018

Faculty of Engineering and the Built Environment

Private Bag 3, Wits 2050, South Africa • Telephone (011) 717 – 7007 • Fax: (011) 717 7009 • Email: febe.pg@wits.ac.za

PLAGIARISM DECLARATION TO BE SIGNED BY ALL HIGHER DEGREE STUDENTS

SENATE PLAGIARISM POLICY: APPENDIX ONE

I Lehlohonolo U. Moche (Student number: 708451) am a student registered for the degree of MSc Engineering (Ind) in the academic year ______.

I hereby declare the following:

- I am aware that plagiarism (the use of someone else's work without their permission and/or without acknowledging the original source) is wrong.
- I confirm that the work submitted for assessment for the above degree is my own unaided work except where I have explicitly indicated otherwise.
- I have followed the required conventions in referencing the thoughts and ideas of others.
- I understand that the University of the Witwatersrand may take disciplinary action against me if there is a belief that this in not my own unaided work or that I have failed to acknowledge the source of the ideas or words in my writing.

Signature: ____________ Date: 6 December 2018.

Abstract

The location allocation decisions for hospitals and clinics are essential to communities. Ensuring that these facilities are firstly, accessible to patient populations and secondly, equitably distributed such that each patient population travels equal or similar distances to their nearest facility is important to urban planning and development. This study thus developed a multi-objective optimisation approach to locate public primary care facilities such that the objectives of accessibility and equity are achieved. The first model formulated the accessibility objective using the p-median location problem and the equity objective using the p-center problem. The second model formulated the accessibility objective using the p-median problem and the coverage objective using the maximal coverage location problem. The trade-offs between the accessibility and equity objectives were investigated and mapped on a Pareto frontier. In the first model it was found that there was a possible trade-off between accessibility and equity, because as equity increased, accessibility decreased. In the second model it was found that there was a trade-off between accessibility and equity until a specific maximum allowable distance value between a facility and the demand points it serves, thereafter a trade-off no longer existed, because as equity increased, access increased for those demand points covered at the expense of those demand points not covered.

Contents

Abstract i

List of Figure iv

List of Tables v

List of Acronyms 1

1 Introduction 2

1.1 Research Background . . . 2
1.2 Problem Statement . . . 3
1.3 Research Questions and Objectives . . . 4
1.3.1 Research Question . . . 4
1.3.2 Research Objectives . . . 4
1.4 Document Structure . . . 4

2 Literature Review 5

2.1 Discrete Facility Location Problems . . . 5
2.1.1 P-median Location Problem . . . 7
2.1.2 P-center Location Problem . . . 8
2.1.3 Maximal Coverage Location Problem . . . 9
2.2 Multi-Objective Decision Making . . . 9
2.2.1 Modelling Approaches . . . 10
2.2.2 Multi-Objective Optimisation . . . 10
2.2.3 Tools and Solution Methods . . . 13
2.3 Multi-Objective Facility Location . . . 14
2.3.1 Related Works . . . 14
2.4 Conclusion . . . 17

3 Method 18

3.1 Introduction . . . 18
3.2 Conceptual Design . . . 18
3.2.1 Model A: Accessibility versus Equity . . . 19
3.2.2 Model B: Accessibility versus Coverage . . . 21
3.3 Data Collection and Analysis . . . 22
3.4 Validation and Verification . . . 23

3.4.1 Model A: Accessibility versus Equity . . . 23
3.4.2 Model B: Accesibility versus Coverage . . . 25
3.5 Method . . . 25
3.5.1 Model A: Accessibility versus Equity . . . 25
3.5.2 Model B: Accesibility versus Coverage . . . 26
3.6 Scope . . . 26
3.7 Tools . . . 27

4 Results and Analysis **28**
4.1 Introduction . . . 28
4.2 Results . . . 28
4.2.1 Model A: Accessibility versus Equity . . . 28
4.2.2 Model B: Accessibility versus Coverage . . . 34
4.3 Analysis . . . 38
4.3.1 Facility Changes . . . 38
4.3.2 Assignment Changes . . . 46
4.3.3 Coverage Changes . . . 51
4.3.4 Accessibility vs Equity . . . 55
4.3.5 Pareto Efficiency . . . 62

5 Conclusion and Future Work **64**
5.1 Conclusion . . . 64
5.2 Future Work . . . 65

Bibliography **67**

Appendices **67**

A The Distance Matrices **68**

B Verification and Validation: Model A Lingo Scripts **71**

C Verification and Validation: Model B Lingo Scripts **75**

D Model A Lingo scripts **79**

E Model B Lingo scripts **90**

F Pareto frontier Graphs for Model A **105**

G Model A Results **109**

H Model B Results **113**

List of Figures

2.1 P-median Location Problem (PMLP) . . . 6
2.2 P-center Location Problem (PCLP) . . . 6
2.3 Maximal Coverage Location Problem (MCLP) . . . 7
2.4 Summary of Related Works . . . 16

4.1 Model A Results for DS1 to DS5 . . . 29
4.2 Model A Results for DS6 to DS13 . . . 30
4.3 Graph showing the Pareto frontier for DS2 . . . 31
4.4 Graph showing the Pareto frontier for DS5 . . . 31
4.5 Graph showing the Pareto frontier for DS12 . . . 32
4.6 Graph showing the Pareto frontier for DS13 . . . 32
4.7 Model A's Results for DS2 . . . 33
4.8 Model A's Results for DS5 . . . 33
4.9 Model A's Results for DS12 . . . 34
4.10 Model A's Results for DS13 . . . 34
4.11 Model B's Results for DS1 . . . 35
4.12 Model B's Results for DS2 . . . 36
4.13 Model B's Results for DS3 . . . 36
4.14 Model B's Results for DS5 . . . 37
4.15 Model B's Results for DS7 . . . 38
4.16 Location Allocation Diagram for DS2: Iteration one . . . 39
4.17 Location Allocation Diagram for DS2: Iteration six . . . 39
4.18 Location Allocation Diagram for DS2: Iteration eight . . . 40
4.19 Location Allocation Diagram for DS7: Iteration one . . . 41
4.20 Location Allocation Diagram for DS7: Iteration nine . . . 42
4.21 Location Allocation Diagram for DS12: Iteration one . . . 42
4.22 Location Allocation Diagram for DS12: Iteration thirteen . . . 43
4.23 Location Allocation Diagram for DS13: Iteration one . . . 43
4.24 Location Allocation Diagram for DS13: Iteration thirteen . . . 44
4.25 Location Allocation Diagram for DS7: Iteration six . . . 47
4.26 Location Allocation Diagram for DS7: Iteration seven . . . 47
4.27 Location Allocation Diagram for DS7: Iteration eight . . . 48
4.28 Location Allocation Diagram for DS13: Iteration seven . . . 49
4.29 Location Allocation Diagram for DS13: Iteration eight . . . 50
4.30 Location Allocation Diagram for DS1: Iteration one . . . 51
4.31 Location Allocation Diagram for DS1: Iteration five . . . 52

4.32 Location Allocation Diagram for DS1: Iteration seven 52
4.33 Location Allocation Diagram for DS1: Iteration eight 53
4.34 Location Allocation Diagram for DS1: Iteration thirteen 53
4.35 Location Allocation Diagram for DS7: Iteration one 54
4.36 Location Allocation Diagram for DS7: Iteration four 54
4.37 Location Allocation Diagram for DS7: Iteration six 55
4.38 Location Allocation Diagram for DS7: Iteration eight 55
4.39 A Median-based Solution 56
4.40 A Center-based Solution 57
4.41 The Location Allocation Network of Facility 4 for DS2: Iteration one 58
4.42 The Location Allocation Network of Facility 4 for DS2: Iteration six 59
4.43 A Maximal Coverage Solution 60
4.44 The Location Allocation Network of Facility 45 for DS1: Iteration eight 61
4.45 The Location Allocation Network of Facility 17 for DS1: Iteration thirteen 62

B.1 A Screenshot of the Excel Spreadsheet of the Test Model Results for Verification . . 74

F.1 Graph showing the Pareto frontier for DS1 105
F.2 Graph showing the Pareto frontier for DS3 106
F.3 Graph showing the Pareto frontier for DS4 106
F.4 Graph showing the Pareto frontier for DS6 107
F.5 Graph showing the Pareto frontier for DS7 107
F.6 Graph showing the Pareto frontier for DS11 108

G.1 Model A's Results for DS1 109
G.2 Model A's Results for DS3 110
G.3 Model A's Results for DS4 110
G.4 Model A's Results for DS6 111
G.5 Model A's Results for DS7 111
G.6 Model A's Results for DS11 112

H.1 Model B's Results for DS4 113
H.2 Model B's Results for DS6 114

List of Tables

3.1 The data used in Model A . . . 23
3.2 The data used in Model B . . . 23
3.3 The facilities selected by the test model . . . 24
3.4 The assignment results of the test model . . . 24
3.5 Validated model solutions . . . 25

4.1 DS2: Iteration one assignment . . . 40
4.2 DS2: Iteration six assignment . . . 40
4.3 DS2: Iteration eight assignment . . . 41
4.4 DS7: Iteration one assignment . . . 44
4.5 DS7: Iteration nine assignment . . . 44
4.6 DS12: Iteration one assignment . . . 45
4.7 DS12: Iteration thirteen assignment . . . 45
4.8 DS12: Iteration one assignment . . . 46
4.9 DS13: Iteration thirteen assignment . . . 46
4.10 DS7: Iteration six assignment . . . 48
4.11 DS7: Iteration seven assignment . . . 48
4.12 DS7: Iteration eight assignment . . . 49
4.13 DS13: Iteration seven assignment . . . 50
4.14 DS13: Iteration eight assignment . . . 51
4.15 The demand points assigned to facility 4 in iteration one . . . 58
4.16 The demand points assigned to facility 4 in iteration six . . . 59
4.17 The demand points assigned to facility 45 in iteration eight . . . 61
4.18 The demand points assigned to facility 17 in iteration thirteen . . . 62

C.1 Comparison of Model A's solutions with Model B's solutions . . . 78

List of Acronyms

FCLP	Fixed Charge Location Problem
FLP	Facility Location Problem
GIS	Geographic Information System
MCLP	Maximal Coverage Location Problem
MIP	Mixed Integer Program
MODM	Multi-Objective Decision Making
MOFLP	Multi-Objective Facility Location Problem
PCF	Primary Care Facility
PCLP	P-center Location Problem
PMLP	P-median Location Problem
SCLP	Set Covering Location Problem

Chapter 1

Introduction

1.1 Research Background

Public healthcare institutions such as hospitals and clinics play an important role in communities. Their locations and demand allocations are of importance to urban planning and development due to an increasing demand for healthcare as a result of population growth [24]. Sub-optimal facility location decisions have serious impacts on a community beyond simple cost and service metrics. Healthcare facilities which are inaccessible and inequitable in terms of travelling distances are likely to be associated with increased mortality and illness [1]. It is therefore necessary to determine the optimal number and location when establishing new facilities, such that objectives like accessibility, efficiency, equity and cost-efficiency are achieved [24].

A primary care facility (PCF) is a healthcare facility like a clinic or hospital that provides first-contact care such as early diagnosis, timely and effective treatment [1]. Primary healthcare is widely regarded as the backbone of a national healthcare system [18]. Access to primary healthcare is considered a fundamental human right and a key facilitator in the attainment of overall population healthcare outcomes [22]. The right to healthcare is provided for in three sections of the South African constitution. These provide for access to healthcare services including reproductive health, emergency services and basic healthcare for all citizens [9]. Furthermore the National Development Plan has as one of its goals to strengthen access to primary healthcare services and broaden district-based healthcare programmes [8].

In developing countries like South Africa, primary healthcare must be accessible to the vast majority of the population to be successful [21]. Poor access to primary care is often associated with adverse pregnancy outcomes, infant mortality, reduced immunisation and vaccination rates. Furthermore, inaccessibility to clinics may also affect adherence to treatment regiments for chronic diseases such as TB [21]. Primary care is considered as the first level of contact of individuals, the family and the community with the national health system [22]. Primary health care brings healthcare as close as possible to where people live and work and constitutes the first element of a continuing healthcare process [22]. In the majority of health systems across the world, people often seek care from their primary physicians before seeking care anywhere else [18]. It has been shown that health is better in areas with more primary care physicians and that people who receive care and treatment from primary care facilities are healthier [18].

This study will focus specifically on public PCF location problems. In managing the location allocation of public healthcare facilities, it is important to consider accessibility and equity [24]. Accessibility is achieved when the weighted distance between demand points and their assigned facilities is minimised. This is so that the total distance travelled by the entire population of patients is minimised such that there is geographic accessibility and efficiency in the utilisation of healthcare services [15]. Equity is achieved when the maximum distance between all demand points and their assigned facilities is minimised [24] or the total weight of demand coverage is maximised [7]. This is so that each patient population travels an equal or similar distance to access their assigned facility such that there is fairness and equity in travelling to their nearest healthcare facility [15]. The demand or weighted demand is defined as the size of the population or the number of people who require access to public healthcare services [1].

Three problem formulations form the basis of discrete facility location problems (FLPs) in this study.

- The p-median location problem (PMLP) locates p number of facilities at candidate points such that the sum of the weighted distance (or time) between demand points and facilities which serve them is minimised [2].
- The p-center location problem (PCLP) locates p number of facilities such that the maximum travel distance (or time) between all demand points and their assigned facility is minimised [11].
- The maximal coverage location problem (MCLP) locates facilities such that the weighted demand covered within a pre-specified maximum coverage distance is maximised [11].

The PMLP focuses on efficiency and accessibility in location allocation [10]. The PCLP focuses on providing an equitable solution with an objective function that minimises the maximum distance between demand nodes and the allocated facility [10]. The MCLP focuses on providing either an equitable or accessible solution based on its variation and use [7, 24]. Thus in choosing one model, a trade-off is made between accessibility, efficiency and equity. It is thus often the case that facility location problems particularly for public facilities are multi-objective.

1.2 Problem Statement

Single objective facility location models do not adequately consider the complexity of the public PCF location allocation problem nor do they take into account that these problems have multiple and often conflicting objectives that must be optimised. Multi-Objective facility location problems (MOFLPs) are required to solve location allocation problems with multiple objectives. A multi-objective optimisation approach is thus required to solve the public PCF location problem such that objectives such as accessibility and equity are achieved.

1.3 Research Questions and Objectives

1.3.1 Research Question

- The **primary research question** is: Can a multi-objective optimisation approach provide an efficient solution to the location allocation of a public PCF?
- The **secondary research question** is: What are the trade-offs between the accessibility and equity objectives?

1.3.2 Research Objectives

- To develop a multi-objective optimisation model to solve the public primary healthcare facility location problem.
- To investigate the trade-offs between the accessibility and equity objectives

1.4 Document Structure

The document is organised as follows. Chapter two provides a full literature review on MOFLPs; a critical survey of similar work, the modelling approaches and solution methods used to solve MOFLPs. Chapter three provides a conceptual design of the model developed, the method, data and tools used in developing the model, as well the model verification and validation. Chapter four provides the model results and analysis. Chapter five provides the conclusion of the study and the recommendation for future work.

Chapter 2

Literature Review

In this chapter, a literature survey is conducted. The purpose of this chapter is to provide a critical literature survey of the location allocation of the public PCF problem.

2.1 Discrete Facility Location Problems

Facility location is a branch of operations research concerning the siting and locating of facilities to serve a given demand in order to optimise (maximise or minimise) at least one objective function [11]. FLPs can be solved using optimal or heuristic models in order to attain efficient solutions.

There are two broad categories for discrete facility location problems. There are covering-based problem and median-based problems. Covering-based problems assume that demand locations need to fall within a pre-specified coverage distance (or time) from the facilities that serve them in order to be covered [1]. Median-based problems locate facilities at candidate points so as to minimise the weighted average distance (or time) between demand points and the facilities which serve them [1]. The PCLP and MCLP fall under the covering-based category and the PMLP and the fixed charge location problem (FCLP) fall under the median-based category [1].

The PMLP locates p facilities such that the sum of the demand-weighted distances between demand points and facilities that serve them is minimised [10]. Figure 2.1 illustrates the PMLP where there are two facilities (p=2) and ten demand points. The facilities are located in such a way that the demand-weighted sum of the network is minimised and thus minimising the quantity of demand-weight that has to travel to its assigned facility. The PMLP however, does not consider the inequity of travel distance that may exist between each demand point and the facility in the network.

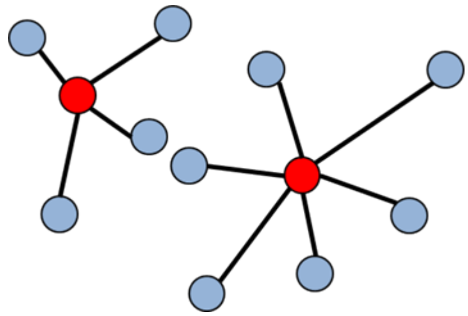

Figure 2.1: P-median Location Problem (PMLP)

The PCLP locates p facilities such that the maximum allowable distance between a demand point and the facility that serve it, is minimised [10]. Figure 2.2 illustrates the PCLP where there are three facilities (p=3) and six demand points. The facilities are located in such a way that each demand point falls within the maximum allowable distance range between its assigned facility. This ensures that there is equity of distance for all demand points and the facilities which serve them.

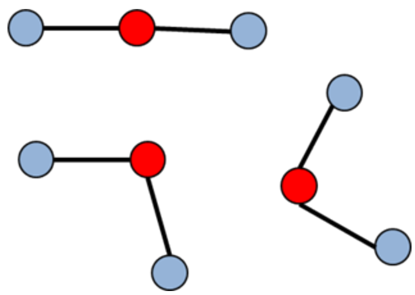

Figure 2.2: P-center Location Problem (PCLP)

The MCLP locates p facilities such that the weighted demand of all the points covered is maximised [10]. The demand points however must fall within a pre-specified distance range between the facility which serve them. Demand points that fall outside the range are considered not to be covered. Figure 2.3 illustrates the MCLP where there are two facilities (p=2), seven demand points covered and three demand points not covered. The MCLP can either ensure equity of distances or accessibility depending on how it is defined and its variation in formulation [7, 24].

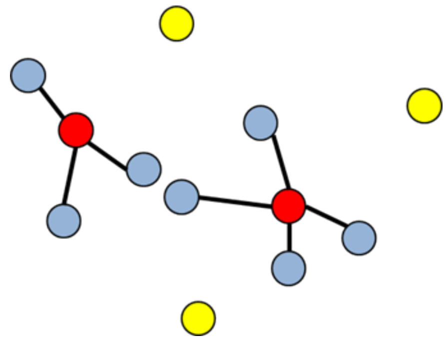

Figure 2.3: Maximal Coverage Location Problem (MCLP)

2.1.1 P-median Location Problem

The basic formulation of the PMLP [1] is as follows:

Sets:
I : The set of demand points
J : The set of candidate facilities

Input Parameters:
d_{ij} : The travel distance between demand point i $\in I$ and candidate facility j $\in J$
w_i : The weight of the demand at demand point i $\in I$
p : The number of facilities to be selected

Decision variables:
x_{ij} : 1, if demand i $\in I$ is assigned to candidate facility j $\in J$; 0, otherwise
y_j : 1, if candidate facility j $\in J$ is opened; 0, otherwise

Formulation:

$$\min z = \sum_{i\in I} \sum_{j\in J} w_i d_{ij} x_{ij} \tag{2.1}$$

Subject to:

$$\sum_{j\in J} y_j = p \tag{2.2}$$

$$\sum_{j\in J} x_{ij} = 1, \forall\ i \in I \tag{2.3}$$

$$x_{ij} \leq y_j, \forall\ i \in I \text{ and } j \in J \tag{2.4}$$

$$x_{ij} \geq 0, \forall\ i \in I \text{ and } j \in J \tag{2.5}$$

$$y_j \in \{0, 1\}, \forall\ j \in J \tag{2.6}$$

The objective function, Eq.(2.1) minimises the sum of the weighted distances between selected facilities and the demand points they serve. Eq.(2.2) ensures that only p amount of facilities are selected. Eq.(2.3) ensures that a demand point is assigned to only one facility. Eq.(2.4) ensures that demand is assigned to a selected facility. Eq.(2.5) and Eq.(2.6) ensure that the binary is adhered to.

2.1.2 P-center Location Problem

The basic formulation of the PCLP [1] is as follows:

Sets:
I : The set of demand points
J : The set of candidate facilities

Input Parameters:
d_{ij} : The travel distance between demand point i $\in I$ and candidate facility j $\in J$
w_i : The weight of the demand at demand point i $\in I$
p : The number of facilities to be opened

Decision variables:
x_{ij} : 1, if demand point i $\in I$ is assigned to facility j $\in J$; 0, otherwise
y_j : 1, if candidate facility j $\in J$ is opened; 0, otherwise
d_{max} : The maximum allowable distance between each demand point the facility which serves it

Formulation:

$$\min z = d_{max} \tag{2.7}$$

Subject to:

$$\textstyle\sum_{j \in J} y_j = p \tag{2.8}$$

$$\textstyle\sum_{j \in J} x_{ij} = 1, \forall\ i \in I \tag{2.9}$$

$$x_{ij} \leq y_j,\ \forall\ i \in I \text{ and } j \in J \tag{2.10}$$

$$d_{ij} x_{ij} \leq d_{max}, \forall\ i \in I \text{ and } j \in J \tag{2.11}$$

$$x_{ij}, y_j \in \{0, 1\}, \forall\ i \in I \text{ and } j \in J \tag{2.12}$$

The objective function, Eq.(2.7) minimises the maximum allowable distance between all demand points and the facilities which serve them. Eq.(2.8) ensures that only p amount of facilities are selected. Eq.(2.9) ensures that a demand point is assigned to one facility. Eq.(2.10) ensures that demand is assigned to a selected facility. Eq.(2.11) ensures that the distance between a demand point and its assigned facility is less than or equal to the maximum allowable distance. The maximum allowable distance is constrained such that the distance between demand points and their assigned facility falls within the distance range. Eq.(2.12) ensures the binary is adhered to.

2.1.3 Maximal Coverage Location Problem

The basic formulation of the MCLP [1] is as follows:

Sets:
I : The set of demand points
J : The set of candidate facilities
N : The set of candidate facilities which can cover the demand point [$j \in J$ given $d_{ij} \leq r$]

Input Parameters:
d_{ij} : The travel distance between demand point i $\in I$ and candidate facility j $\in J$
w_i : The weight of the demand at demand point i $\in I$
p : The number of facilities to be selected
r : The maximum allowable distance range between a demand point and the facility which serves it

Decision variables:
c_i : 1, if demand i $\in I$ is assigned to a selected facility; 0, otherwise
y_j : 1, if candidate facility j $\in J$ is selected; 0, otherwise

Formulation:

$$\max z = \sum_{i \in I} w_i c_i \tag{2.13}$$

Subject to:

$$\sum_{j \in J} y_j = p \tag{2.14}$$

$$\sum_{j \in N} y_j \geq c_i, \forall\ i \in I \tag{2.15}$$

$$c_i, y_j \in \{0, 1\}, \forall\ j \in J \tag{2.16}$$

The objective function, Eq.(2.13) ensures that the sum of the weighted demand points covered is maximised. Eq.(2.14) ensures that only p amount of facilities are selected. Eq.(2.15) that demand is assigned to a selected facility. The model can however choose not to assign a demand point to a facility if the demand point does not fall into the specified distance range. Eq.(2.16) ensures that the binary is adhered to.

2.2 Multi-Objective Decision Making

Multi-Objective decision making (MODM) or also referred to as multi-criteria decision making, is the evaluation of a set of possible courses of action or alternatives. The evaluation may result in the selection of a preferred alternative, ranking alternatives from best to worst or separating alternatives into categories. The set of alternatives may be explicitly defined and discrete in number or implicitly defined via constraints in a mathematical programming formulation or weightings in the objective function [15].

Many techniques have been used to solve MODM problems. Optimisation methods are widely used, while other methods include geographic information system (GIS) and simulation modelling. Simulation modelling is rarely used in literature and GIS is typically used in conjunction with other optimisation methods [11]. The most widely used optimisation methods are global criterion method, utility function, bounded objective method, sequential multi-objective problem solving, iterative goal programming, genetic algorithms and simulated annealing [11]. Classical approaches like utility function, bounded objective method and goal programming attempt to convert multi-objective problems into a single objective problem and optimise a new single objective problem. Pareto optimal approaches like iterative goal programming and sequential multi-objective optimisation produce a set of solutions. If the problems are too complex, then they are typically solved using an evolutionary algorithm like a genetic algorithm [11].

FLPs have historically been rooted in single objective formulation; however as the use of multi-objective techniques have become widely used in operations research, they have also been adopted in solving FLPs [11]. This is because decision makers have realised that some FLPs are too complex and thus require the consideration of more than one objective. This is particularly the case for the location optimisation problem for public facilities. Application areas for MOFLPs include location allocation problems for warehousing, waste disposal, street lighting, cell phone towers, WIFI hotspots and emergency and humanitarian relief efforts such as shelters, medical centers and water points [11].

2.2.1 Modelling Approaches

A multi-objective mixed integer program (MIP) is an approach that optimises more than one objective in a selection problem [16]. In contrast to a single objective optimisation problem the notion of one optimal solution is not the case in a multi-objective optimisation problem. A solution that proves best for one objective may rate poorer on another objective. Therefore multi-objective optimisation approaches should always render efficient points or Pareto optimal points. In most cases this means that if one objective is improved the other is deteriorated. These solutions are mapped on a Pareto frontier and thus the decision maker can visually see the trade-offs between improving one objective over another.

2.2.2 Multi-Objective Optimisation

The constraint method

The Pareto frontier is a plot of the ordered collection of the efficient points of a multi-objective model [16]. The set of points is constructed through iterative optimisations of the model. This can be done through the constraint method where one objective is optimised while the other objectives are constrained through a range of feasible values [15]. This is particularly suited to problems with only two objectives. This is achieved by enforcing new constraints each time the model is iterated for one objective while the other objective is optimised and is essentially treated like a single-objective optimisation problem [16].

The constraint method or ϵ-constraint method optimises the single most important objective function while the other objective functions are constrained such that they form the lower and upper bounds for the objective function [13]. The lower bound can be obsolete unless the intent is to achieve a goal or fall within a range of values, rather than to determine a minimum [13].

For example, assume the following MODM problem [14]:

$$\max z = (F_1(x), F_2(x), F_3(x), F_n(x)) \tag{2.17}$$

Subject to:

$$x \in S \tag{2.18}$$

where x is the vector of decision variables and $F_1(x), ..F_n(x)$ are the n number of objective functions and S is the feasible region. Using the constraint method the problem will be formulated as follows [14]:

$$\max z = F_1(x) \tag{2.19}$$

Subject to:

$$F_2(x) \geq \epsilon_2 \tag{2.20}$$

$$F_3(x) \geq \epsilon_3 \tag{2.21}$$

$$F_n(x) \geq \epsilon_n \tag{2.22}$$

$$x \in S \tag{2.23}$$

where x is the vector of decision variables, $F_1(x), ..F_n(x)$ are the n number of objective functions, $\epsilon_1, ..., \epsilon_n$ are the bounds for each constrained objective function and S is the feasible region.

The pre-emptive method

When a multi-objective problem has more than one constraint it is computationally impractical to construct a Pareto frontier. Thus, to obtain useful solutions, the problem is converted into a single-objective formulation using the pre-emptive method or the weights method [16].

The pre-emptive method performs multi-objective optimisation by considering objectives one at a time by order of importance [16]. The most important objective is optimised first; then the second objective and so on, subject to the preceding objective achieving its optimal value first. This method recognises that not all objectives are of priority and thus optimises each objective by priority [16].

The formulation [13] of a problem using this method is as follows:

$$\min z = F_i(x) \tag{2.24}$$

Subject to:

$$F_j(x) \leq F_j(x_j*), j = [1, 2, ..., i-1, i \geq 1] \tag{2.25}$$

$$i = [1, 2, ..., k] \tag{2.26}$$

$$x \in S \tag{2.27}$$

where i represents an objective function's position in the priority sequence and $F_j(x_j*)$ represents the optimum of the jth objective function found in the jth iteration.

The weights method

The weights method performs multi-objective optimisation by combining all the objectives into a single objective function to be optimised [16]. Positive weights are multiplied with objectives that are to be maximised and negative weights are multiplied with objectives that are to be minimised in a maximisation function, while the converse is done in a minimisation function [16]. The signs ensure that the objectives are in the same direction while the weights reflect the objective's relative importance or priority.

The formulation [13] of a problem using this method is as follows:

$$\max z = w_1F_1(x) + w_2F_2(x) + w_3F_3(x) + w_nF_n(x) \tag{2.28}$$

Subject to:

$$x \in S \tag{2.29}$$

where x is the vector of decision variables, $F_1(x), ..F_n(x)$ are the n number of objective functions, $w_1, ..., w_n$ are the weights of each function and S is the feasible region.

Goal Programming

Goal Programming is a multi-objective formulation that seeks to achieve a set of targets or goals [20]. Goal Programming allows decision makers to set target values that the objectives must achieve. For example decision makers would set a minimum target distance between demand point and open facilities. The objectives are therefore formulated as target or goals to which the solution must be as close as possible to the target. A disadvantage of goal programming is that it introduces another variable (the target/goals) that decision makers must set.

The goals in a goal programming model can be thought of as soft constraints. Soft constraints are the targets that the model may violate in solving for a feasible solution [16]. Once the goal is formulated as a soft constraint, it is then necessary to introduce a non-negative deficiency variable to the model. For a target that the decision maker wishes to exceed, the deficiency variable is written as an under achievement that must be added; and for a target that the decision maker wishes to minimize, the deficiency variable is written as an excess that needs to be subtracted. The objective function of the model then minimises the sum of the deficiency variables. Lastly non-negativity constraints are added such that the deficiency variables are not rendered negative by the model. The formulation [16] is written as follows:

Input Parameters:
o_1 : Objective 1 to be achieved
o_2 : Objective 2 to be achieved
o_3 : Objective 3 to be achieved
t_1 : Target 1 set
t_2 : Target 2 set
t_3 : Target 3 set

Decision variables:
d_1 : Deficiency variable 1
d_2 : Deficiency variable 2
d_3 : Deficiency variable 3

Formulation:

$$\min z = d_1 + d_2 + d_3 \tag{2.30}$$

Subject to:

$$o_1 - d_1 \leq t_1 \tag{2.31}$$

$$o_2 + d_2 \geq t_2 \tag{2.32}$$

$$o_3 + d_3 - d_3 = t_3 \tag{2.33}$$

$$d_1, d_2, d_3 \geq 0 \tag{2.34}$$

Goal Programs can also be formulated using the pre-emptive and weights method [19]. In the weights method the single objective function is the weighted sum of the functions representing the goals. In the pre-emptive method the goals are prioritized and solved in order of importance. The model then optimised the goals one at a time in order of priority [19].

2.2.3 Tools and Solution Methods

Mathematical formulations of MOFLPs such as goal programs and MIPs can be solved using optimisation softwares such as CPLEX, Lingo, Xpress or Gams [1]. Heuristic and metaheuristic methods can be solved by a tabu search, a genetic algorithm or simulated annealing. [1].

2.3 Multi-Objective Facility Location

A comprehensive survey of healthcare FLPs was conducted by Ahmadi-Javid et al [1]. The review provides a thorough classification of healthcare FLPs and surveys the literature on healthcare facility location in the last decade. The survey categorises healthcare facilities into non-emergency and emergency locations. Non-emergency facilities are defined as first-contact care facilities and specialised facilities like clinics, doctors offices and healthcare centers, blood banks and long-term nursing homes. Emergency facilities are defined as trauma centers, emergency centers, ambulance stations and temporary medical centers. Hospitals and clinics are classified as non-emergency facilities thus the literature on emergency healthcare facilities was negligible for this review. Ahmadi-Javid et al [1] classify hospitals that provide primary care or first-contact care as their main task under the non-emergency category. However if a hospital focuses on a combination of emergency and non-emergency tasks it was not classified as a PCF or under the non-emergency category.

Therefore this study will similarly focus on clinics and hospitals that provide primary care or first-contact care as their main task. The survey was useful because it classified current literature on PCF location problems by considering the solution method, modelling approach, basic model formulation, constraints, decision variables, objective function(s), input data and if the problem was multi-period and if it considered uncertainty. A summary of the literature surveyed is shown in Figure 2.4.

2.3.1 Related Works

Mitropoulos et al [15] determined the locations of health centers and hospitals in Greece using a multi-level bi-ojective model by considering two objectives. The two objectives were the minimisation of distance between patients and facilities, and the equitable distribution of facilities among citizens. The problem was formulated using a multi-level variation of the the PMLP to achieve the first objective and the classic PCLP for the second objective. Mitropoulos et al [15] argued that an efficient healthcare system is one that maximises the social welfare by locating healthcare facilities such that firstly the total travelling distance (and therefore travelling cost) is minimised and that secondly that there is equitable and fair distribution of facilities such that patients with the same health profile have equitable access to a facility.

The model firstly differentiated between hospitals and health centers and thus capacitated health centers in terms of the number of patients a health center was allowed to serve and uncapacitated hospitals. It then secondly considered patient preferences by first determining public preference between hospitals and health centers and thus estimated a parameter (α) that represented the patients preference. The parameter was then used as an input for bi-objective location allocation model. The model was solved by a MIP model using the constraint method with Xpress Solver to produce a set of efficient solutions. The solutions were graphed on a Pareto frontier to determine the trade-offs between the equity and efficiency objectives. The sets of solutions are further mapped using GIS software and compared to existing health care facilities in Greece. The study found that there is existed a trade-off between the accessibility and equity objective by constructing the Pareto frontier.

Burkey et al [7] examined the efficiency, service availabilty and equity objectives provided by hospitals in the United States. Their study compared existing locations with optimal locations by minimising the distance between patient demand nodes and hospital facilities as well as maximising the number of patients covered within a pre-specified distance. The first objective measured efficiency while the second objective measured service availability. The problem was formulated using the PMLP and the MCLP to achieve each objective respectively while the equity objective was formulated and measured in terms of a Lorenz curve. Burkey et al [7] argued that accessibility of service is measured in terms of both efficiency and service availabity; thus the distance a between patient and a facility and the expected coverage of the population within a given amount of distance or time.

This study is relevant as it combined two single-objective basic facility location problems into a multi-objective formulation in order to locate public PCF facilities in order to ensure accessibility in terms of efficiency and service availability. It did not however address the equity objective using an FLP nor did it address the trade-offs between the objectives. This study was different from the Mitropoulos et al, [15] study in the way it defined and formulated the objectives . Mitropoulos et al, [15] formulates access (efficiency) and equity using the PMLP and PCLP while Burkey et al [7] formulates efficiency, service availability and equity using the PMLP, MCLP and the Lorenz curve respectively. Furthermore Burkey et al [7] found that when the goal was improving service availability that an increase in availability resulted in reduced travel times for for the worst served however this also resulted in increasing times for the majority and thus decreasing efficiency. In contrast, when the goal was improving efficiency this resulted in increased availability.

Beheshiftar & Alimoahmmadi [4] determined optimal sites for new clinics by considering four objectives. The four objectives were minimising total travel cost, minimising inequity in access to clinics, minimising the land-use incompatibility in the study area and minimising costs of land acquisition. The problem was formulated using the PMLP. The problem is solved using a combination of GIS and a non-dominated genetic algorithm approach in order to produce a set of efficient solutions. This study was useful in identifying that a number of objectives can be used in solving location problems for clinics as well as illustrating that a set of solutions can be achieved by changing the variable p. The study did to some extent discuss the trade-offs between objectives.

Zhang et al [24] determined optimal locations for public healthcare facilities in Hong Kong by considering four objectives. The four objectives were maximising the accessibility for the entire population, minimising the inequity of accessibility, minimising the uncovered population and minimising the cost of building new facilities. The problem was formulated using the PCLP, fixed charge location problem (FCLP) and a variation of the MCLP. It was solved using a genetic algorithm to produce a set of efficient solutions. The study further evaluated the trade-offs between conflicting objectives.

Although the study done by Sahin et al [17] did not focus specifically on public PCFs but rather on the location of blood banks in Turkey, it was useful for this literature review because the location of blood banks is classified under the non-emergency category. Furthermore the study made use of a hierarchical PMLP and a set covering location problem (SCLP) to achieve the access and equity objectives respectively. The model was developed and solved as a MIP in CPLEX. The study however did not critically investigate the trade-offs between the objectives.

Similarly Bruni et al [6] located transplant centers (also classified as non-emergency facilities) in Italy. The study made use of the PMLP to achieve the efficiency objective and set a maximum distance as a constraint in the model between a demand node and a facility in order to achieve the equity objective. The model was developed and solved as a MIP in Lingo. This study too, did not investigate the trade-offs between objectives.

Similarly, the study done by Karatas & Yakici, [12] did not focus specifically on a public PCFs, it combined the objectives of the PMLP, PCLP and MCLP to develop a methodology for solving a MOFLP for public emergency service stations. The study found a set of efficient solutions using a combination of a branch and bound method and an iterative goal programming algorithm. Wichapa & Khokhajaikiat, [23] used fuzzy analytical hierarchy process and goal programming to determine optimal locations for infectious waste disposal facilities.

Authors	Facilities	Objectives	Discrete FLP used	Modelling Approach	Solution Method	Trade-offs evaluated
Mitropoulos et al	Hospitals and health centres	Access Equity	Multi-level PMLP PCLP	Mixed Integer Program	Optimisation software	Pareto frontier
Burkey et al	Hospitals	Efficiency Service availability Equity	PMLP MCLP	Mixed integer program	Optimisation software	-
Beheshiftar & Alimoahmmadi	Clinics	Equity Access Land-use Land cost	PMLP	Mixed Integer Program	Non-dominated genetic algorithm	Set of efficient solutions
Sahin et al	Blood banks	Access Equity	Multi-level PMLP SCLP	Mixed Integer Program	Optimisation software	-
Bruni et al	Transplant centres	Efficiency Equity	PMLP	Mixed Integer Program	Optimisation software	-
Zhang et al	Public healthcare facilities	Equity Access	PCLP MCLP FCLP	-	Genetic algorithm	Set of efficient solutions
Karatas & Yakici	Emergency fire stations	Access Efficiency Equity	PMLP MCLP PCLP	Branch and Bound, Goal Programming	Optimisation software	-

Figure 2.4: Summary of Related Works

In problems of locating public healthcare institutions, various objectives have been considered. It is clear that the majority of the literature reviewed agree that in locating PCFs there is a trade-off between the accessibility, efficiency, equity and cost objectives. The most crucial objective is firstly that of accessibility and efficiency [7, 15]. The differences are in the definitions of accessibility and efficiency. Some papers differentiate between efficiency and accessibility and some papers assume that accessibility is equal to efficiency. Mitropoulos et al, [15] consider accesibility and efficiency to be the same because geographic access to healthcare services correlate with utilisation of these services. While Burkey et al [7] consider accessibility to be equal to efficiency and service availabity thus minimising the total distance and maximising coverage. This difference in assumption determines how the problem is formulated and which basic discrete facility location problem is used to formulate the objectives. The PMLP is most commonly used to formulate accessibility and efficiency. Secondly the recognition of improving the equity of access has grown recently in literature [4, 7, 15, 24]. The PCLP is used to formulate the equity objective. Lastly reducing cost (formulated using the FCLP), increasing flexibility in service location and increasing the number of people covered by atleast one facility within an acceptable travel distance has been used in literature as objectives [4, 5, 7, 24].

Exact solutions methods used to solve MOFLPs for public PCFs include MIPs and goal programs while heuristic solutions include non-dominated genetic algorithms [11]. Some models have been iterated using the constraint, branch and bound and bounded objective methods to produced a set of pareto efficient solutions for which decision makers can choose from while others have only produced a single solution. Producing a set of efficient solutions has allowed decision makers to investigate the trade-offs between objectives by mapping Pareto frontiers and thus determining the effects of choosing one solution over another.

Literature shows that most models have been developed using MIPs or goal programs and solved with optimisation softwares. Mixed integer programming along with the constraint method provides a more suitable approach compared to goal programming because goal programming requires additional variables; the targets to be achieved. Furthermore the constraint method allows decision makers to repeatedly iterate the model through a range of feasible values to find the efficient points and thus construct a Pareto frontier and investigate the trade-offs between the objectives.

2.4 Conclusion

The literature review provided a critical review of MOFLPs for public PCF facilities by first surveying literature on single-objective FLPs and MODM. It was concluded that the problem was essentially a MOFLP. Thereafter a thorough literature review was conducted on MOFLPs for public PCFs. Literature showed the most common objectives were that of accessibility, efficiency and equity. These were modelled using the PMLP, PCLP and/or MCLP respectively. Literature showed that these problems could be modelled and solved using MIPs or Goal Programs in optimisation softwares or using evolutionary algorithms like genetic algorithms. This study therefore draws on literature to solve the public PCF problem by modelling the access and equity using the PMLP, PCLP and MCLP respectively.

Chapter 3

Method

3.1 Introduction

In this chapter, the development of the model is discussed. The purpose of this chapter is to provide a conceptual design of the model; the data, methods of verification and validation and the solutions and tools used to develop the model.

The development of the model was divided into two main phases. The first phase was developing a model in which the access and equity objectives were formulated using the PMLP and PCLP respectively. The second phase was developing a model in which the access and coverage objectives were formulated using the PMLP and MCLP respectively. Each model was then run using data from the JE Beasley OR Library [3]

3.2 Conceptual Design

The purpose of this study was to develop a multi-objective optimisation approach for the FLP of a public PCF like a clinic or hospital. The objectives of the model were accessibility or access (used interchangeably) and equity.

The accessibility objective was thus assumed to be equal to efficiency since geographic accessibility results in higher utilisation of the services [15]. This is because when a reduced demand population has to travel to their assigned facility, the total travelling distance (or travelling time and cost) is reduced and patients are likely to utilise the services [15]. Thus accessibility was defined as the minimisation of the sum of the weighted distances between demand points and the facilities which serve them and was formulated using the PMLP.

The equity objective was chosen to ensure that all patients travelled distances that are the the same or less between their demand point and their assigned facility. This objective was thus defined as the minimisation of the maximum allowable distance between a demand point and its assigned facility and was formulated using the PCLP.

The study conducted by Burkey et al [7] provided important points to consider for locating public PCF facilities and thus a second model was developed to explore coverage and how coverage interacts with accessibility and equity as defined in the first model. Burkey et al [7] defined accessibility as efficiency and service availability where efficiency was formulated using the PMLP and service availability was formulated using the MCLP.

A model was developed where these objectives were investigated. However the first objective was defined as accessibility or access (the minimisation of the sum of the weighted distance between demand points and the facilities which serve them) while the second objective was defined as coverage (the maximisation of the weighted coverage of demand points covered within a pre-specified distance range) in order to compare critically with the first model. Lastly since the equity objective is encompassed in the coverage definition since coverage is defined as the population of demand points covered within a maximum allowable distance range, there was be no need to use the PCLP to explicitly define it the model.

The models were then formulated and solved using the constraint method. This method firstly allows decision makers to iteratively produce a set of efficient solutions such that these solutions can be mapped on a Pareto frontier. The study by Mitropoulos et al [15] makes use of the constraint method to effectively construct a Pareto frontier and assess the trade-offs between the accessibility and equity objectives. Secondly the constraint method is suitable for problems where there are only two objectives to be achieved. Thus in this method was appropriate for this study since two objectives were investigated and compared with each other by mapping the solutions on a Pareto frontier.

3.2.1 Model A: Accessibility versus Equity

The formulation of Model A derived from Ahmadi et al [1] and Mitropoulos et al [15] is as follows:

Sets:
I : The set of demand points
J : The set of candidate facilities

Input Parameters:
d_{ij} : The travel distance between demand point i $\in I$ and candidate facility j $\in J$
w_i : The weight of the demand at demand point i $\in I$
p : The number of facilities to be selected
$d_{allowable}$: The allowable distance between each demand point and its assigned facility
n : The number of demand points

Decision variables:
x_{ij} : 1, if demand i $\in I$ is assigned to candidate facility j $\in J$; 0, otherwise
y_j : 1, if candidate facility j $\in J$ is selected; 0, otherwise
d_{max} : The maximum distance between each demand point and its assigned facility

Formulation:

Objective Function: Access

$$\min z = \sum_{i \in I} \sum_{j \in J} w_i d_{ij} x_{ij} \tag{3.1}$$

Objective function 3.1 minimises the weighted sum of the distances between demand points and the facilities which serve them.

Objective Function: Equity

$$\min z = d_{max} \tag{3.2}$$

Objective function 3.2 minimises the maximum allowable distance between a demand point and the facility which serves it.

Combined Objective Function

$$\min z = \frac{1}{n} \sum_{i \in I} \sum_{j \in J} w_i d_{ij} x_{ij} + d_{max} \tag{3.3}$$

Since the first two objective functions are minimised it is thus possible to combine them into a single objective such that a single optimal solution is sought by the model. However since the accessibility objective is significantly larger than the equity objective in terms of distance value, it is thus necessary to introduce a weight so as to balance out the objective function. Thus the accessibility objective is divided by the number of demand points (n).

Subject to:

$$\sum_{j \in J} y_j \leq p \tag{3.4}$$

$$\sum_{j \in J} x_{ij} = 1, \forall\ i \in I \tag{3.5}$$

$$x_{ij} \leq y_j,\ \forall\ i \in I \text{ and } j \in J \tag{3.6}$$

$$d_{ij} x_{ij} \leq d_{allowable}, \forall\ i \in I \text{ and } j \in J \tag{3.7}$$

$$d_{max} \leq d_{allowable} \tag{3.8}$$

$$x_{ij} \in \{0, 1\}, \forall\ i \in I \text{ and } j \in J \tag{3.9}$$

$$y_j \in \{0, 1\},\ j \in J \tag{3.10}$$

Eq.(3.4) ensures that only p amount of facilities are selected. Eq.(3.5) ensures that a demand point is assigned to only one facility. Eq.(3.6) ensures that a demand point can only be assigned to a facility if it is selected. Eq.(3.7) ensures that the distance between a demand point and its facility is less than or equal to the maximum allowable distance. Eq.(3.9) and Eq.(3.10) ensure the binaries are adhered to.

3.2.2 Model B: Accessibility versus Coverage

The formulation of Model B derived from Ahmadi et al [1] and Burkey et al [7] is as follows:

Sets:
I : The set of demand points
J : The set of candidate facilities
N : The set of candidate facilities which can cover the demand points [$j \in J$ given $d_{ij} \leq r$]

Input Parameters:
d_{ij} : The travel distance between demand point i $\in I$ and candidate facility j $\in J$
w_i : The weight of the demand at demand point i $\in I$
r : The maximum allowable distance range between a facility and the demand points it serves
p : The number of facilities to be selected

Decision variables:
x_{ij} : 1, if demand i $\in I$ is assigned to candidate facility j $\in J$; 0, otherwise
y_j : 1, if candidate facility j $\in J$ is selected; 0, otherwise
c_i : 1, if demand i $\in I$ is covered by a selected facility; 0, otherwise

Formulation:

Objective Function: Access

$$\min z = \sum_{i \in I} \sum_{j \in J} w_i d_{ij} x_{ij} \tag{3.11}$$

Objective function 3.11 minimises the weighted sum of the distances between the demand points and the facilities which serve them.

Objective Function: Equity

$$\max z = \sum_{i \in I} w_i c_i \tag{3.12}$$

Objective function 3.12 maximises the sum of the demand points covered by the candidate facilities in the maximum allowable distance range.

Since one objective function is a minimised and the other is a maximised, it is therefore not possible to combine them into a single objective function to obtain a single optimal solution.

Subject to:

$$\sum_{j \in J} y_j \leq p \tag{3.13}$$

$$\sum_{j \in J} x_{ij} \leq 1, \forall\ i \in I \tag{3.14}$$

$$x_{ij} \leq y_j,\ \forall\ i \in I \text{ and } j \in J \tag{3.15}$$

$$\sum_{j \in N} y_j \geq c_i, \forall\ i \in I \tag{3.16}$$

$$x_{ij} \in \{0, 1\}, \forall\ i \in I \text{ and } j \in J \tag{3.17}$$

$$y_j \in \{0, 1\},\ j \in J \tag{3.18}$$

$$c_i \in \{0, 1\}, \forall\ i \in I \tag{3.19}$$

Eq.(3.13) ensures that only p amount of facilities are selected. Eq.(3.14) ensures that a demand point is assigned to only one facility. Eq.(3.15) ensures that a demand point can only be assigned to a facility if it is selected. Eq.(3.16) ensures that a demand point is assigned to a selected facility in the maximum allowable distance range. Eq.(3.17), Eq.(3.18) and Eq.(3.19) ensure the binaries are adhered to.

3.3 Data Collection and Analysis

The JE Beasley OR library [3] provided datasets for uncapacitated and capacitated PMLPs. Ten datasets from the capacitated p-median text file were used for the study. The selected datasets were named DS1 to DS7 and DS11 to DS13 to correspond with the numbering in the JE Beasley OR library [3].

These datasets were used because they included the demand variables which were necessary for this study that the uncapacitated datasets did not have. The capacity and demand variables were included in the validation phase of the model. However the capacity variables were ignored in the full model tests and iterations due to the nature of the public PCF problem not taking into account capacity.

For Model A, all ten datasets were assessed while for the Model B only datasets DS1 to DS7 were assessed. The data was in the format of a text file with the first line of each row being; the dataset number and the the best solution found while the second row; the number of points, the median and the capacity of each median. The following rows were the number of the point, its associated x and y coordinate and its demand weight.

Each file was read into a python program such that the distances between each of the points could be calculated to form a distance matrix. Lastly the distance matrix and the demand variables for each point was exported to an excel file for ease of use. The python script can be found in Appendix A.

A table summarising the datasets in Figures 3.1 and 3.2 shows each dataset, the number of points, the medians and the $d_{allowable}$ value used for each iteration for each model.

Table 3.1: The data used in Model A

Dataset name	Data points	Medians	$d_{allowable}$ tested
DS1	50	5	≥0 ,120,100,80,60,45,40,35,33,30
DS2	50	5	≥0 ,120,100,80,60,50,40,35,33,31
DS3	50	5	≥0 ,120,100,80,60,40,35,33,30,29,28,27
DS4	50	5	≥0 ,120,100,80,60,40,35,33,32
DS5	50	5	≥0 ,120,100,80,60,50,40,35,33,30,29,28,27
DS6	50	5	≥0 ,120,100,80,60,50,40,35,33,30,29,28
DS7	50	5	≥0 ,120,100,80,60,50,40,35,33, 31,30
DS11	100	10	≥0 ,120,100,80,60,50,45,40,35,30,29,28,26,25,20,19
DS12	100	10	≥0 ,120,100,80,60,50,45,40,35,30,25,22,19
DS13	100	10	≥0 ,120,100,80,60,50,45,40,35,30,25,22,20,19

Table 3.2: The data used in Model B

Dataset name	Data points	Medians	r value tested
DS1	50	5	29,28,27,26,25,24,23,22,21,20,19,18,17
DS2	50	5	30,29,28,27,26,24,23,22,21,20,19,17
DS3	50	5	26,25,24,23,22,21,20
DS4	50	5	31,30,29,28,27,26,25,24,23,22,21,20,19,18,17
DS5	50	5	26,25,24,23,22,21,20,19,18,17
DS6	50	5	27,26,25,24,23,22,21,20,19,18,17
DS7	50	5	29,28,27,26,25,24,23,22,21,20,19,18,17

3.4 Validation and Verification

3.4.1 Model A: Accessibility versus Equity

A test model was developed in which only the first five data points from the DS1 were run. The median (p value) was set to two and $d_{allowable}$ was constrained to be greater than 0.

The output of the model was checked to ensure that:

- Only two facilities were selected
- Each demand point was assigned to one facility

- A demand point was assigned to an open/selected facility
- d_{max} was greater than zero
- All the binary variables were adhered to.

The results of the test model shown in Tables 3.3 and 3.4 show that the constraints are adhered to and thus the model is verified. The Lingo script and a screenshot of the excel spreadsheet showing the output of the test model can also be found in Appendix B.

Table 3.3: The facilities selected by the test model

Facilities	Value
1	1
2	0
3	0
4	1
5	0

Table 3.4: The assignment results of the test model

Demand/Facilities	1	2	3	4	5
1	1	0	0	0	0
2	0	0	0	1	0
3	1	0	0	0	0
4	0	0	0	1	0
5	0	0	0	1	0

The model was validated by comparing the best solution found by the model with the solutions given by the JE Beasley OR library [3]. In running the model, $d_{allowable}$ was negligible. The Lingo script can be found in Appendix B and the results of the solutions for each dataset are shown in Table 3.5. The small differences in solutions from some of the datasets are as a result of rounding off differences when the distance matrices were calculated. The distance matrices used for the model were rounded to the nearest integer.

Table 3.5: Validated model solutions

Dataset	Model Solution	Data points	Medians
DS1	706	50	5
DS2	740	50	5
DS3	735	50	5
DS4	650	50	5
DS5	661	50	5
DS6	769	50	5
DS7	761	50	5
DS11	996	100	10
DS12	962	100	10
DS13	1013	100	10

3.4.2 Model B: Accesibility versus Coverage

The model was verified by checking that:

- Only p amount of facilities were selected
- A demand point was considered covered if it was assigned to a selected facility
- A demand point was assigned to only one facility.

The model was verified using DS1. The test model's lingo script can be found in Appendix C. The model was validated by comparing its solutions with Model A's solutions. Since this model had one objective (coverage) that was different to that of the first model; the weighted coverage was maximised and the weighted distance was constrained or set to the values found in the first model solutions. In order to ensure model validation, the model solution thus had to cover all demand points, open the same facilities and assign the same demand points to selected facilities as Model A when the same weighted distance and $d_{allowable}$ (or in this model, r value) values were set.

Table C.1 in Appendix C shows that for the same weighted distance and distance values, the first and second model selected the same facility. Furthermore it must be highlighted second model's weighted coverage result is equal to the sum of the total demand weight for each dataset. This should be the case, since every demand point is assigned to a facility and is therefore considered covered.

3.5 Method

3.5.1 Model A: Accessibility versus Equity

This model was run using the constraint method. The first iteration was run such that $d_{allowable}$ was constrained to be greater than zero and the weighted sum of the distances between the facilities and the demand points was minimised. Thereafter $d_{allowable}$ was reduced with each model iteration until a feasible solution could no longer be found. The model was run using the selected JE Beasley OR Library data. The lingo script for each dataset can be found in Appendix D.

3.5.2 Model B: Accesibility versus Coverage

This model was run using the constraint method. The model was run by maximising the coverage and then constraining the weighted distance and maximum allowable distance distance range ($d_{allowable}$) through a range of feasible values for each iteration. For this model $d_{allowable}$ was named r.

Since the weighted coverage was maximised, and the goal was to see how coverage changes when r and the weighted distance is constrained this model was not first iterated by letting r be greater than equal to zero (like in the previous model) but rather was constrained to be less than equal to the last feasible $d_{allowable}$ value found in the first model. This is because this model would always yield the maximum coverage value (which is equal to the sum of the weighted demand for each dataset) for all feasible $d_{allowable}$ values of the previous model since all demand points were assigned to a facility in the previous model and would thus be considered covered.

For example the last feasible value for $d_{allowable}$ in Model A for DS2 was 31. Therefore in Model B, the model was first iterated by constraining r to be less than or equal to 30. Secondly, the weighted distance was constrained to be greater than zero. Once a solution was found the model was then iterated again such that the weighted distance was constrained to be less than equal to the last achieved value for the same weighted coverage and r value. This was done so that the model could find a lower solution for the weighted distance value if it existed. For example, in dataset two, in the first iteration r was constrained to be less than equal to 30 and the weighted distance constrained to be greater than zero. The model found a solution where the weighted distance was 11099 and the weighted coverage was 497. The model was iterated again such that r was still constrained to be less than equal to 30 but the weighted distance was constrained to be less than 11099. The solution that was thus found by the model was 4023 and 497 for the weighted distance and the weighted coverage respectively. The Lingo script as it was run for each dataset can be found in Appendix E

3.6 Scope

The assumptions in developing the model were as follows:

- the facilities were uncapacitated in terms of the number of patients served
- the facilities were not multi-level therefore there were no referrals
- the regions studied were assumed to have population with an identical health profile, and
- the demand was homogeneous

3.7 Tools

The data analysis and the converting of the data points into the distance matrices was run in Python. The models were developed and run in Lingo17 on a laptop with an Intel i5 processor running at 1.70 GHz using 4.00 MB of RAM running on 64-bit Windows 10 operating system.

Chapter 4

Results and Analysis

4.1 Introduction

This section provides the results and analysis of the models. The first section provides the results for selected datasets for each model. The second section provides an in-depth analysis of the results using selected datasets as examples to demonstrate observations about facility changes, assignment changes, coverage changes. Lastly, accessibility versus equity and the Pareto efficiency of each model are discussed.

4.2 Results

4.2.1 Model A: Accessibility versus Equity

Figures 4.1 and 4.2 show the model results for each iteration or time the model was run through a range of feasible values. Access represents the sum of the weighted distances between the facilities and the demand points they serve. Equity represents the maximum allowable distance between an open facility and the demand point it serves.

The model results show that firstly, as d_{max} decreased then the weighted sum of the distances increased. This means that when equity increased (by way of d_{max} decreasing) then access decreased (by way of the sum of the weighted distances between demand points and facilities increasing).

Secondly, the first few iterations of the model were Pareto inefficient. Model results are said to be Pareto efficient if by improving one of the objective function results in the deterioration of the other objective function [16]. Thus, Figures 4.1 and 4.2 show that the first few the solutions were Pareto inefficient. The results became Pareto efficient once the equity values became smaller. This was because, for example, when $d_{allowable}$ was constrained to be less than or equal to 100 the model stopped running once that constraint was met even if a value existed that was much lower for the same access objective. It was therefore necessary to iterate the model multiple times through a range of feasible values such that the Pareto efficient solutions were determined.

	DS1		DS2		DS3		DS4		DS5	
	Access	**Equity**	**Access**	**Equity**	**Access**	**Equity**	**Access**	**Equity**	**Access**	**Equity**
Iteration 1	1145	120	1713	131	995	119	993	122	1573	116
Iteration 2	1145	100	1713	100	995	100	993	100	1573	100
Iteration 3	1145	80	1713	80	995	80	993	80	1573	80
Iteration 4	1145	60	1723	60	995	60	995	60	1582	60
Iteration 5	1145	45	1752	50	1047	40	1081	50	1587	50
Iteration 6	1524	40	1820	40	1072	35	1800	40	1913	40
Iteration 7	1762	35	2705	35	1896	33	3461	35	2484	35
Iteration 8	2782	33	3515	33	2807	30	4283	33	2549	33
Iteration 9	4386	30	6323	31	2807	29	4871	32	2950	31
Iteration 10					2807	28			4389	30
Iteration 11					5178	27			4426	29
Iteration 12									5266	28
Iteration 13									5414	27
Iteration 14										
Iteration 15										
Combined	**63.94**	**41**	**37.34**	**38**	**21.44**	**35**	**32.28**	**42**	**22.66**	**46**

Figure 4.1: Model A Results for DS1 to DS5

	DS6		DS7		DS11		DS12		DS13	
	Access	**Equity**	**Access**	**Equity**	**Access**	**Equity**	**Access**	**Equity**	**Access**	**Equity**
Iteration 1	1601	115	1999	120	1759	123	1531	126	2187	131
Iteration 2	1601	100	1999	80	1759	100	1531	100	2187	100
Iteration 3	1601	80	1999	60	1759	80	1531	80	2187	80
Iteration 4	1617	70	2058	50	1759	60	1531	60	2187	60
Iteration 5	1645	60	2259	40	1759	50	1531	50	2213	50
Iteration 6	1857	50	4424	35	1759	45	1531	45	2223	45
Iteration 7	2171	40	4728	33	1808	40	1531	40	2264	40
Iteration 8	7802	30	4894	31	1890	35	1540	35	2292	35
Iteration 9	7802	29	7844	30	1979	30	1576	30	2588	30
Iteration 10	7966	28			1990	29	1720	25	2930	25
Iteration 11					2001	28	2821	22	3687	22
Iteration 12					2024	26	6305	20	4319	20
Iteration 13					2038	25	8817	19	7941	19
Iteration 14					5612	20				
Iteration 15					7974	19				
Combined	**46.78**	**36**	**46.08**	**37**	**20.38**	**25**	**15.78**	**26**	**26.74**	**26**

Figure 4.2: Model A Results for DS6 to DS13

Figures 4.1 and 4.2 were then constructed into graphs to determine the Pareto frontiers for each dataset. Figures 4.3 to 4.6 show the Pareto graphs for selected datasets while the rest of the Pareto graphs can be found in Appendix F. The Pareto frontiers are essentially the collection of all the efficient solutions mapped on a graph. Solutions below or above the frontier are considered inefficient while those on the frontier are considered efficient.

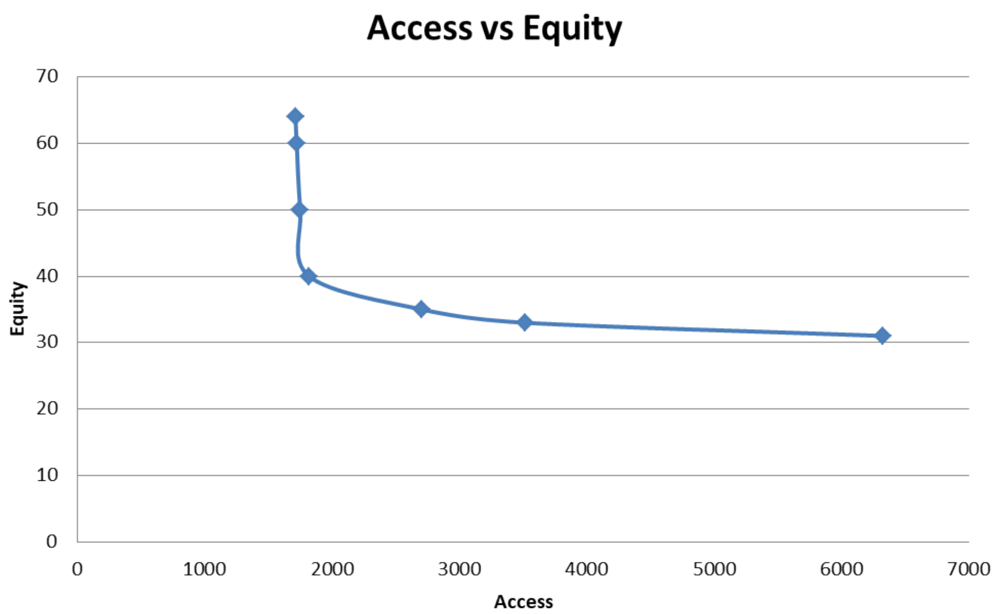

Figure 4.3: Graph showing the Pareto frontier for DS2

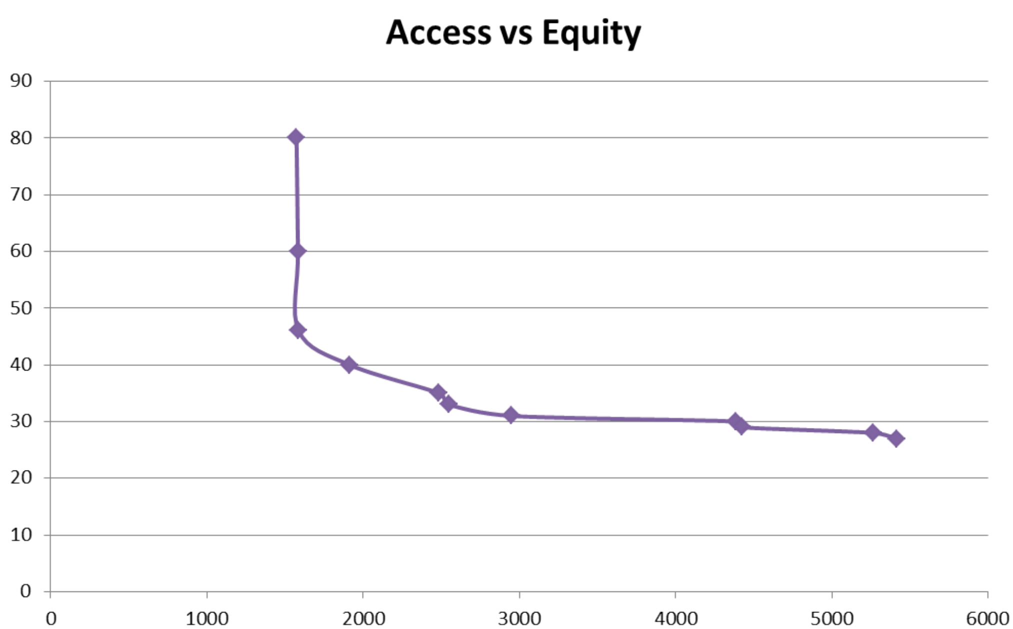

Figure 4.4: Graph showing the Pareto frontier for DS5

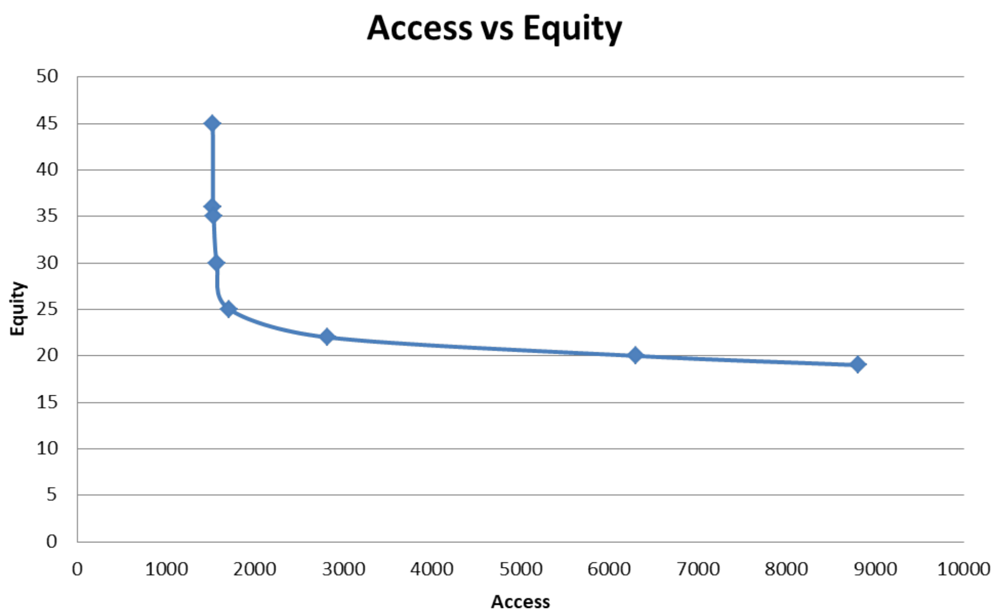

Figure 4.5: Graph showing the Pareto frontier for DS12

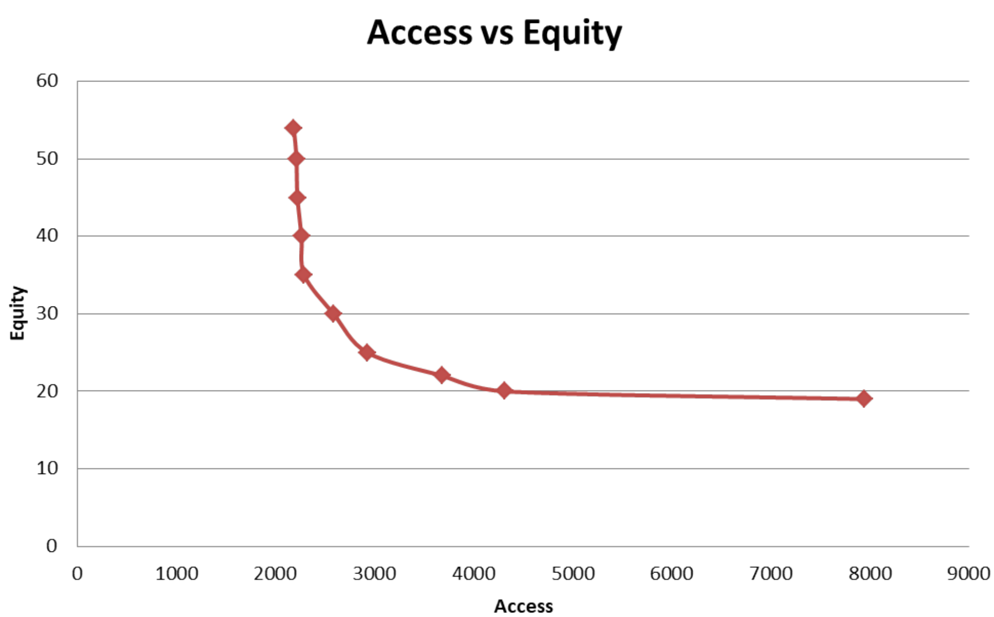

Figure 4.6: Graph showing the Pareto frontier for DS13

Thirdly, the model results show that as the number of data points increased the range of feasible values and therefore feasible solutions increased. This is shown by comparing DS1 to DS7 with DS11 to DS13 which have 50 and 100 data points respectively. It was observed that the smallest equity values were 29 and 19 respectively. This means that the model could search through a larger distance matrix to find solutions that were feasible. Furthermore it was observed that small changes in the $d_{allowable}$ values in DS11, DS12 and DS13 resulted in Pareto efficient solutions. The results for each dataset can be found in figures G.1 to G.6 in Appendix G.

Iteration	Access	Equity	Facilities opened
Iteration 1	1713	131	4,29,37,39,44
Iteration 2	1713	100	4,29,37,39,44
Iteration 3	1713	80	4,29,37,39,44
Iteration 4	1723	60	4,29,37,39,44
Iteration 5	1752	50	4,29,37,39,44
Iteration 6	1820	40	4,37,39,44,46
Iteration 7	2705	35	4,19,26,37,39
Iteration 8	3515	33	11,19,26,37,39
Iteration 9	6323	31	24,29,32,39,46
Combined	37.34	38	4,23,37,39,44

Figure 4.7: Model A's Results for DS2

Iteration	Access	Equity	Facilities opened
Iteration 1	1573	116	6,9,29,42,43
Iteration 2	1573	100	6,9,29,42,43
Iteration 3	1573	80	6,9,29,42,43
Iteration 4	1582	60	6,9,29,43,49
Iteration 5	1587	50	6,9,29,42,43
Iteration 6	1913	40	6,9,29,31, 43
Iteration 7	2484	35	13,24,29,35,43
Iteration 8	2549	33	13,24,29,35,43
Iteration 9	2950	31	13,24,29,36,43
Iteration 10	4389	30	12,15,32,34,35
Iteration 11	4426	29	15,24,30,32,35
Iteration 12	5266	28	15,24,30,35,38
Iteration 13	5414	27	15,24,30,36,41
Combined	22.66	46	10,14,30,36,38

Figure 4.8: Model A's Results for DS5

Iteration	Access	Equity	Facilities opened
Iteration 1	1531	126	1,5,28,44,53,65,66,68,75,77
Iteration 2	1531	100	1,5,28,44,53,65,66,68,75,77
Iteration 3	1531	80	1,5,28,44,53,65,66,68,75,77
Iteration 4	1531	60	1,5,28,44,53,65,66,68,75,77
Iteration 5	1531	50	1,5,28,44,53,65,66,68,75,77
Iteration 6	1531	45	1,5,28,44,53,65,66,68,75,77
Iteration 7	1531	40	1,5,28,44,53,65,66,68,75,77
Iteration 8	1540	35	1,5,28,34,53,62,65,68,75,77
Iteration 9	1576	30	1,5,34,53,65,68,75,77,97,98
Iteration 10	1720	25	1,5,28,52,53,65,68,75,77,98
Iteration 11	2821	22	1,5,34,49,52,53,66,74,77,87
Iteration 12	6305	20	5,13,27,34,49,52,71,74,77,87
Iteration 13	8817	19	13,27,34,50,52,71,74,77,80,87
Combined	15.78	26	1,5,28,53,62,65,66,68,75,77

Figure 4.9: Model A's Results for DS12

Iteration	Access	Equity	Facilities opened
Iteration 1	**2187**	**131**	**10,23,46,49,61,64,68,71,82,99**
Iteration 2	2187	100	10,23,46,49,61,64,68,71,82,99
Iteration 3	2187	80	10,23,46,49,61,64,68,71,82,99
Iteration 4	2187	60	10,23,46,49,61,64,68,71,82,99
Iteration 5	2213	50	10,23,49,61,64,68,71,82,99,100
Iteration 6	2223	45	10,23,49,61,64,68,71,82,99,100
Iteration 7	2264	40	10,23,49,61,62,64,68,71,99,100
Iteration 8	2292	35	10,23,49,61,62,64,68,71,99,100
Iteration 9	2588	30	10,23,34,49,62,64,68,71,99,100
Iteration 10	2930	25	10,23,49,61,62,64,68,71,86,99
Iteration 11	3687	22	23,49,62,64,68,71,86,91,99
Iteration 12	4319	20	23,64,68,70,71,83,86,87,91,100
Iteration 13	7941	19	1,7,20,59,64,82,83,90,99,100
Combined	26.74	26	10,23,46,49,61,64,68,71,82,99

Figure 4.10: Model A's Results for DS13

4.2.2 Model B: Accessibility versus Coverage

The results for this model for each dataset are shown in the Figures 4.11 to 4.15. Figure 4.11 shows the model solutions for DS1. For this dataset from iterations one and four when the r value and the sum of weighted coverage decreased, the sum of the weighted distances increased. This means as equity increased (by way of the r value decreasing), access decreased (by way of the sum of the weighted distance increasing). However from iteration four to thirteen, when the r value and the sum of the weighted coverage decreased, the sum of the weighted distance decreased. This means that when equity increased, access increased for only those demand points covered. This was because as the number of demand points not covered increased (and therefore both access and equity decreased for these points) the model was able to optimise both access and equity for the demand points that fall within the maximum allowable distance range of coverage.

Iteration	"r" value	Access	Coverage	Facilities opened	Demand points not covered
Iteration 1	29	6793	485	1,4,16,43,45	46
Iteration 2	28	6219	485	1,16,24, 43, 45	46
Iteration 3	27	9270	480	1,5,12,45,48	25,46
Iteration 4	26	8998	480	4,26,40,44,45	25,46
Iteration 5	25	7590	472	22,30,40,42,44	16,34
Iteration 6	24	7197	471	19,40,44,45,48	25,46,49
Iteration 7	23	6447	470	12,19,44,45,48	3,25,46,9
Iteration 8	22	6263	450	12,19,44,45,48	3,25,42,46,49
Iteration 9	21	7611	435	19,21,35,45,48	3,6,14,25,41,42,46,49
Iteration 10	20	6927	425	12,19,21,45,48	3,6,11,14,25,41,42,46,49
Iteration 11	19	6927	425	12,19,21,45,48	3,6,11,14,25,41,42,46,49
Iteration 12	18	5911	399	12,18,19,45,48	3,6,8,11,14,25,30,32,41,42,46,49
Iteration 13	17	5933	391	12,17,19,36,48	3,6,7,8,11,14,16,23,25,27,32,41,42,46
Iteration 14	15	5197	351	12,17,19,21,42	6,7,8,14,16,20,25,26,27,29,32,33,34,41,47,48

Figure 4.11: Model B's Results for DS1

Figure 4.12 shows the model solutions for DS2. For this dataset each iteration was recorded to demonstrate the method described in chapter 3. For the first iteration of the model r was constrained to 30 and the weighted distance (access) was constrained to greater than zero. Then the in the second iteration, the weighted distance was constrained to less than 11099 and thus the solution was 5347 for the same r value and the same weighted coverage value. This means that in the first iteration the model stopped running once a weighted distance value greater than zero was achieved even if a value lower than the achieved value existed. Thus it was necessary to constrain the weighted distance through a range of feasible values. This method of constraining was done for each r value.

Secondly, if the lowest achieved weighted distance values for the same r and weighted coverage values are compared it was observed that between iterations one through to eleven, as equity increased (by way of r decreasing), access decreased (by way of the sum of the weighted distance increasing). However once r was less than 23, when equity increased, then access increased for those demand points covered. Similar to DS1, this was because as the number of demand points not covered increased, the model was able to optimise both access and equity for the demand points covered.

Iteration	"r" value	Access	Coverage	Facilities opened	Demand Points not covered
Iteration 1	30	11099	497	4,14,24,26,29	46
Iteration 2	30	5347	497	4,14,24,26,29	46
Iteration 3	30	4023	497	4,14,24,26,29	46
Iteration 4	29	9066	497	15,16,31,36,47	46
Iteration 5	29	5949	497	24,26,29,31,32	46
Iteration 6	29	4035	497	21,24,29,31,39	46
Iteration 7	26	8240	494	16,21,31,41,43	11,12,13,18,31,37,42,46
Iteration 8	26	6005	494	21,24,31,43,44	18,46
Iteration 9	26	5981	494	21,24,31,43,44	18,46
Iteration 10	24	7824	491	16,21,22,41,43	10,18,46
Iteration 11	23	8655	474	22,28,20,36,43	5,10,29,42,46
Iteration 12	22	6266	451	18,31,36,43,47	4,5,29,34,37,40,42,46
Iteration 13	21	7976	452	14,32,33,43,50	10,11,18,21,30,37,40,42,46
Iteration 14	20	7178	440	14,32,33,43,50	10,11,17,18,21,30,37,39,40,42,46
Iteration 15	19	4231	416	12,28,32,33,43	10,16,17,18,24,27,29,30,37,39,40,42,46
Iteration 16	17	3854	393		4,7,10,16,18,19,24,27,29,30,34,37,39,40,42,44,46

Figure 4.12: Model B's Results for DS2

Figure 4.13 shows the model results for DS3. The model solutions show that as equity increased (by way of r decreasing), access also increased (by way of the sum of the weighted distance decreasing). Even though there were few exceptions like when r was equal to 24 and 23. The general trend shows that there was no trade-off between access and equity in this dataset. The first iteration begins at r equal to 26, because $d_{allowable}$ equal to 27 is the last feasible maximum allowable distance range value in the first model for this dataset.

Iteration	"r" value	Access	Coverage	Facilities opened	Demand points not covered
Iteration 1	26	6972	511	15,17,25,29,48	1
Iteration 2	26	5403	511	15,17,29,38,48	1
Iteration 3	25	3673	486	12,15,25,45,48	1,9,21,40,43
Iteration 4	24	5630	479	15,17,25,27,29	6,8,9,40,43
Iteration 5	24	4914	479	15,17,25,27,29	6,8,9,40,43
Iteration 6	23	4929	456	15,17,25,27,29	1,5,6,8,9,11,40,43,50
Iteration 7	22	4307	434	15,17,25,27,29	1,4,5,6,8,9,11,19,32,40,43,50
Iteration 8	21	3595	414	15,17,27,34,38	1,4,5,6,8,9,11,19,32,40,43,44,50
Iteration 9	20	3348	397	15,21,38,46,48	1,5,6,8,9,14,18,19,22,29,32,34,40,43,44,45

Figure 4.13: Model B's Results for DS3

Figure 4.14 shows the model results for DS5. If the lowest achieved weighted distance value for each r values is compared (from iteration one to eight), when equity increased then access decreased. However once r was equal to 22, when equity increased then access increased for those demand points covered. Similarly, the more demand points excluded by the model, the more the model was able to optimise access and equity for those demand points covered.

Iteration	"r" value	Access	Coverage	Facilities opened	Demand Points not covered
Iteration 1	26	9897	538	3,8,17,22,25	12
Iteration 2	26	4438	538	24,25,36,41	12
Iteration 3	25	10195	538	18,25,36,40,44	12
Iteration 4	25	4482	538	24,25,29,36,41	12
Iteration 5	24	10542	534	8,17,18,40,48	12,30
Iteration 6	24	7596	534	13,17,18,29,40	12,30
Iteration 7	23	10176	529	8,17,18,48,50	2,12,30
Iteration 8	23	9368	529	8,13,17,18,50	2,12,30
Iteration 9	22	7750	503	8,13,18,35,50	2,12,19,30,42
Iteration 10	21	4742	488	10,13,22,32,35	2,12,26,30,42,46
Iteration 11	20	4858	488	8,13,22,32,35	2,12,26,30,42,46
Iteration 12	19	4784	468	13,17,23,29,40	2,11,12,26,30,34,42,46
Iteration 13	19	3556	468	13,17,22,29,32	2,11,12,26,30,34,42,46
Iteration 14	18	3499	462	13,17,22,29,32	2,11,12,15,26,30,34,42,46
Iteration 15	17	3775	453	13,22,29,36,40	2,9,11,12,15,20,25,26,30,34,46

Figure 4.14: Model B's Results for DS5

Figure 4.15 shows the model results for DS7. For this dataset the iterations were recorded such that for the same r value, the weighted distance was constrained until the weighted coverage value deteriorated. Thus for iterations one to eight when r was 29 the weighted distance was constrained to determine what effect it had not only the solution weighted distance but the solution weighted coverage value. Therefore from iteration one to three it was observed that the weighted coverage value stayed the same even when the weighted distance (access) value decreased. However once the weighted distance was constrained to less than 6638 then the weighted coverage value decreased. This means for the same r value; when access increased (by way of the weighted distance decreasing) then the weighted coverage decreased, thus more demand points got excluded; shown by comparing the model solutions of iteration one through to iteration eight.

Secondly if the lowest achieved weighted distance value for each r value and weighted coverage value (iteration three and iteration ten) is compared, the general trend is observed; that as equity increases (by way of r decreasing) then access decreases (by way of the weighted distance increasing) until r is equal to 27, then access increases and equity increases for all those demand points covered because more demand points are excluded. There is however an exception to this trend when r is equal to 25 and 24 as shown in the figure.

Iteration	"r" value	Access	Coverage	Facilities opened	Demand Points not covered
Iteration 1	29	8622	548	4,6,17,28,43	12
Iteration 2	29	7731	548	11,15,27,35,45	12
Iteration 3	29	6638	548	3,4,17,26,43	12
Iteration 4	29	5988	545	3,4,17,26,43	12,23
Iteration 5	29	4644	536	3,4,17,32,43	2,12,
Iteration 6	29	3865	531	3,17,21,35,43	2,8,12
Iteration 7	29	2956	515	3,17,21,24,35	2,8,12,30,44
Iteration 8	29	1998	503	21,24,30,35,36	2,5,8,12,17,23,32,39,44
Iteration 9	28	14342	538	1,7,28,44,45,	12,37
Iteration 10	28	10871	538	3,7,11,43,45	12,37
Iteration 11	27	9850	534	6,17,25,28,43	2,13,30
Iteration 12	26	11731	531	3,7,11,25,29	12,41
Iteration 13	26	10721	531	1,3,7,25,43	12,41
Iteration 14	26	7928	531	11,25,28,32,43	2,12,23,30
Iteration 15	25	10714	519	3,7,10,16,45	12,17,33,39
Iteration 16	25	9417	519	3,7,29,38,45	12,17,33,39
Iteration 17	25	8904	519	3,7,38,43,45	12,17,33,39
Iteration 18	25	8445	519	3,7,38,43,45	12,17,33,39
Iteration 19	24	9855	515	3,7,20,29,45	12,17,33,36,39
Iteration 20	24	8672	515	7,29,37,45	12,17,33,36,39
Iteration 21	22	7560	488	3,7,29,37,35	1,2,12,17,26,33,36,39,43
Iteration 22	21	6942	466	7,13,29,37,45	1,2,11,12,17,24,26,33,34,36,39,43
Iteration 23	20	6464	446	7,13,2937,45	1,2,11,12,17,19,24,26,33,34,36,39,43
Iteration 24	19	4712	425	10,13,19,21,48	1,2,7,8,11,12,16,17,23,26,33,39,43,44
Iteration 25	17	3391	369	5,15,16,20,46	1,2,3,6,8,11,12,14,17,21,23,26,30,32,33,34,36,39,41,45,48

Figure 4.15: Model B's Results for DS7

4.3 Analysis

4.3.1 Facility Changes

The analysis of the changes in facility selection focus on the results of Model A. The results of Model A show that when the access objective significantly increased and the equity objective decreased through each model iteration, the solution resulted in the selection of new facilities and thus the new assignment of demand. Figures 4.16, 4.17 and 4.18 show the change in facility selection and location as well as the change in facility assignment for DS2. When the location allocation diagram of iteration one is compared to that of iteration eight there is significant changes in the location of facilities and the allocation of demand. This is because iteration one's solution favoured a more median-based (access) solution where the location allocation focused mainly on minimising the sum of the weighted distance between facilities and demand points while iteration eight favoured a more center-based (equity) solution where the location allocation focused mainly on minimising the maximum allowable distance between a selected facility and the demand point it serves. This is illustrated in the diagrams. Figure 4.16 locates median facilities such that demand is fairly evenly spread around the facilities. In contrast, Figure 4.18 locates facilities such that demand points are clustered around their respective selected facility.

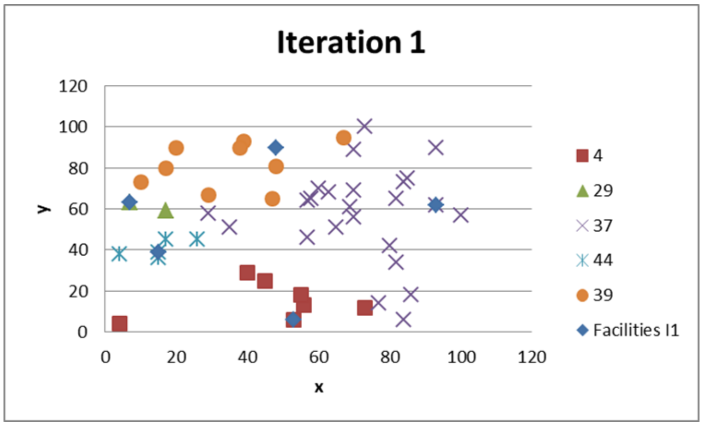

Figure 4.16: Location Allocation Diagram for DS2: Iteration one

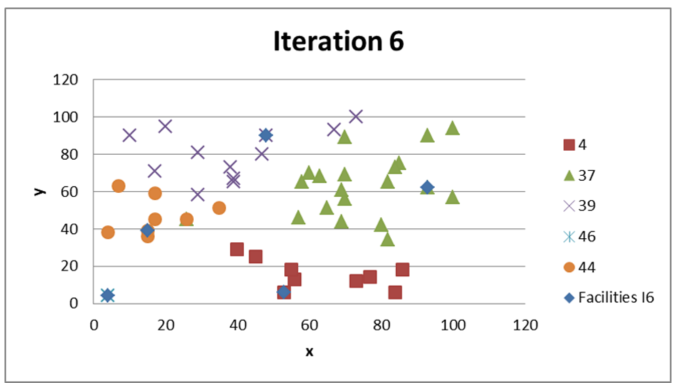

Figure 4.17: Location Allocation Diagram for DS2: Iteration six

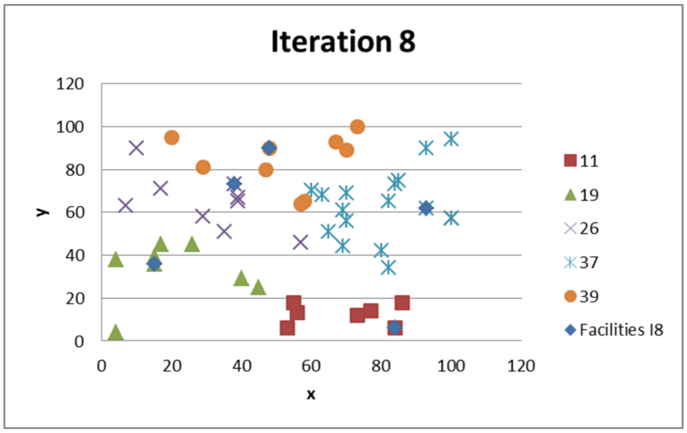

Figure 4.18: Location Allocation Diagram for DS2: Iteration eight

Table 4.1: DS2: Iteration one assignment

Facility selected	Demand point assigned
4	2,3,4,5,22,32,46
29	29,41
37	6,8,11,12,13,14,16,18,20,21,23,24,25,27,30,31,35,37,38,40,42,45,47,49,50
39	1,7,9,10,15,17,26,39,43,45
44	19,33,34,36,44

Table 4.2: DS2: Iteration six assignment

Facility selected	Demand point assigned
4	2,3,4,5,11,12,22,31,32
37	6,8,11,13,14,16,20,21,23,24,25,27,28,30,36,37,38,40,42,45,47,50
39	1,7,9,10,15,17,26,39,43,45, 48 49
44	19, 25,33,34,36, 41,44
46	46

Table 4.3: DS2: Iteration eight assignment

Facility Selected	Demand point assigned
11	2,4,11,12,22,31,32
19	3,5,19,33,34,36,44,46
26	1,9,17,21,25,26,29,42,48,49
37	6,8,13,14,16,20,24,25,28,30,37,38,40,42,45,47,50
39	10,15,18,23,27,35,39,43

Tables 4.1, 4.2 and 4.3 show the facilities selected by the model and the demand points each facility served. It must be noted that a facility chosen serves its own demand point as demonstrated by the results.

Facility changes can also most notably be observed in Figures 4.19 and 4.20 for DS7; Figures 4.21 and 4.22 for DS12 and Figures 4.23 and 4.24 for DS13.

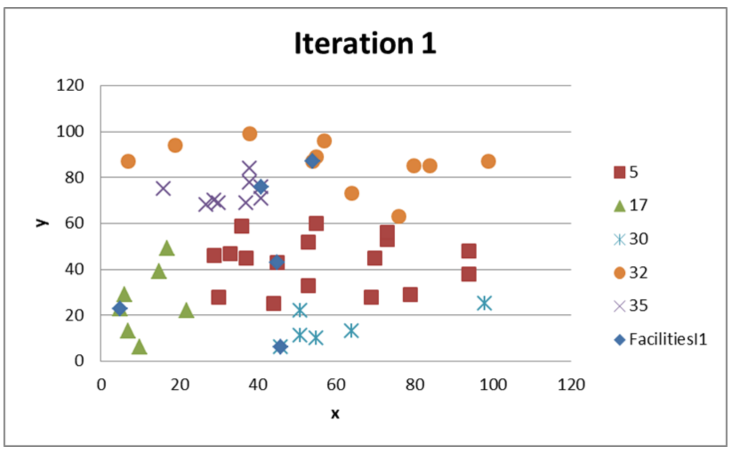

Figure 4.19: Location Allocation Diagram for DS7: Iteration one

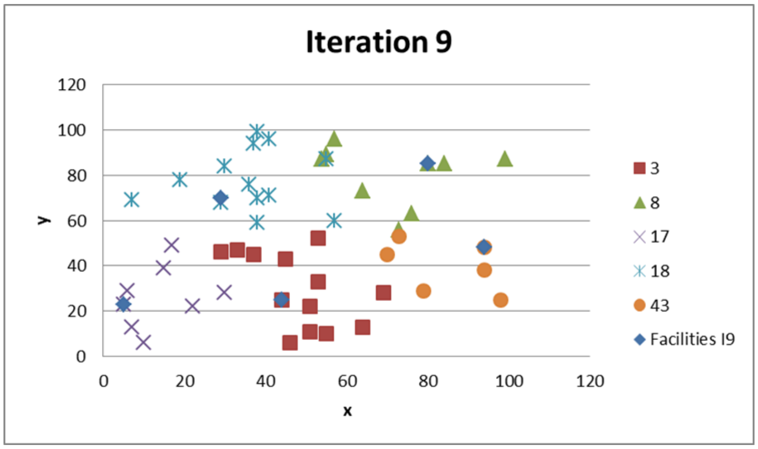

Figure 4.20: Location Allocation Diagram for DS7: Iteration nine

The location allocation diagrams for DS7 (Figures 4.19 and 4.20) show that the first iteration's solution favoured a median based solution where the total weighted distance for each facility and its assigned demand points was minimised. The individual distance between a facility and its demand point was not the focus of this solution but rather, the total demand-weighted distance of the network. For example; iteration one and specifically, facility 5 and its assigned demand points (shown by the red squares in the diagram), the demand points are evenly spread out. However in iteration nine the facilities and their assigned demand points tend to be more clustered around their facility. Furthermore facility 8 and 43 were selected in iteration nine to serve demand that was previously served by facility 5 (in iteration one) in order to produce a solution that was more equity-based.

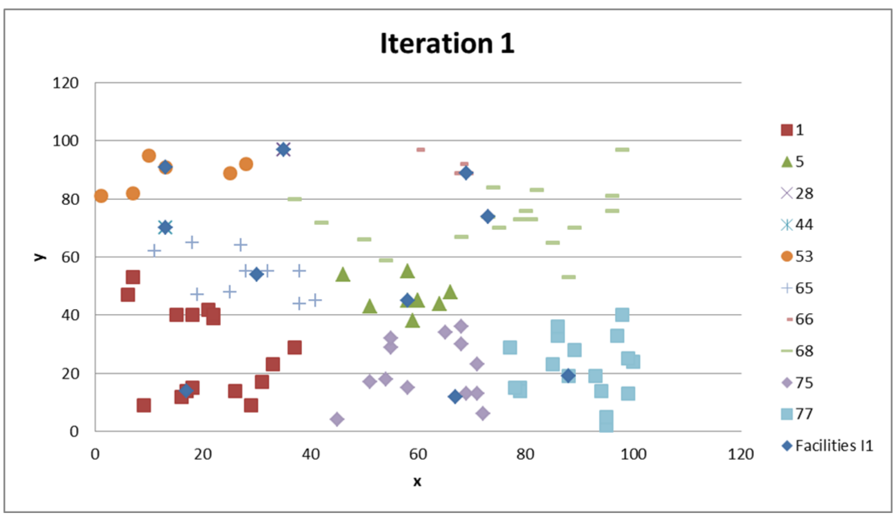

Figure 4.21: Location Allocation Diagram for DS12: Iteration one

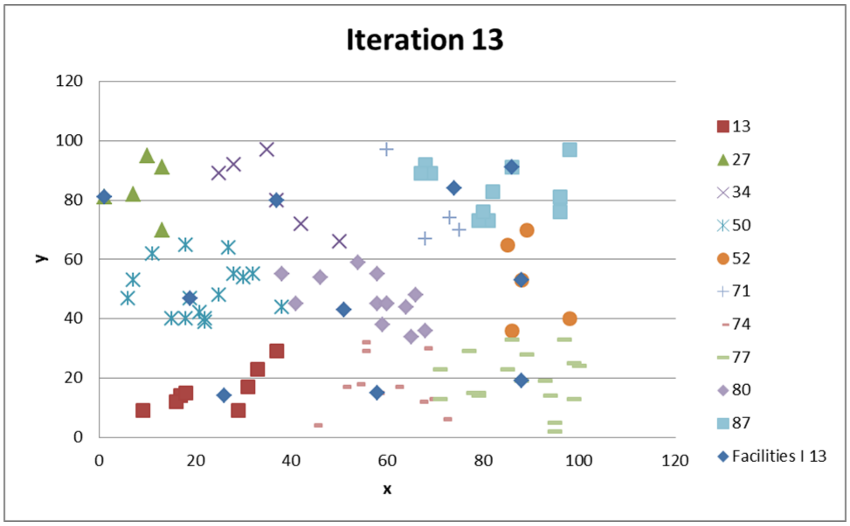

Figure 4.22: Location Allocation Diagram for DS12: Iteration thirteen

The location allocation diagrams for DS12 (Figures 4.21 and 4.22) and DS13 (Figures 4.23 and 4.24) show that for larger data points, the model solutions also produced a solution that was median-based in the first iteration and center-based in the last iteration. As shown in Figure 4.24 the demand points tended to cluster around their facility so as to adhere to the given $d_{allowable}$ constraint. Furthermore new facilities were selected and therefore new demand assignments occurred to give rise to the clustering effect.

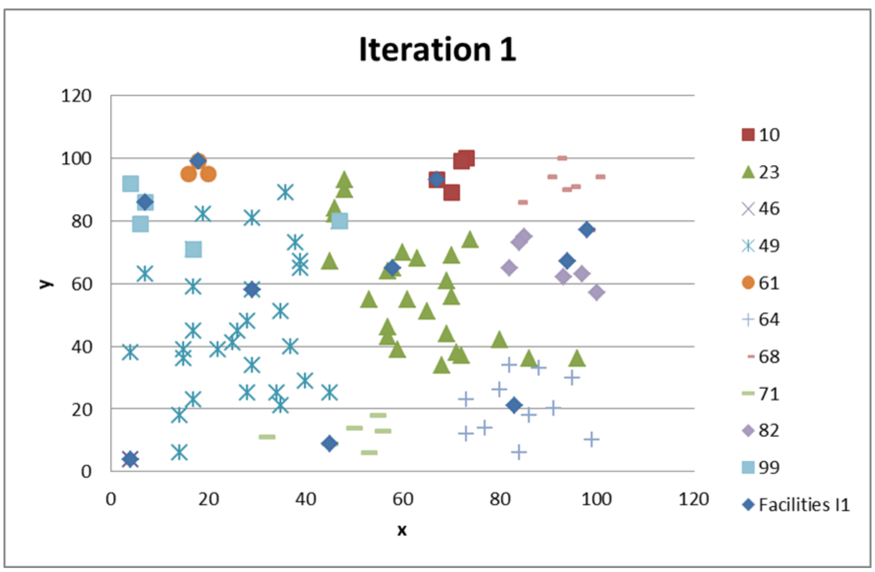

Figure 4.23: Location Allocation Diagram for DS13: Iteration one

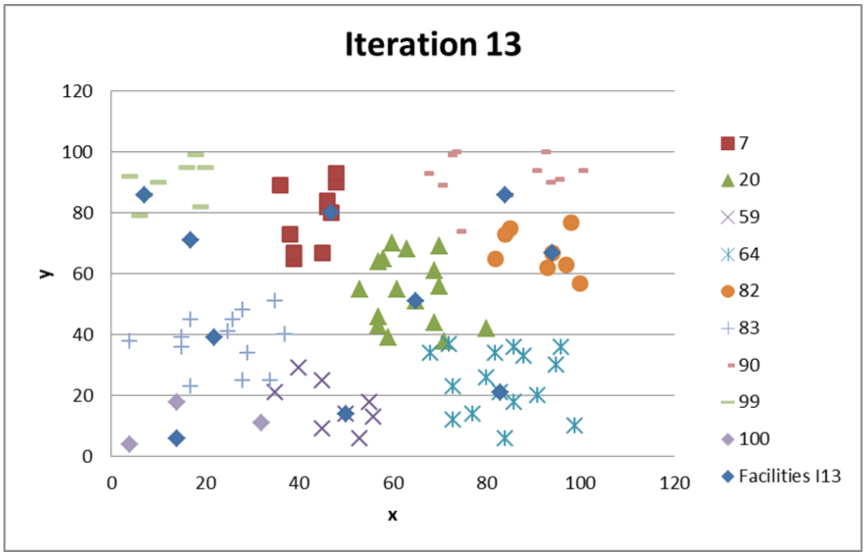

Figure 4.24: Location Allocation Diagram for DS13: Iteration thirteen

Table 4.4: DS7: Iteration one assignment

Facility selected	Demand point assigned
5	3,5,9,10,16,22,23,24,27,28,29,34,37,38,43,48,49
17	1,11,14,17,33,36,40
30	13,30,40,44,46,50
32	2,6,7,8,12,15,21,26,32,41,45
35	4,18,19,20,25,31,35,42,47

Table 4.5: DS7: Iteration nine assignment

Facility selected	Demand point assigned
3	3,5,9,10,13,22,27,28,30,38,40,46,50
8	2,6,7,8,12,15,26,32,49
17	1,11,14,17,33,34,36,39
18	4,18,19,20,21,23,26,31,35,37,41,42,45,47
43	16,24,29,43,44,48

Table 4.6: DS12: Iteration one assignment

Facilities Selected	Demand point assigned
1	1,11,13,21,33,35,36,48,51,62,69,73,81,82,93,97
5	3,5,12,22,70,78,80,92,96
28	28
44	44
53	14,27,42,53,60,84
65	4,9,10,17,26,31,49,50,65,86,94
66	20,40,66,95
68	24,30,32,34,37,39,41,47,52,55,57,63,68,71,79,83,87,98,99
75	8,15,16,19,38,43,45,56,68,74,75,85,89,90,91
77	2,6,7,18,23,25,46,54,58,59,61,64,72,76,77,88,100

Table 4.7: DS12: Iteration thirteen assignment

Facilities Selected	Demand point assigned
13	1,11,13,21,33,35,36,51
27	27,42,44,53,84
34	14,28,34,39,41,60
50	4,9,17,26,31,48,49,50,62,65,73,81,82,86,93,97
52	24,46,52,54,83
71	20,47,68,71,79
74	8,15,16,19,45,56,67,74,75,85,90
77	2,6,7,18,23,25, 29,38,58,59,61,64,72,76,77,88,89,100
80	3,5,10,12,22,43,63,70,78,80,91,92,94,96
87	30,32,37,40,55,57,66,87,95,98,99

Table 4.8: DS12: Iteration one assignment

Facilities Selected	Demand point assigned
10	10,18,27,91
23	7,8,13,20,21,23,28,35,38,39,45,47,50,51,53,54,57,66,77,78,79,81,86,89,93,96,97
46	46
49	1,3,5,9,19,25,26,29,33,34,36,41,43,44,48,49,56,60,62,65,67,74,80,83,85,87,92,95,100
61	15,61,88
64	11,12,14,22,31,52,58,63,64,72,73,94
68	30,40,68,75,84,90,98
71	2,4,32,55,59,71
82	6,16,24,37,42,76,82
89	17,69,70,99

Table 4.9: DS13: Iteration thirteen assignment

Facilities Selected	Demand point assigned
1	1,29,41,43,49
7	7,9,26,39,48,51,57,77,87,89
20	8,13,20,21,23,28,35,38,45,47,50,53,78,96,97
59	2,4,5,32,55,59,60,71
64	11,12,14,22,31,52,58,63,64,72,73,79,81,86,93,94
82	6,16,24,37,42,68,76,82
83	19,25,33,34,36,44,56,65,67,74,80,83,92,95
90	10,18,27,30,40,66,75,84,90,91,98
99	15,17,61,69,70,85,88,99
100	46,55,62,100

4.3.2 Assignment Changes

The analysis of the changes in demand assignment focus on the results of Model A. The results of Model A show that when there was a marginal increase in the access objective, a decrease in the equity objective and the selected facilities remained the same between iterations, then there was a change in the assignment of demand points to the selected facilities. The location allocation diagrams of DS7 and DS13 in Figures 4.25 to 4.29 illustrate this point.

The location allocation graphs for DS7 shown in Figures 4.25 to 4.27 as well as Tables 4.10 to 4.12 show that when facilities remained the same the assignment of demand points to the selected facilities changed. For iteration six the access objective was 4424 and the equity objective was 35; for iteration seven the access objective was 4728 and the equity objective was 33 while for iteration eight the access objective was 4894 and the equity objective was 31.

The aforementioned figures and tables show that between iterations six and seven; demand point 50 moved from facility 5 to facility 29. This was such that the constraint $d_{allowable}$ be less than equal to 33 be adhered to and thus the assignment changed so that demand point 50 could be in the maximum allowable range between its assigned facility. Between iterations seven and eight; demand point 35 moved from facility 5 to facility 19 such that the constraint $d_{allowable}$ be less than 31 be adhered to. This resulted in the sum of the weighted sum of the distances increasing to allow for that constraint to be adhered to.

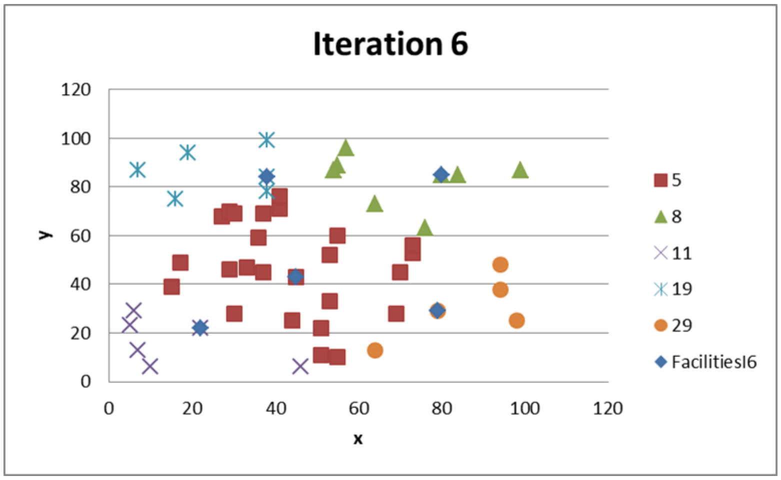

Figure 4.25: Location Allocation Diagram for DS7: Iteration six

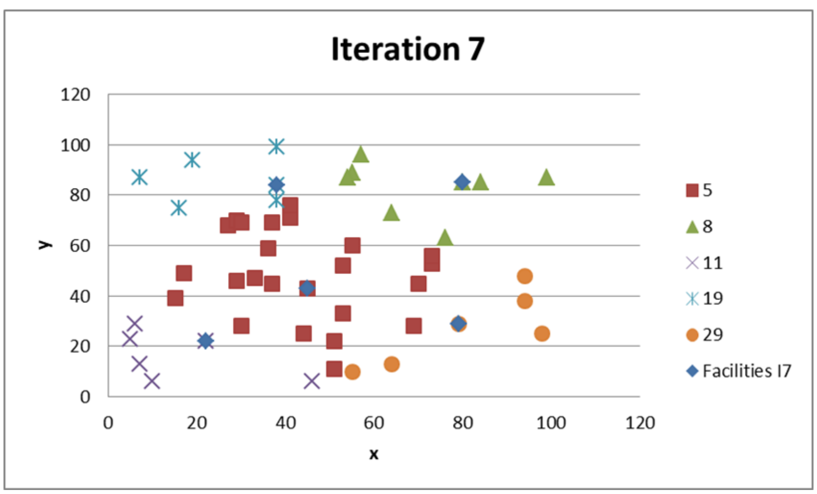

Figure 4.26: Location Allocation Diagram for DS7: Iteration seven

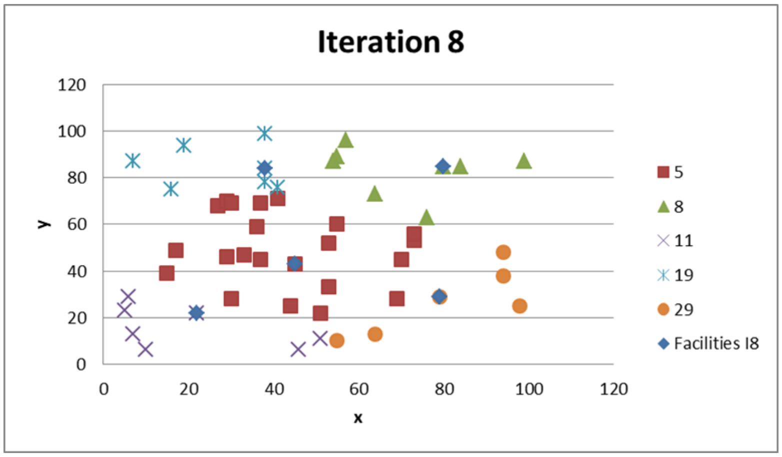

Figure 4.27: Location Allocation Diagram for DS7: Iteration eight

Table 4.10: DS7: Iteration six assignment

Facilities selected	Demand point assigned
5	3,4,5,9,10,13,14,18,20,22,23,24,27,28,31,34,35,36,37,38,40,47,48,49,50
8	2,6,7,8,12,15,26,32
11	1,11,17,30,33,39
19	19,21,25,41,42,45
29	16,29,43,44,46

Table 4.11: DS7: Iteration seven assignment

Facilities selected	Demand point assigned
5	3,4,5,9,10,13,14,18,20,22,23,24,27,28,31,34, 35,36,37,38,40,47,48,49
8	2,6,7,8,12,15,26,32
11	1,11,17,30,33,39
19	19,21,25,41,42,45
29	16,29,43,44,46,50

Table 4.12: DS7: Iteration eight assignment

Facilities selected	Demand point assigned
5	3,4,5,9,10,13,14,18,20,22,23,24,27,28,31,34,36,37,38,47,48,49
8	2,6,7,8,12,15,26,32
11	1,11,17,30,33,39,40
19	19,21,25,35,41,42,45
29	16,29,43,44,46,50

Similarly, the location allocation graphs for DS13 shown in Figures 4.28 and 4.29 as well as Tables 4.13 and 4.14 show that when the facilities stayed the same the assignment of demand points to the selected facilities changed. For iteration seven the access objective was 2264 and the equity objective was 40 while for iteration eight the access objective was 2292 and the equity objective was 35. The aforementioned figures and tables show that demand point 74 moved from facility 49 between iteration seven and eight; demand points 5 and 60 moved from facility 49 to facility 71 and demand point 81 moved from facility 10 to facility 64. These assignment changes occurred such that each demand point could fall into the maximum allowable distance range.

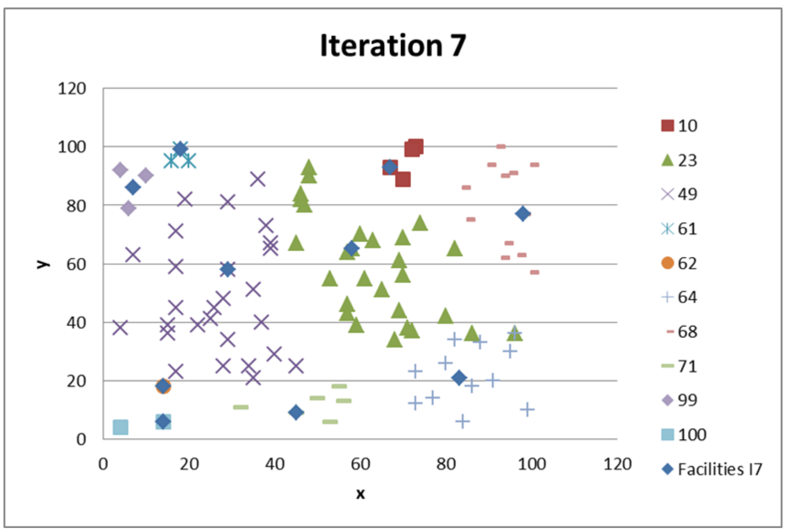

Figure 4.28: Location Allocation Diagram for DS13: Iteration seven

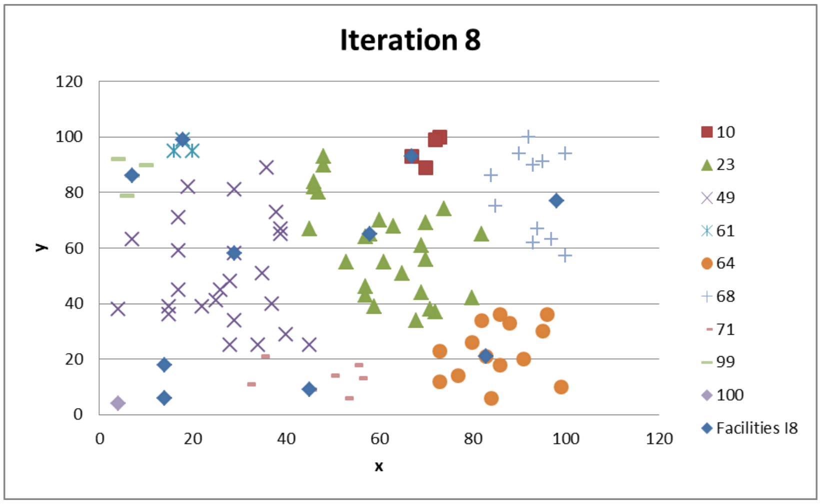

Figure 4.29: Location Allocation Diagram for DS13: Iteration eight

Table 4.13: DS13: Iteration seven assignment

Facilities Selected	Demand point assigned
10	10,18,27,91
23	6,7,8,13,20,21,23,24,28,35,38,39,45,47,50,51,53,54,57,66,77,78,79,81,86,89,96,97
49	1,3,5,9,19,25,26,29,33,34,36,41,43,44,48,49,56,60,65,67,74,80,83,85,87,92,95
61	15,61,88
62	62
64	11,12,14,22,31,52,58,63,64,72,73,93,94
68	16,30,37,40,42,68,75,76,82,84,90,98
71	2,4,32,55,59,71
99	17,69,70,99
100	46,100

Table 4.14: DS13: Iteration eight assignment

Facilities Selected	Demand point assigned
10	10,18,27,91
23	6,7,8,13,20,21,23,24,28,35,38,39,45,47,50,51,53,54,57,66,77,78,79,86,89,96,97
49	1,3,9,19,25,26,29,33,34,36,41,43,44,48,49,56,65,67,80,83,85,87,92,95
61	15,61,88
62	62,74
64	11,12,14,22,31,52,58,63,64,72,73,81,93,94
68	16,30,37,40,42,68,75,76,82,84,90,98
71	2,4,5,32,55,59,60,71
99	17,69,70,99
100	46,100

4.3.3 Coverage Changes

The analysis of coverage focuses on the results of Model B. Figures 4.30 to 4.34 are the location allocation diagrams for DS1 while Figures 4.35 to 4.38 are the location allocation diagrams for DS7.

In DS1 it was observed how coverage changes from iterations one through to thirteen. In the first iteration when r was equal to 29, then only one demand point was excluded from coverage and therefore not assigned to a facility. In the fifth iteration when r was equal to 25 then two demand points were excluded from coverage. The clustering of facility and demand points remained largely the same as in the first iteration but the facility selection changed which resulted in the demand points that were excluded being in a different region than that of in the first iteration.

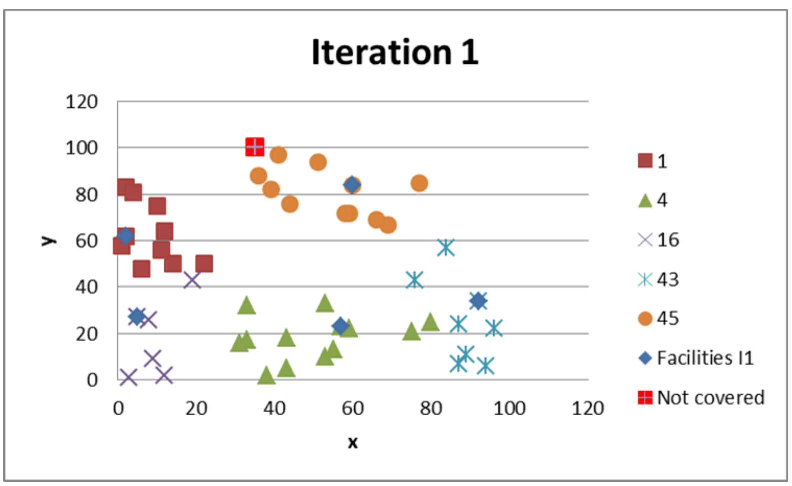

Figure 4.30: Location Allocation Diagram for DS1: Iteration one

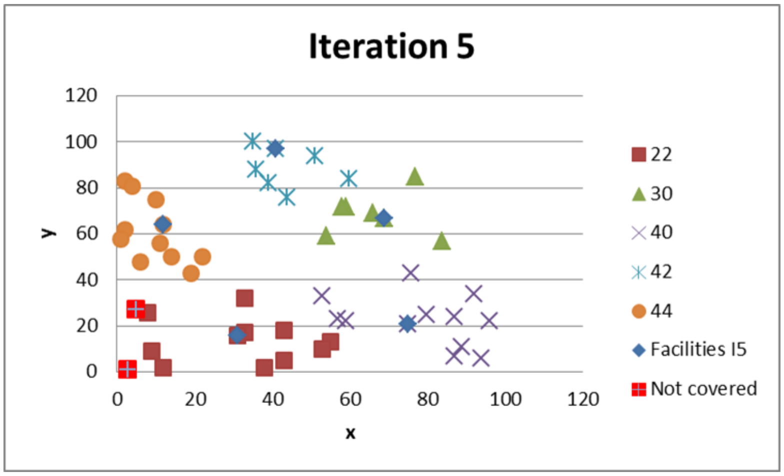

Figure 4.31: Location Allocation Diagram for DS1: Iteration five

In the seventh and eighth iteration, four and five demand points were excluded from coverage respectively. The facilities and assignment remained the same, except demand point 42 was no longer covered by facility 45 in iteration eight.

In iteration thirteen, 12 demand points were excluded from coverage. Each cluster region of facility and demand points lost atleast one demand point from coverage with the most affected region being the cluster served by facility 17. This resulted in the increase in accessibility (the decrease in the sum of the weighted distance of each facility and demand network) for those demand points covered by their facility.

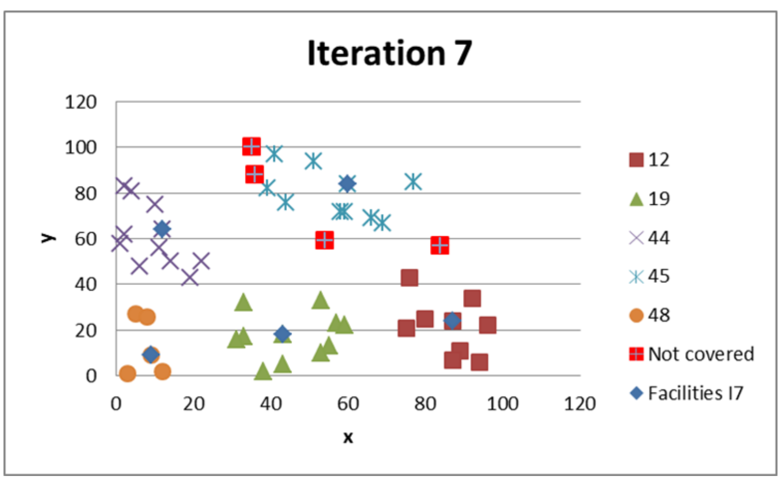

Figure 4.32: Location Allocation Diagram for DS1: Iteration seven

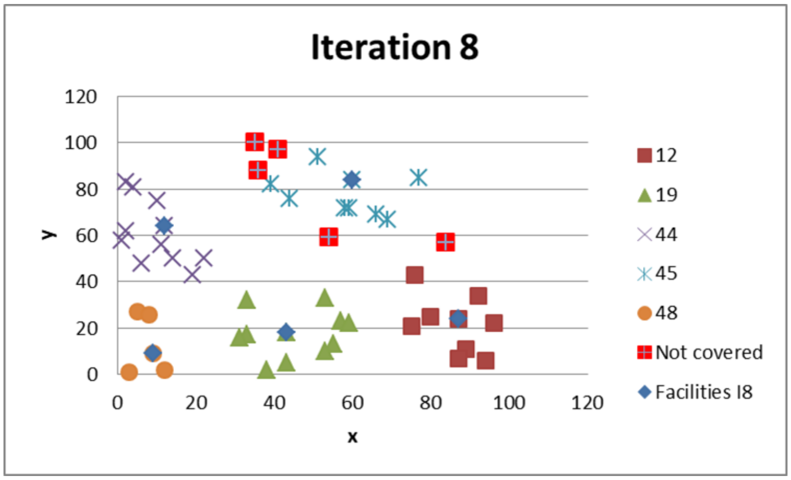

Figure 4.33: Location Allocation Diagram for DS1: Iteration eight

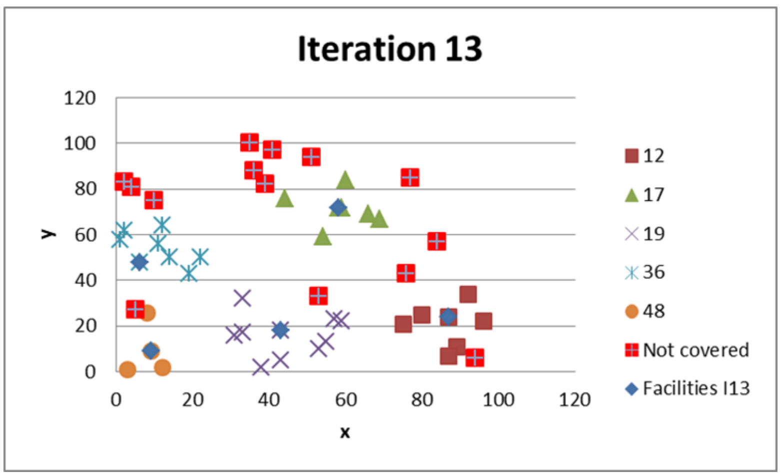

Figure 4.34: Location Allocation Diagram for DS1: Iteration thirteen

In DS7, the location allocation diagrams show the effect of constraining the sum of the weighted distance had on the solution weighted coverage for the same r value. In the first iteration only one demand point was excluded from coverage. However as the sum of the weighted distance was constrained, then coverage decreased as shown by two, three and nine demand points being excluded from iterations four, six and eight respectively. This was because when the sum of the weighted distance was constrained and thus only allowed to be equal to a certain value, even if a demand point fell within the maximum allowable coverage range, it was excluded if it had a large weighted demand value to reduce not only the total sum of the weighted distance but also the sum of the weighted distance for each facility and demand point network.

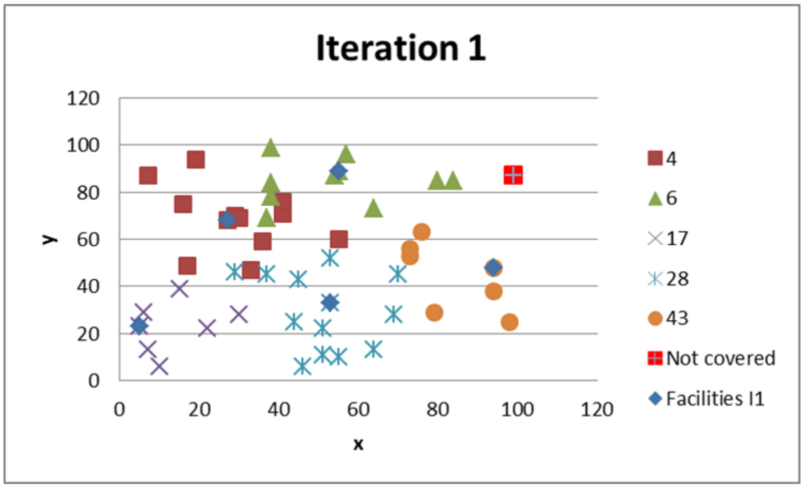

Figure 4.35: Location Allocation Diagram for DS7: Iteration one

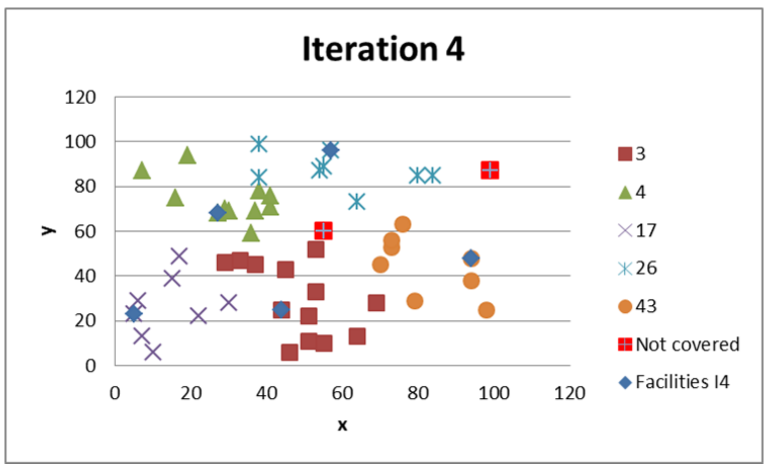

Figure 4.36: Location Allocation Diagram for DS7: Iteration four

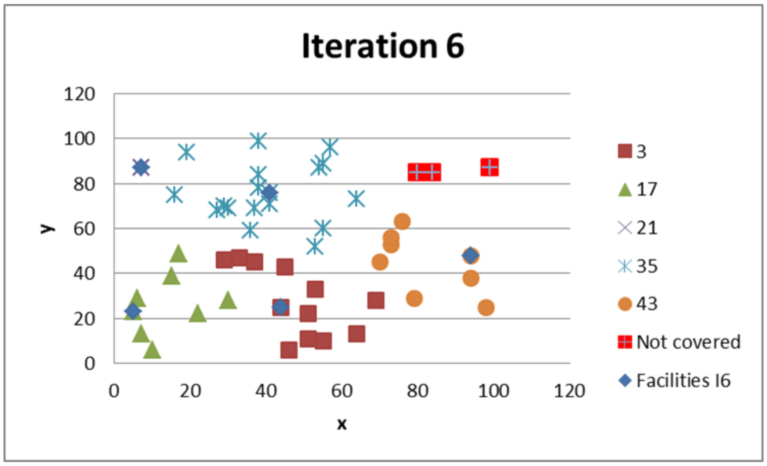

Figure 4.37: Location Allocation Diagram for DS7: Iteration six

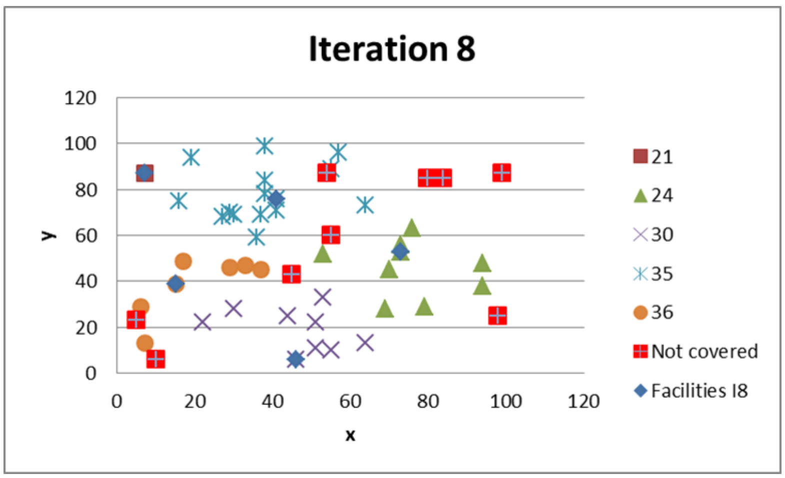

Figure 4.38: Location Allocation Diagram for DS7: Iteration eight

4.3.4 Accessibility vs Equity

Model A: Accesibility versus Equity

The results of the first model show that there was a possible trade-off between accessibility and equity even though both objective functions were minimised. As equity increased, by way of d_{max} decreasing then accessibility decreased by way of the sum weighted of the distances between facilities and demand points.

This possible trade-off is best explained using Figures 4.39 and 4.40. Figure 4.39 shows that in a network of points, when there is no constraint on $d_{allowable}$, the model will minimise the sum of the demand-weighted distances such that a median-based solution is produced. This will result in the solution choosing an accessible solution at the expense of an equitable solution in terms of the distance between each demand point and its assigned facility. This results in a lower quantity of weighted demand travelling to its nearest facility but at different distances for each demand point. The sum of the weighted distances is minimised to 10075 while the maximum allowable distance is 23.

However once a constraint is introduced; for example if $d_{allowable}$ is constrained to be less than or equal to 20 then the solution (shown in Figure 4.40) will be one that favours a center-based solution where an equitable solution will be favoured at the expense of an accessible one. This means that a higher weighted demand will have to travel to its nearest facility but at similar distances for each demand point. The sum of the weighted distances thus increases to 10850 while the maximum allowable distance decreases to 18.

The Pareto graphs also show a clear trade-off between the access and equity objectives. Furthermore, plotting the Pareto frontiers can help decision makers choose a preferred solution depending on the accessibility and equity objective they would like to achieve. This may have significant impacts on not only on location allocation decisions for healthcare facilities but also on policy and capacity planning for these facilities. Where demand allocation is highest or the total sum of the demand-weighted distance is larger such that d_{max} between a demand point and its assigned facility can be lower then policy makers and decision makers can choose to build larger hospitals and clinics in order to meet to the increased demand.

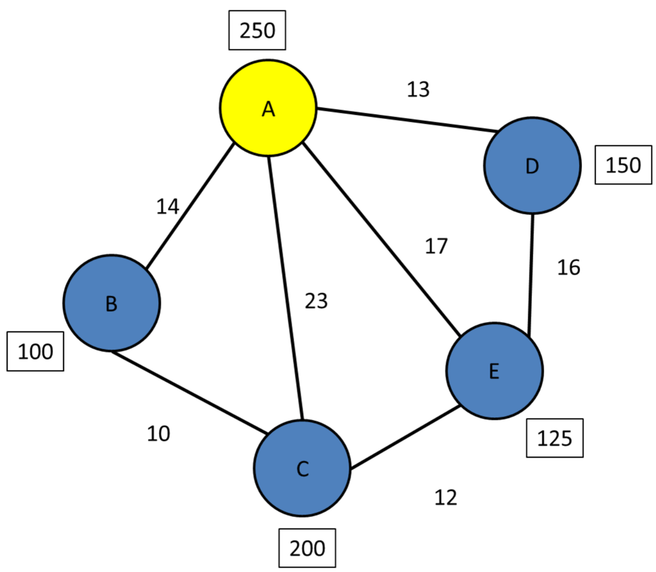

Figure 4.39: A Median-based Solution

$$\sum_{i\in I}\sum_{j\in J} w_i d_{ij} x_{ij} = 10075 \quad (4.1)$$

$$d_{max} = 23 \quad (4.2)$$

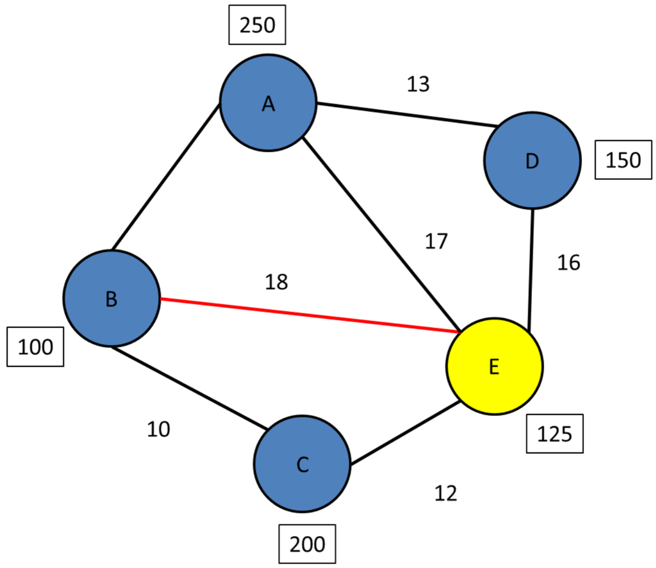

Figure 4.40: A Center-based Solution

$$\sum_{i\in I}\sum_{j\in J} w_i d_{ij} x_{ij} = 10850 \tag{4.3}$$

$$d_{max} = 18 \tag{4.4}$$

An example of the trade-off between accessibility and equity can also be shown graphically using the results of Model A. Using DS2 as an example; the network assignment for facility 4 for iteration one and six is shown in Figures 4.41 and 4.42. The figures show that in the first iteration when $d_{allowable}$ is constrained to be greater than zero and the weighted distance is minimised, the solution thus minimises the weighted distance such that the result is 1704. However once $d_{allowable}$ is constrained to be less than 40 in iteration six, the solution ensures that all demand points are within the maximum allowable distance range and then minimises the weighted distance such that the result is 2195. The diagrams demonstrate the clustering effect caused by the constraining on $d_{allowable}$.

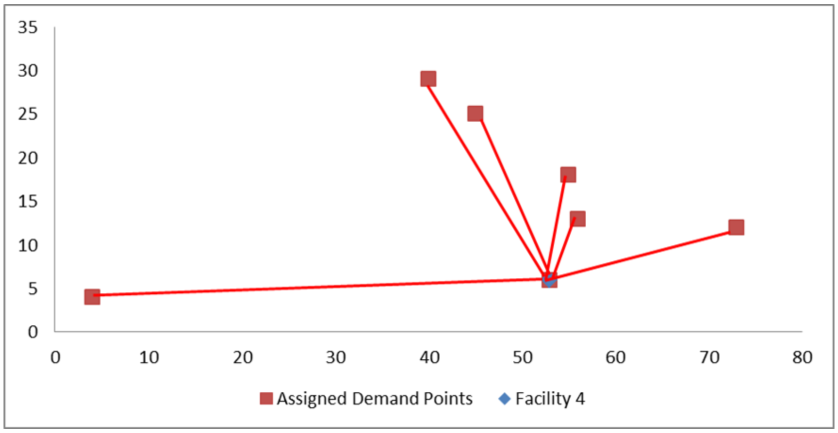

Figure 4.41: The Location Allocation Network of Facility 4 for DS2: Iteration one

Table 4.15: The demand points assigned to facility 4 in iteration one

point	x	y	demand	distance to point 4	weighted distance
2	56	13	17	7	119
3	40	29	20	26	520
4	53	6	2	0	0
5	45	25	12	20	240
22	73	12	20	20	400
32	55	18	15	12	180
46	4	4	5	49	245
					1704

$$\sum_{i\in I}\sum_{j\in J} w_i d_{ij} x_{ij} = 1704 \tag{4.5}$$

$$d_{max} = 49 \tag{4.6}$$

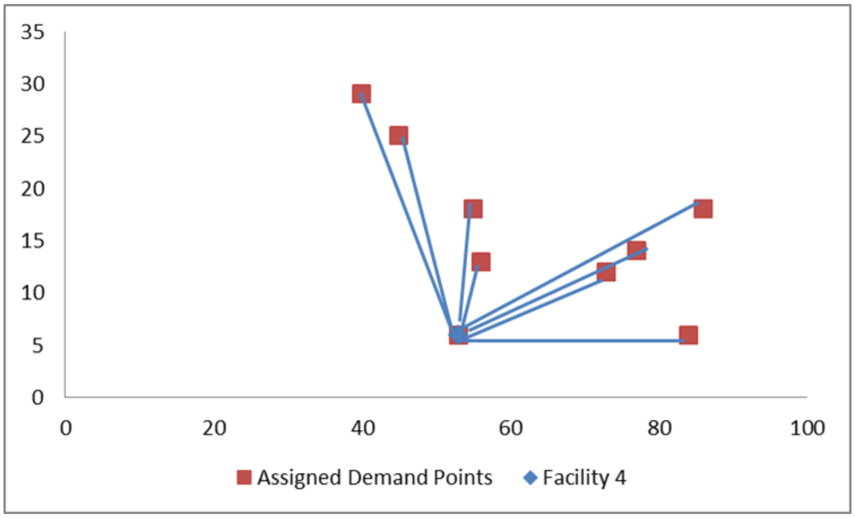

Figure 4.42: The Location Allocation Network of Facility 4 for DS2: Iteration six

Table 4.16: The demand points assigned to facility 4 in iteration six

point	x	y	demand	distance to point 4	weighted distance
2	56	13	17	7	119
3	40	29	20	26	520
4	53	6	2	0	0
5	45	25	12	20	240
11	84	6	6	31	186
12	86	18	5	35	175
22	73	12	20	20	400
31	73	12	20	20	400
32	77	14	15	25	375
					2195

$$\sum_{i \in I} \sum_{j \in J} w_i d_{ij} x_{ij} = 2195 \tag{4.7}$$

$$d_{max} = 35 \tag{4.8}$$

Model B: Accessibility versus Coverage

The results of Model B show firstly a general trend; that when equity increased then accessibility decreased until a certain r value. Then after that r value there existed no trade-off between access and equity because the model excluded more demand points from the coverage range. Thus access and equity increased for the demand points covered. However accessibility and equity was significantly reduced for the demand points that were not covered. Thus this model not only highlighted a trade-off between accessibility and equity but also introduced another dimension of coverage.

An advantage of this model is that it allows decision makers to use maximum distance ranges that were infeasible in the Model A. It however comes at the expense of excluding a number of demand points to not only achieve a lower r value but to achieve even lower weighted distance values as demonstrated by the model results.

Figure 4.43 explains how both accessibility and equity increase for demand points covered when another demand point is excluded from the coverage network. For example if decision makers let r be less than 22, then naturally point c will be excluded from coverage thus increasing access and equity for points A,B,D and E.

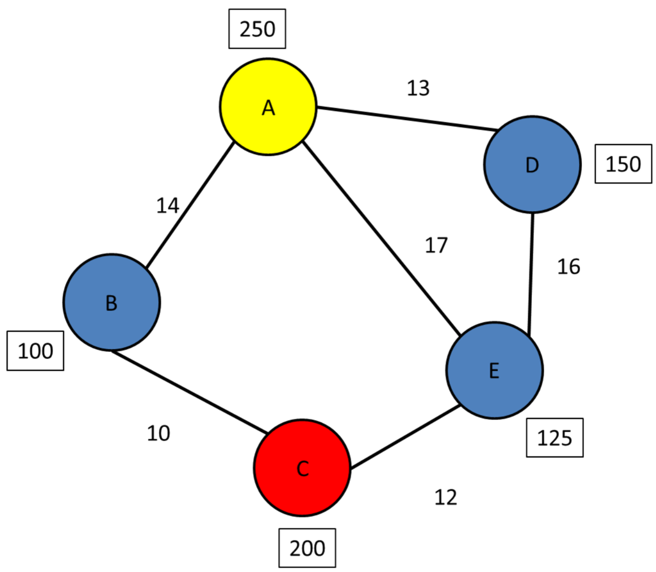

Figure 4.43: A Maximal Coverage Solution

$$\sum_{i\in I}\sum_{j\in J} w_i d_{ij} x_{ij} = 5475 \tag{4.9}$$

$$r \leq 22 \tag{4.10}$$

DS1 can be used to illustrate the aforementioned concept more clearly. The facility-demand networks of iterations eight and thirteen are compared with each other in Figures 4.44 and 4.45. Facilities 45 and 17 are chosen because they are in the same region or network. Between iterations eight and thirteen it can be observed that demand points 7, 11 and 23 are excluded as r decreases from 22 to 17. Thus as equity increases for the demand points covered so does access, as the sum of the weighted distance of the network decreases.

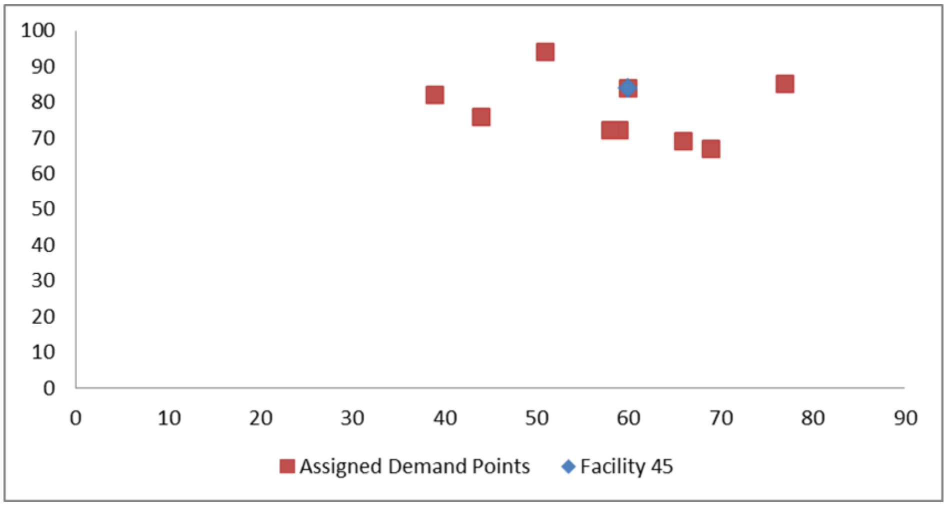

Figure 4.44: The Location Allocation Network of Facility 45 for DS1: Iteration eight

Table 4.17: The demand points assigned to facility 45 in iteration eight

point	x	y	demand	distance to point 45	weighted distance
7	77	85	14	17	238
10	59	72	6	12	72
11	39	82	10	21	210
13	44	76	3	18	54
17	58	72	14	12	168
23	51	94	13	13	169
30	69	67	17	19	323
38	66	69	9	16	144
45	60	84	8	0	0
					1378

$$\sum_{i\in I}\sum_{j\in J} w_i d_{ij} x_{ij} = 1378 \tag{4.11}$$

$$r \leq 22 \tag{4.12}$$

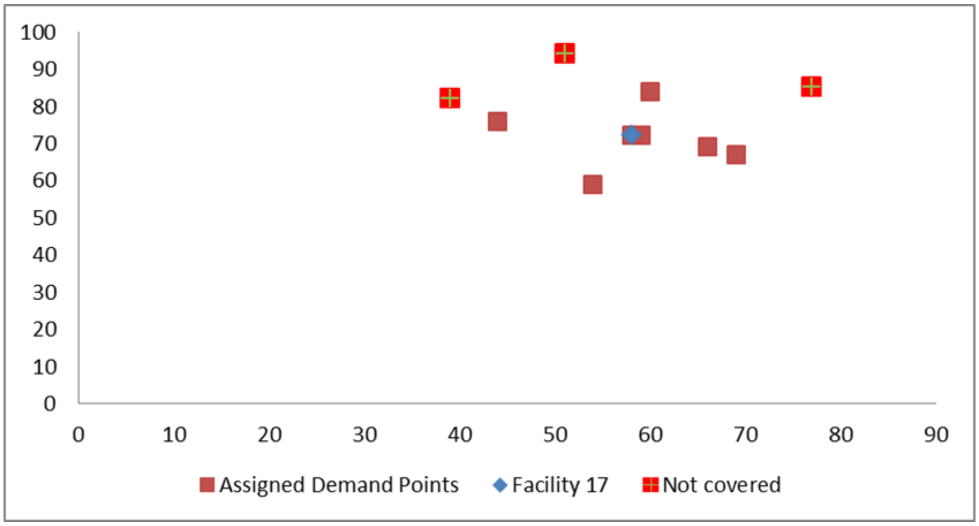

Figure 4.45: The Location Allocation Network of Facility 17 for DS1: Iteration thirteen

Table 4.18: The demand points assigned to facility 17 in iteration thirteen

point	x	y	demand	distance to point 17	weighted distance
10	59	72	6	1	6
13	44	76	3	15	45
17	58	72	14	0	0
30	69	67	17	30	510
38	66	69	9	19	171
45	60	84	8	12	96
48	54	59	9	14	126
					954

$$\sum_{i\in I}\sum_{j\in J} w_i d_{ij} x_{ij} = 954 \tag{4.13}$$

$$r \leq 17 \tag{4.14}$$

4.3.5 Pareto Efficiency

Model A: Accesibility versus Coverage

A solution of a multi-objective model is defined as Pareto efficient if by improving one objective deteriorates the other objective [16].

Model A's solutions show that the first few iterations for all datasets were not Pareto efficient because as the equity objective increased the access objective did not change. Thus for the first few iterations most dataset's solution were not efficient until iteration four for DS4, DS5, DS6 and DS7; iteration five for DS2, DS3 and DS13; iteration six for DS1; iteration seven for DS11 and iteration eight for DS12 (see Figures 4.1 and 4.2).

Although the model eventually yielded efficient solutions as $d_{allowable}$ got smaller, it requires decision makers to repeatedly run the model and thus construct a Pareto frontier to determine where the efficient solutions lie. However if decision makers are not necessarily looking to repeatedly run the model a combined solution can be found since both the sum of the weighted distance and d_{max} are minimised.

Lastly, the Pareto frontier shows the optimal or efficient solutions for each objective at a given value. Furthermore it demonstrated the possible trade-off between access and equity thus proving that this model and approach does indeed produce efficient solutions.

Model B: Accessibility versus Coverage

Model B's solutions did not definitively demonstrate that they were Pareto optimal. This was because there was no clear trend that showed that when r decreased, the sum of the weighted distance increased like in Model A. The added dimension of coverage resulted in a model where even though r decreased, the fact that the model could choose not to cover certain points resulted in a weighted distance that decreased. Therefore there was no clear trade-off between r and the sum of the weighted distances. Thus a Pareto frontier could necessarily be constructed. However it could be observed that there was a trade-off between r and coverage; thus equity and service availabilty (as defined by Burkey et al [7]). The model results for each dataset showed that when r decreased, coverage decreased.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Public healthcare is an essential public good to communities all over the world. Hospitals and clinics that provide primary care; first contact care, treatment and diagnosis of diseases are essential to the productivity and vitality of a nation. The location of these facilities is thus important. Inaccessible facilities are linked with higher mortality and morbidity rates. It is therefore important to locate facilities in which the population can access them as well as access them equally.

The purpose of this study was thus to develop a multi-objective optimisation approach for the FLP of a public PCF like a clinic or hospital. The objectives of the model were accessibility and equity.

Model A defined accessibility as the sum of the weighted distance between a facility and its assigned demand points using the PMLP; and defined equity as the maximum allowable distance range between a facility and its assigned demand point using the PCLP. This model yielded Pareto efficient solutions and allowed for the construction of a Pareto frontier to determine the trade-off between the accessibility and equity objectives. It was observed that as equity increased, accessibility decreased. Thus decision makers can readily determine and identify their preferred solution based on what accessibility and equity values they would like to achieve.

The disadvantage of this model was that for certain maximum allowable distance values the model solutions are infeasible because every demand point had to be assigned to a facility within the maximum allowable range. Thus if there was only one facility that fell outside this range the model produced an infeasible solution.

Model B defined accessibility as the sum of the weighted distance between a facility and its assigned demand points using the PMLP; and defined equity as the sum of the weighted coverage given a maximum allowable distance range. This model yielded solutions in the first few iterations where, as equity increased, access decreased. However for certain maximum allowable distance values when equity increased, access also increased. Thus there existed no clear trade-off (in terms of all demand points) between access and equity. However there existed a trade-off between equity and coverage; as equity of distance increased, coverage decreased. This model was advantageous in that it allowed decision makers to investigate maximum allowable distance values that are considered infeasible in the Model B. Furthermore if decision makers allow flexibility in terms of coverage and one or two demand points can be excluded from coverage then this model allows decision makers to find solutions for that flexibility. The disadvantage of this model is that the model must be repeatedly iterated to find the lowest weighted sum value for the same maximum allowable distance value and weighted coverage values. This could be time-consuming for decision makers. Another disadvantage of this model is that access and equity of coverage for demand points not covered essentially do not exist and this is not suitable for the public PCF problem.

It can thus be concluded that the Model A thus proves best suited to locate public PCF as the objectives are clearly defined, the solutions are effiecient and a Pareto frontier can be constructed to select a solution that is preferred by decision makers.

5.2 Future Work

Recommendations for future work of this project is to apply Model A and Model B to a real life case study. Secondly the cost objective needs to be investigated and thus investigated if there exists a trade-off between cost, accessibility and equity. Lastly a multi-level formulation of the problem could also be investigated in order to introduce referrals to the system.

Bibliography

[1] Amir Ahmadi-Javid, Pardis Seyedi, and Siddhartha S Syam. A survey of healthcare facility location. *Computers & Operations Research*, 79:223–263, 2017.

[2] Alireza Boloori Arabani and Reza Zanjirani Farahani. Facility location dynamics: An overview of classifications and applications. *Computers & Industrial Engineering*, 62(1):408–420, 2012.

[3] JE Beasley. P-median- uncapacitated. operations research library, April 2018.

[4] Sara Beheshtifar and Abbas Alimoahmmadi. A multiobjective optimization approach for location-allocation of clinics. *International Transactions in Operational Research*, 22(2):313–328, 2015.

[5] Samira Bolouri, Alireza Vafaeinejad, Ali Asghar Alesheikh, and Hossein Aghamohammadi. The ordered capacitated multi-objective location-allocation problem for fire stations using spatial optimization. *ISPRS International Journal of Geo-Information*, 7(2):44, 2018.

[6] Maria Elena Bruni, Domenico Conforti, Nicola Sicilia, and Sandro Trotta. A new organ transplantation location–allocation policy: a case study of italy. *Health Care Management Science*, 9(2):125–142, 2006.

[7] Mark L Burkey, J Bhadury, and Horst A Eiselt. A location-based comparison of health care services in four us states with efficiency and equity. *Socio-Economic Planning Sciences*, 46(2):157–163, 2012.

[8] National Planning Commission et al. National development plan 2030: Our future–make it work. *Pretoria: Presidency of South Africa*, 2012.

[9] South African Constitution and GE Devenish. South african constitution. *Pretoria. Government Printers*, 2005.

[10] Subhash Datta. Multi-criteria multi-facility location in niwai block, rajasthan. *IIMB Management Review*, 24(1):16–27, 2012.

[11] Reza Zanjirani Farahani, Maryam SteadieSeifi, and Nasrin Asgari. Multiple criteria facility location problems: A survey. *Applied Mathematical Modelling*, 34(7):1689–1709, 2010.

[12] Mumtaz Karatas and Ertan Yakıcı. An iterative solution approach to a multi-objective facility location problem. *Applied Soft Computing*, 62:272–287, 2018.

[13] R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.

[14] George Mavrotas. Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Applied mathematics and computation*, 213(2):455–465, 2009.

[15] Panagiotis Mitropoulos, Ioannis Mitropoulos, Ioannis Giannikos, and Aris Sissouras. A biobjective model for the locational planning of hospitals and health centers. *Health Care Management Science*, 9(2):171–179, 2006.

[16] Ronald L Rardin. *Optimization in operations research.* Prentice Hall, 2016.

[17] Güvenç Şahin, Haldun Süral, and Sedef Meral. Locational analysis for regionalization of turkish red crescent blood services. *Computers & Operations Research*, 34(3):692–704, 2007.

[18] Barbara Starfield, Leiyu Shi, and James Macinko. Contribution of primary care to health systems and health. *The milbank quarterly*, 83(3):457–502, 2005.

[19] Hamdy A Taha. *Operations research: An introduction.* Pearson Education Limited, 2017.

[20] Mehrdad Tamiz, Dylan Jones, and Carlos Romero. Goal programming for decision making: An overview of the current state-of-the-art. *European Journal of operational research*, 111(3):569–581, 1998.

[21] Frank Tanser. Methodology for optimising location of new primary health care facilities in rural communities: a case study in kwazulu-natal, south africa. *Journal of Epidemiology & Community Health*, 60(10):846–850, 2006.

[22] Xiuli Wang, Huazhen Yang, Zhanqi Duan, and Jay Pan. Spatial accessibility of primary health care in china: A case study in sichuan province. *Social Science & Medicine*, 209:14–24, 2018.

[23] Narong Wichapa and Porntep Khokhajaikiat. Solving multi-objective facility location problem using the fuzzy analytical hierarchy process and goal programming: a case study on infectious waste disposal centers. *Operations Research Perspectives*, 4:39–48, 2017.

[24] Wenting Zhang, Kai Cao, Shaobo Liu, and Bo Huang. A multi-objective optimization approach for health-care facility location-allocation problems in highly developed cities such as hong kong. *Computers, Environment and Urban Systems*, 59:220–230, 2016.

Appendix A

The Distance Matrices

This section includes the Python script for the calculation of the distance matrices.

```
# -*- coding: utf-8 -*-
"""
Created on Fri Jun 15 11:10:19 2018

@author: Lehlohonolo
"""
#import all the required packages
import math
from numpy import array
from numpy import zeros
from numpy import reshape
#Read the data from the text file
f= open("pmedcap13.txt", "r")
xcoords = []
ycoords = []
demand = []

#Initialise the line by line reading of the data
for line in f:
    columns = line.split()
    xcoords.append(columns[1])
    ycoords.append(columns[2])
    demand.append(columns[3])

#Combine the x and y coordinates into a matrix
coords= [list(a) for a in zip(xcoords, ycoords)]
coords= array(coords, dtype=int)
coords = reshape(coords, (100,2))
points= zeros((100,2))

# Make a second matrix for the points
for line in f:
    columns = line.split()
    xcoords.append(columns[1])
    ycoords.append(columns[2])

points= [list(a) for a in zip(xcoords, ycoords)]
points= array(points, dtype=int)
points = reshape(points, (100,2))

#Begin calculations for the distance matrix
rows,cols= coords.shape
dmat= zeros((rows,rows))
```

```
d=0
r=0
c=0
i=0
j=0


for P in points:
    r= r +1 # To keep track of which row in the coords we're i
    i= r -1 # To keep track of which row in the distance matri
    for A in coords:
        d= math.sqrt((P[0]-A[0])**2 + (P[1]-A[1])**2)
        c= c+1 # To keep track of the column number iteration
        j= c-(rows*i) -1 #calculating the distance matrix colu
        dmat[i][j]= round(d)

#convert matrix into an integer array
dmat=array(dmat, dtype=float)
print dmat
print demand

#write the distance matrix into a csv file
import csv

#create a csv file
with open("pmedcap13.csv", "wb") as csvfile:
     csvwriter = csv.writer(csvfile, delimiter= ",")
     for r in dmat: #write each row of the matrix into the fi
         csvwriter.writerow(r)

#write the demand points into a csv file
#with open("pmedcap13demand.csv", "wb") as csvfile:
     #csvwriter = csv.writer(csvfile, delimiter= ",")
     #csvwriter.writerow(demand)
```

Appendix B

Verification and Validation: Model A Lingo Scripts

This section includes the Lingo scripts of the verification and validation Model A and a screenshot of the Excel output of the test model.

```
MODEL:
SETS:
I:weight;
J: open;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;

MIN= @SUM(Links:distance* assign);

access= @SUM(Links:distance* assign);

!Objective 2 constrained;
dmax>0;

!Objective function 2;
!MIN = dmax;

!Objective 1 contsrained;
!@SUM(Links: distance* assign)>0;

!Combined Objective function;

!MIN= @SUM(Links:distance* assign) + dmax;


!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to one facilities;
@FOR(I: @SUM(J: assign)= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links: distance*assign <= dmax);

!Enforcing the binary for facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));


Data:
!Importing data from excel
!set members;
I= @OLE('Pmedshorttest.xlsx','demandpoints');
J= @OLE('Pmedshorttest.xlsx','facilities');

!Input values;
weight=@OLE('Pmedshorttest.xlsx','weights');
distance=@OLE('pmedshorttest.xlsx','distance') ;

p=2;


!Exporting Results to the Excel File;
@OLE('pmedshorttest.xlsx','open')=open;
@OLE('pmedshorttest.xlsx','dmax')= dmax;
@OLE('pmedshorttest.xlsx','access')= access;
@OLE('pmedshorttest.xlsx','assign')= assign;
ENDDATA

END
```

```
MODEL:
SETS:
I: weight;
J: open ;
Links(I, J): distance, assign;
ENDSETS


!Objective function 1;

MIN= @SUM(Links: distance* assign);

access= @SUM(Links: distance* assign);

!Objective 2 constrained;
dmax>0;

!Only p number of facilities can be opened;
@SUM(J: open ) = p;

!Demand can only be assigned to one facilities;
@FOR(I: @SUM(J: assign)= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);


!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links: distance*assign <= dmax);

!Enforcing the capacity;
@For(J: @SUM(I: weight*assign) <= capacity*open);

!Enforcing the binary for facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));


Data:
!Importing data from excel
!set members;
I= @OLE('Pmedcapinputs.xlsx','demandpoints');
J= @OLE('Pmedcapinputs.xlsx','facilities');

!Input values;
capacity=@OLE('pmedcapinputs.xlsx','capacities') ;
weight= @OLE('pmedcapinputs.xlsx','weights1') ;
distance=@OLE('pmedcap1.xlsx','distance1') ;

p=5;


!Exporting Results to the Excel File;
@OLE('pmedcap1.xlsx','open')=open;
@OLE('pmedcap1.xlsx','dmax')= dmax;
@OLE('pmedcap1.xlsx','access')= access;

ENDDATA

END
```

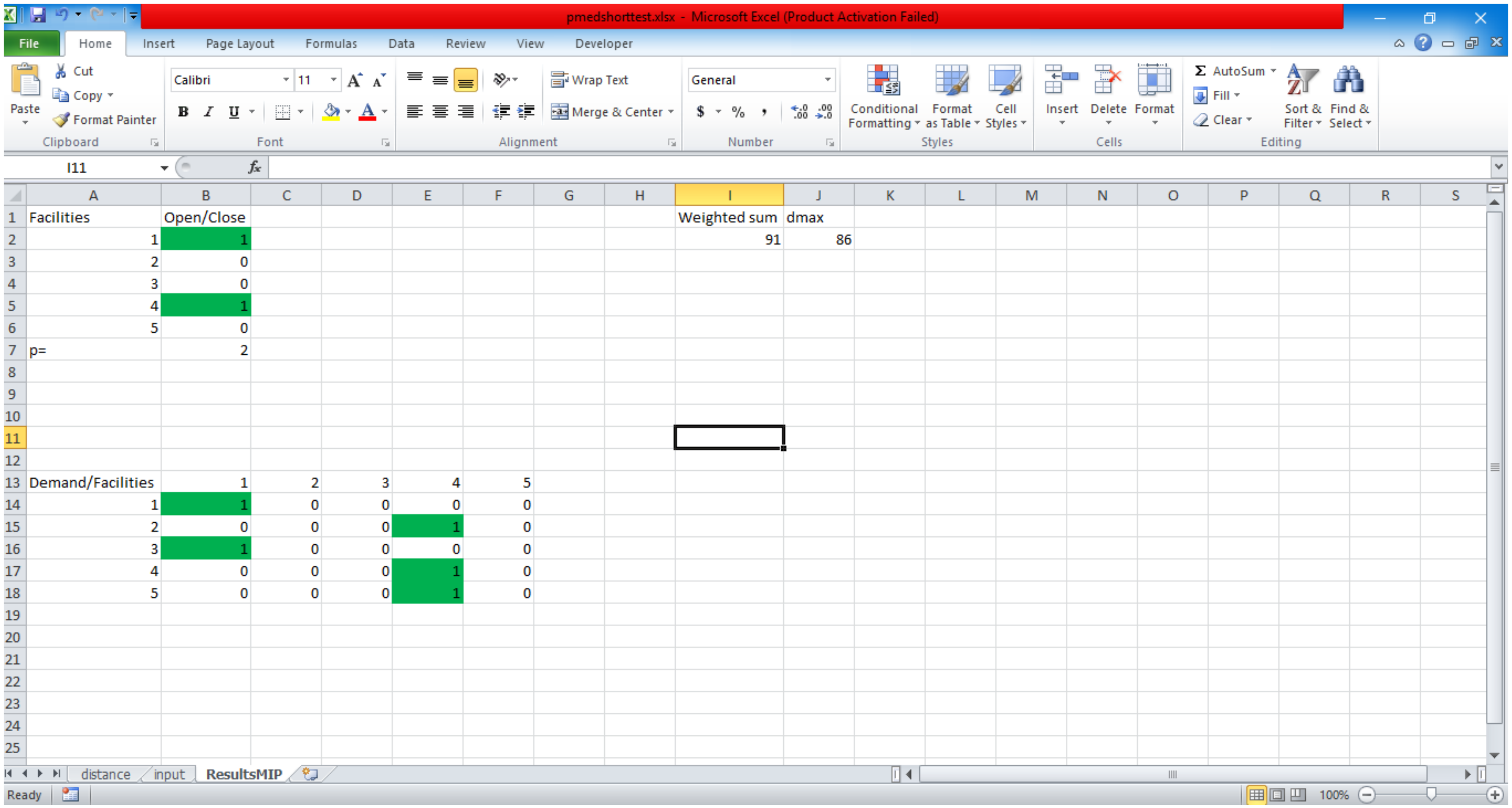

Figure B.1: A Screenshot of the Excel Spreadsheet of the Test Model Results for Verification

Appendix C

Verification and Validation: Model B Lingo Scripts

This section includes the Lingo script of the verification and validation of Model B and the table of results validating Model B.

```
MODEL:
SETS:
I: weight, coverage;
J: open ;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;
!MIN= @SUM(Links:weight*distance*assign);

!Objective 2;
Max= @SUM(I: weight*coverage);

!Subject to the following constraints;

!Objective 1 constrained;

@SUM(Links: weight* distance* assign)<=1145 ;

!Objective 2 constrained;
!@SUM(I: weight*coverage)>0;

!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to a facilities if it is in the coverage range;
@FOR(I: @SUM(J: assign)<= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links:distance*assign <= r);

!Linking coverage with its associated facility assignment;
@FOR(I: @SUM(J: assign) >= coverage);

!Enforcing the binary for open facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));

!Enforcing the binary for coverage;
@FOR(Links: @BIN(coverage));

!Tracking variables;
access= @SUM(Links: weight*distance*assign);
equity= @SUM(I: weight*coverage);

Data:
!Importing data from excel
!set members;
I= @OLE('2Pmedcapinputs.xlsx','demandpoints');
J= @OLE('2Pmedcapinputs.xlsx','facilities');

!Input values;
weight= @OLE('2Pmedcapinputs.xlsx','weights1') ;
distance=@OLE('2pmedcap1.xlsx','distance1') ;

p=5;
r=120;

!Exporting Results to the Excel File;
@OLE('2pmedcap1.xlsx','open')=open;
@OLE('2pmedcap1.xlsx','assign')=assign;
@OLE('2pmedcap1.xlsx','coverage')= coverage;
```

```
@OLE('2pmedcap1.xlsx','access')= access;
@OLE('2pmedcap1.xlsx','dmax')= equity;
ENDDATA

END
```

Table C.1: Comparison of Model A's solutions with Model B's solutions

Dataset	Weighted distance	$d_{allowable}$	Facilities selected by model one	Facilities selected by model two	Weighted coverage
Dataset one	1145	120	3,6,9,44,47	3,6,9,44,47	490
Dataset one	1145	100	3,6,9,44,47	3,6,9,44,47	490
Dataset one	1524	40	6,9,13,44,47	6,9,13,44,47	490
Dataset two	1713	131	4,7,29,37,44	4,7,29,37,44	502
Dataset two	1723	40	4,7,29,37,44	4,7,29,37,44	502
Dataset two	1752	33	4,29,37,39,44	4,29,37,39,44	502
Dataset three	995	119	8,27,38,43,45	8,27,38,43,45	512
Dataset three	1047	40	12,27,38,43,45	12,27,38,43,45	512
Dataset three	1896	33	12,38,40,43,45	12,38,40,43,45	512
Dataset four	993	122	2,10,14,30,39	2,10,14,30,39	517
Dataset four	1081	50	10,14,30,38,39	10,14,30,38,39	517
Dataset four	1800	40	2,10,30,39,48	2,10,30,39,48	517
Dataset five	1573	116	6,9,29,31,43	6,9,29,31,43	541
Dataset five	1587	50	6,9,29,42,43	6,9,29,42,43	541
Dataset five	1913	40	6,9,29,31,43	6,9,29,31,43	541

Appendix D

Model A Lingo scripts

This section includes the Lingo script of Model A for each dataset. Each page is the individual script for each dataset.

```
MODEL:
SETS:
I:weight;
J: open;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;

MIN= @SUM(Links: weight * distance* assign);

access= @SUM(Links: weight * distance* assign);

!Objective 2 constrained;
dmax>0;

!Objective function 2;
!MIN = dmax;

!Objective 1 contsrained;
!@SUM(Links: distance* assign)>0;

!Combined Objective function;

!MIN= @SUM(Links: weight * distance* assign) + dmax;


!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to one facilities;
@FOR(I: @SUM(J: assign)= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links: distance*assign <= dmax);

!Enforcing the binary for facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));


Data:
!Importing data from excel
!set members;
I= @OLE('Pmedcapinputs.xlsx','demandpoints');
J= @OLE('Pmedcapinputs.xlsx','facilities');

!I100nput values;
weight=@OLE('Pmedcapinputs.xlsx','weights1');
distance=@OLE('pmedcap1.xlsx','distance1') ;

p=5;


!Exporting Results to the Excel File;
@OLE('pmedcap1.xlsx','open')=open;
@OLE('pmedcap1.xlsx','dmax')= dmax;
@OLE('pmedcap1.xlsx','access')= access;
@OLE('pmedcap1.xlsx','assign')= assign;
ENDDATA

END
```

```
MODEL:
SETS:
I:weight;
J: open;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;

MIN= @SUM(Links: weight * distance* assign);

access= @SUM(Links: weight * distance* assign);

!Objective 2 constrained;
dmax>0;

!Objective function 2;
!MIN = dmax;

!Objective 1 contsrained;
!@SUM(Links: distance* assign)>0;

!Combined Objective function;

!MIN= @SUM(Links: weight * distance* assign) + dmax;


!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to one facilities;
@FOR(I: @SUM(J: assign)= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links: distance*assign <= dmax);

!Enforcing the binary for facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));


Data:
!Importing data from excel
!set members;
I= @OLE('Pmedcapinputs.xlsx','demandpoints');
J= @OLE('Pmedcapinputs.xlsx','facilities');

!I100nput values;
weight=@OLE('Pmedcapinputs.xlsx','weights2');
distance=@OLE('pmedcap2.xlsx','distance') ;

p=5;


!Exporting Results to the Excel File;
@OLE('pmedcap2.xlsx','open')=open;
@OLE('pmedcap2.xlsx','dmax')= dmax;
@OLE('pmedcap2.xlsx','access')= access;
@OLE('pmedcap2.xlsx','assign')= assign;
ENDDATA

END
```

```
MODEL:
SETS:
I:weight;
J: open;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;

MIN= @SUM(Links: weight * distance* assign);

access= @SUM(Links: weight * distance* assign);

!Objective 2 constrained;
dmax>0;

!Objective function 2;
!MIN = dmax;

!Objective 1 contsrained;
!@SUM(Links: distance* assign)>0;

!Combined Objective function;

!MIN= @SUM(Links: weight * distance* assign) + dmax;


!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to one facilities;
@FOR(I: @SUM(J: assign)= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links: distance*assign <= dmax);

!Enforcing the binary for facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));


Data:
!Importing data from excel
!set members;
I= @OLE('Pmedcapinputs.xlsx','demandpoints');
J= @OLE('Pmedcapinputs.xlsx','facilities');

!I100nput values;
weight=@OLE('Pmedcapinputs.xlsx','weights3');
distance=@OLE('pmedcap3.xlsx','distance1') ;

p=5;


!Exporting Results to the Excel File;
@OLE('pmedcap3.xlsx','open')=open;
@OLE('pmedcap3.xlsx','dmax')= dmax;
@OLE('pmedcap3.xlsx','access')= access;
@OLE('pmedcap3.xlsx','assign')= assign;
ENDDATA

END
```

```
MODEL:
SETS:
I:weight;
J: open;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;

MIN= @SUM(Links: weight * distance* assign);

access= @SUM(Links: weight * distance* assign);

!Objective 2 constrained;
dmax>0;

!Objective function 2;
!MIN = dmax;

!Objective 1 contsrained;
!@SUM(Links: distance* assign)>0;

!Combined Objective function;

!MIN= @SUM(Links: weight * distance* assign) + dmax;


!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to one facilities;
@FOR(I: @SUM(J: assign)= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links: distance*assign <= dmax);

!Enforcing the binary for facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));


Data:
!Importing data from excel
!set members;
I= @OLE('Pmedcapinputs.xlsx','demandpoints');
J= @OLE('Pmedcapinputs.xlsx','facilities');

!I100nput values;
weight=@OLE('Pmedcapinputs.xlsx','weights4');
distance=@OLE('pmedcap4.xlsx','distance1') ;

p=5;


!Exporting Results to the Excel File;
@OLE('pmedcap4.xlsx','open')=open;
@OLE('pmedcap4.xlsx','dmax')= dmax;
@OLE('pmedcap4.xlsx','access')= access;
@OLE('pmedcap4.xlsx','assign')= assign;
ENDDATA

END
```

```
MODEL:
SETS:
I:weight;
J: open;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;

MIN= @SUM(Links: weight * distance* assign);

access= @SUM(Links: weight * distance* assign);

!Objective 2 constrained;
dmax>0;

!Objective function 2;
!MIN = dmax;

!Objective 1 contsrained;
!@SUM(Links: distance* assign)>0;

!Combined Objective function;

!MIN= @SUM(Links: weight * distance* assign) + dmax;


!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to one facilities;
@FOR(I: @SUM(J: assign)= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links: distance*assign <= dmax);

!Enforcing the binary for facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));


Data:
!Importing data from excel
!set members;
I= @OLE('Pmedcapinputs.xlsx','demandpoints');
J= @OLE('Pmedcapinputs.xlsx','facilities');

!I100nput values;
weight=@OLE('Pmedcapinputs.xlsx','weights5');
distance=@OLE('pmedcap5.xlsx','distance1') ;

p=5;


!Exporting Results to the Excel File;
@OLE('pmedcap5.xlsx','value')=open;
@OLE('pmedcap5.xlsx','dmax')= dmax;
@OLE('pmedcap5.xlsx','access')= access;
@OLE('pmedcap5.xlsx','assign')= assign;
ENDDATA

END
```

```
MODEL:
SETS:
I:weight;
J: open;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;

MIN= @SUM(Links: weight * distance* assign);

access= @SUM(Links: weight * distance* assign);

!Objective 2 constrained;
dmax>0;

!Objective function 2;
!MIN = dmax;

!Objective 1 contsrained;
!@SUM(Links: distance* assign)>0;

!Combined Objective function;

!MIN= @SUM(Links: weight * distance* assign) + dmax;


!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to one facilities;
@FOR(I: @SUM(J: assign)= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links: distance*assign <= dmax);

!Enforcing the binary for facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));


Data:
!Importing data from excel
!set members;
I= @OLE('Pmedcapinputs.xlsx','demandpoints');
J= @OLE('Pmedcapinputs.xlsx','facilities');

!I100nput values;
weight=@OLE('Pmedcapinputs.xlsx','weights6');
distance=@OLE('pmedcap6.xlsx','distance1') ;

p=5;


!Exporting Results to the Excel File;
@OLE('pmedcap6.xlsx','open')=open;
@OLE('pmedcap6.xlsx','dmax')= dmax;
@OLE('pmedcap6.xlsx','access')= access;
@OLE('pmedcap6.xlsx','assign')= assign;
ENDDATA

END
```

```
MODEL:
SETS:
I:weight;
J: open;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;

MIN= @SUM(Links: weight * distance* assign);

access= @SUM(Links: weight * distance* assign);

!Objective 2 constrained;
dmax>0;

!Objective function 2;
!MIN = dmax;

!Objective 1 contsrained;
!@SUM(Links: distance* assign)>0;

!Combined Objective function;

!MIN= @SUM(Links: weight * distance* assign) + dmax;


!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to one facilities;
@FOR(I: @SUM(J: assign)= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links: distance*assign <= dmax);

!Enforcing the binary for facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));


Data:
!Importing data from excel
!set members;
I= @OLE('Pmedcapinputs.xlsx','demandpoints');
J= @OLE('Pmedcapinputs.xlsx','facilities');

!I100nput values;
weight=@OLE('Pmedcapinputs.xlsx','weights7');
distance=@OLE('pmedcap7.xlsx','distance1') ;

p=5;


!Exporting Results to the Excel File;
@OLE('pmedcap7.xlsx','value')=open;
@OLE('pmedcap7.xlsx','dmax')= dmax;
@OLE('pmedcap7.xlsx','access')= access;
@OLE('pmedcap7.xlsx','assign')= assign;
ENDDATA

END
```

```
MODEL:
SETS:
I:weight;
J: open;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;

MIN= @SUM(Links: weight * distance* assign);

access= @SUM(Links: weight * distance* assign);

!Objective 2 constrained;
dmax>0;

!Objective function 2;
!MIN = dmax;

!Objective 1 contsrained;
!@SUM(Links: distance* assign)>0;

!Combined Objective function;

!MIN= @SUM(Links: weight * distance* assign) + dmax;


!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to one facilities;
@FOR(I: @SUM(J: assign)= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links: distance*assign <= dmax);

!Enforcing the binary for facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));


Data:
!Importing data from excel
!set members;
I= @OLE('Pmedcapinputs.xlsx','demandpoints100');
J= @OLE('Pmedcapinputs.xlsx','facilities100');

!I100nput values;
weight=@OLE('Pmedcapinputs.xlsx','weights11');
distance=@OLE('pmedcap11.xlsx','distance1') ;

p=10;


!Exporting Results to the Excel File;
@OLE('pmedcap11.xlsx','open')=open;
@OLE('pmedcap11.xlsx','dmax')= dmax;
@OLE('pmedcap11.xlsx','access')= access;
@OLE('pmedcap11.xlsx','assign')= assign;
ENDDATA

END
```

```
MODEL:
SETS:
I:weight;
J: open;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;

MIN= @SUM(Links: weight * distance* assign);

access= @SUM(Links: weight * distance* assign);

!Objective 2 constrained;
dmax>0;

!Objective function 2;
!MIN = dmax;

!Objective 1 contsrained;
!@SUM(Links: distance* assign)>0;

!Combined Objective function;

!MIN= @SUM(Links: weight * distance* assign) + dmax;


!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to one facilities;
@FOR(I: @SUM(J: assign)= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links: distance*assign <= dmax);

!Enforcing the binary for facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));


Data:
!Importing data from excel
!set members;
I= @OLE('Pmedcapinputs.xlsx','demandpoints100');
J= @OLE('Pmedcapinputs.xlsx','facilities100');

!I100nput values;
weight=@OLE('Pmedcapinputs.xlsx','weights12');
distance=@OLE('pmedcap12.xlsx','distance1') ;

p=10;


!Exporting Results to the Excel File;
@OLE('pmedcap12.xlsx','open')=open;
@OLE('pmedcap12.xlsx','dmax')= dmax;
@OLE('pmedcap12.xlsx','access')= access;
@OLE('pmedcap12.xlsx','assign')= assign;
ENDDATA

END
```

```
MODEL:
SETS:
I:weight;
J: open;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;

MIN= @SUM(Links: weight * distance* assign);

access= @SUM(Links: weight * distance* assign);

!Objective 2 constrained;
dmax>0;

!Objective function 2;
!MIN = dmax;

!Objective 1 contsrained;
!@SUM(Links: distance* assign)>0;

!Combined Objective function;

!MIN= @SUM(Links: weight * distance* assign) + dmax;


!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to one facilities;
@FOR(I: @SUM(J: assign)= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links: distance*assign <= dmax);

!Enforcing the binary for facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));


Data:
!Importing data from excel
!set members;
I= @OLE('Pmedcapinputs.xlsx','demandpoints100');
J= @OLE('Pmedcapinputs.xlsx','facilities100');

!I100nput values;
weight=@OLE('Pmedcapinputs.xlsx','weights13');
distance=@OLE('pmedcap13.xlsx','distance1') ;

p=10;


!Exporting Results to the Excel File;
@OLE('pmedcap13.xlsx','open')=open;
@OLE('pmedcap13.xlsx','dmax')= dmax;
@OLE('pmedcap13.xlsx','access')= access;
@OLE('pmedcap13.xlsx','assign')= assign;
ENDDATA

END
```

Appendix E

Model B Lingo scripts

This section includes the Lingo script of Model B for each dataset. Each page is the individual script for each dataset.

```
MODEL:
SETS:
I: weight, coverage;
J: open ;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;
!MIN= @SUM(Links:weight*distance*assign);

!Objective 2;
Max= @SUM(I: weight*coverage);

!Subject to the following constraints;

!Objective 1 constrained;

@SUM(Links: weight* distance* assign)>0 ;

!Objective 2 constrained;
!@SUM(I: weight*coverage)>0;

!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to a facilities if it is in the coverage range;
@FOR(I: @SUM(J: assign)<= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links:distance*assign <= r);

!Linking coverage with its associated facility assignment;
@FOR(I: @SUM(J: assign) >= coverage);

!Enforcing the binary for open facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));

!Enforcing the binary for coverage;
@FOR(Links: @BIN(coverage));

!Tracking variables;
access= @SUM(Links: weight*distance*assign);
equity= @SUM(I: weight*coverage);

Data:
!Importing data from excel
!set members;
I= @OLE('2Pmedcapinputs.xlsx','demandpoints');
J= @OLE('2Pmedcapinputs.xlsx','facilities');

!Input values;
weight= @OLE('2Pmedcapinputs.xlsx','weights1') ;
distance=@OLE('2pmedcap1.xlsx','distance1') ;

p=5;
r=29;

!Exporting Results to the Excel File;
@OLE('2pmedcap1.xlsx','open')=open;
@OLE('2pmedcap1.xlsx','assign')=assign;
@OLE('2pmedcap1.xlsx','coverage')= coverage;
```

```
@OLE('2pmedcap1.xlsx','access')= access;
@OLE('2pmedcap1.xlsx','dmax')= equity;
ENDDATA

END
```

```
MODEL:
SETS:
I: weight, coverage;
J: open ;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;
!MIN= @SUM(Links:weight*distance*assign);

!Objective 2;
Max= @SUM(I: weight*coverage);

!Subject to the following constraints;

!Objective 1 constrained;

@SUM(Links: weight* distance* assign)>0 ;

!Objective 2 constrained;
!@SUM(I: weight*coverage)>0;

!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to a facilities if it is in the coverage range;
@FOR(I: @SUM(J: assign)<= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links:distance*assign <= r);

!Linking coverage with its associated facility assignment;
@FOR(I: @SUM(J: assign) >= coverage);

!Enforcing the binary for open facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));

!Enforcing the binary for coverage;
@FOR(Links: @BIN(coverage));

!Tracking variables;
access= @SUM(Links: weight*distance*assign);
equity= @SUM(I: weight*coverage);

Data:
!Importing data from excel
!set members;
I= @OLE('2Pmedcapinputs.xlsx','demandpoints');
J= @OLE('2Pmedcapinputs.xlsx','facilities');

!Input values;
weight= @OLE('2Pmedcapinputs.xlsx','weights2') ;
distance=@OLE('2pmedcap2.xlsx','distance') ;

p=5;
r=30;

!Exporting Results to the Excel File;
@OLE('2pmedcap2.xlsx','open')=open;
@OLE('2pmedcap2.xlsx','assign')=assign;
@OLE('2pmedcap2.xlsx','coverage')= coverage;
```

```
@OLE('2pmedcap2.xlsx','access')= access;
@OLE('2pmedcap2.xlsx','dmax')= equity;
ENDDATA

END
```

```
MODEL:
SETS:
I: weight, coverage;
J: open ;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;
!MIN= @SUM(Links:weight*distance*assign);

!Objective 2;
Max= @SUM(I: weight*coverage);

!Subject to the following constraints;

!Objective 1 constrained;

@SUM(Links: weight* distance* assign)>0 ;

!Objective 2 constrained;
!@SUM(I: weight*coverage)>0;

!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to a facilities if it is in the coverage range;
@FOR(I: @SUM(J: assign)<= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links:distance*assign <= r);

!Linking coverage with its associated facility assignment;
@FOR(I: @SUM(J: assign) >= coverage);

!Enforcing the binary for open facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));

!Enforcing the binary for coverage;
@FOR(Links: @BIN(coverage));

!Tracking variables;
access= @SUM(Links: weight*distance*assign);
equity= @SUM(I: weight*coverage);

Data:
!Importing data from excel
!set members;
I= @OLE('2Pmedcapinputs.xlsx','demandpoints');
J= @OLE('2Pmedcapinputs.xlsx','facilities');

!Input values;
weight= @OLE('2Pmedcapinputs.xlsx','weights3') ;
distance=@OLE('2pmedcap3.xlsx','distance1') ;

p=5;
r=26;

!Exporting Results to the Excel File;
@OLE('2pmedcap3.xlsx','open')=open;
@OLE('2pmedcap3.xlsx','assign')=assign;
@OLE('2pmedcap3.xlsx','coverage')= coverage;
```

```
@OLE('2pmedcap3.xlsx','access')= access;
@OLE('2pmedcap3.xlsx','dmax')= equity;
ENDDATA

END
```

```
MODEL:
SETS:
I: weight, coverage;
J: open ;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;
!MIN= @SUM(Links:weight*distance*assign);

!Objective 2;
Max= @SUM(I: weight*coverage);

!Subject to the following constraints;

!Objective 1 constrained;

@SUM(Links: weight* distance* assign)>0 ;

!Objective 2 constrained;
!@SUM(I: weight*coverage)>0;

!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to a facilities if it is in the coverage range;
@FOR(I: @SUM(J: assign)<= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links:distance*assign <= r);

!Linking coverage with its associated facility assignment;
@FOR(I: @SUM(J: assign) >= coverage);

!Enforcing the binary for open facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));

!Enforcing the binary for coverage;
@FOR(Links: @BIN(coverage));

!Tracking variables;
access= @SUM(Links: weight*distance*assign);
equity= @SUM(I: weight*coverage);

Data:
!Importing data from excel
!set members;
I= @OLE('2Pmedcapinputs.xlsx','demandpoints');
J= @OLE('2Pmedcapinputs.xlsx','facilities');

!Input values;
weight= @OLE('2Pmedcapinputs.xlsx','weights4') ;
distance=@OLE('2pmedcap4.xlsx','distance1') ;

p=5;
r=31;

!Exporting Results to the Excel File;
@OLE('2pmedcap4.xlsx','open')=open;
@OLE('2pmedcap4.xlsx','assign')=assign;
@OLE('2pmedcap4.xlsx','coverage')= coverage;
```

```
@OLE('2pmedcap4.xlsx','access')= access;
@OLE('2pmedcap4.xlsx','dmax')= equity;
ENDDATA

END
```

```
MODEL:
SETS:
I: weight, coverage;
J: open ;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;
!MIN= @SUM(Links:weight*distance*assign);

!Objective 2;
Max= @SUM(I: weight*coverage);

!Subject to the following constraints;

!Objective 1 constrained;

@SUM(Links: weight* distance* assign)>0 ;

!Objective 2 constrained;
!@SUM(I: weight*coverage)>0;

!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to a facilities if it is in the coverage range;
@FOR(I: @SUM(J: assign)<= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links:distance*assign <= r);

!Linking coverage with its associated facility assignment;
@FOR(I: @SUM(J: assign) >= coverage);

!Enforcing the binary for open facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));

!Enforcing the binary for coverage;
@FOR(Links: @BIN(coverage));

!Tracking variables;
access= @SUM(Links: weight*distance*assign);
equity= @SUM(I: weight*coverage);

Data:
!Importing data from excel
!set members;
I= @OLE('2Pmedcapinputs.xlsx','demandpoints');
J= @OLE('2Pmedcapinputs.xlsx','facilities');

!Input values;
weight= @OLE('2Pmedcapinputs.xlsx','weights5') ;
distance=@OLE('2pmedcap5.xlsx','distance1') ;

p=5;
r=26;

!Exporting Results to the Excel File;
@OLE('2pmedcap5.xlsx','open')=open;
@OLE('2pmedcap5.xlsx','assign')=assign;
@OLE('2pmedcap5.xlsx','coverage')= coverage;
```

```
@OLE('2pmedcap5.xlsx','access')= access;
@OLE('2pmedcap5.xlsx','dmax')= equity;
ENDDATA

END
```

```
MODEL:
SETS:
I: weight, coverage;
J: open ;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;
!MIN= @SUM(Links:weight*distance*assign);

!Objective 2;
Max= @SUM(I: weight*coverage);

!Subject to the following constraints;

!Objective 1 constrained;

@SUM(Links: weight* distance* assign)>0 ;

!Objective 2 constrained;
!@SUM(I: weight*coverage)>0;

!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to a facilities if it is in the coverage range;
@FOR(I: @SUM(J: assign)<= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links:distance*assign <= r);

!Linking coverage with its associated facility assignment;
@FOR(I: @SUM(J: assign) >= coverage);

!Enforcing the binary for open facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));

!Enforcing the binary for coverage;
@FOR(Links: @BIN(coverage));

!Tracking variables;
access= @SUM(Links: weight*distance*assign);
equity= @SUM(I: weight*coverage);

Data:
!Importing data from excel
!set members;
I= @OLE('2Pmedcapinputs.xlsx','demandpoints');
J= @OLE('2Pmedcapinputs.xlsx','facilities');

!Input values;
weight= @OLE('2Pmedcapinputs.xlsx','weights6') ;
distance=@OLE('2pmedcap6.xlsx','distance1') ;

p=5;
r=27;

!Exporting Results to the Excel File;
@OLE('2pmedcap6.xlsx','open')=open;
@OLE('2pmedcap6.xlsx','assign')=assign;
@OLE('2pmedcap6.xlsx','coverage')= coverage;
```

```
@OLE('2pmedcap6.xlsx','access')= access;
@OLE('2pmedcap6.xlsx','dmax')= equity;
ENDDATA

END
```

```
MODEL:
SETS:
I: weight, coverage;
J: open ;
Links(I, J): distance, assign;
ENDSETS

!Objective function 1;
!MIN= @SUM(Links:weight*distance*assign);

!Objective 2;
Max= @SUM(I: weight*coverage);

!Subject to the following constraints;

!Objective 1 constrained;

@SUM(Links: weight* distance* assign)>0 ;

!Objective 2 constrained;
!@SUM(I: weight*coverage)>0;

!Only p number of facilities can be opened;
@SUM(J: open ) <= p;

!Demand can only be assigned to a facilities if it is in the coverage range;
@FOR(I: @SUM(J: assign)<= 1);

!A facility can only be opened if demand is assigned to it;
@FOR(Links: assign <= open);

!The distance between demand and a facility must be less than the maximum allowable
distance;
@FOR(Links:distance*assign <= r);

!Linking coverage with its associated facility assignment;
@FOR(I: @SUM(J: assign) >= coverage);

!Enforcing the binary for open facilities;
@FOR( J: @BIN( open));

!Enforcing the binary for assignment;
@FOR(Links: @BIN(assign));

!Enforcing the binary for coverage;
@FOR(Links: @BIN(coverage));

!Tracking variables;
access= @SUM(Links: weight*distance*assign);
equity= @SUM(I: weight*coverage);

Data:
!Importing data from excel
!set members;
I= @OLE('2Pmedcapinputs.xlsx','demandpoints');
J= @OLE('2Pmedcapinputs.xlsx','facilities');

!Input values;
weight= @OLE('2Pmedcapinputs.xlsx','weights7') ;
distance=@OLE('2pmedcap7.xlsx','distance1') ;

p=5;
r=29;

!Exporting Results to the Excel File;
@OLE('2pmedcap7.xlsx','value')=open;
@OLE('2pmedcap7.xlsx','assign')=assign;
@OLE('2pmedcap7.xlsx','coverage')= coverage;
```

```
@OLE('2pmedcap7.xlsx','access')= access;
@OLE('2pmedcap7.xlsx','dmax')= equity;
ENDDATA

END
```

Appendix F

Pareto frontier Graphs for Model A

This section includes the Pareto frontier graphs for DS1 to DS11. These graphs describe the possible trade-off between the accessibility and equity objectives in the results of Model A.

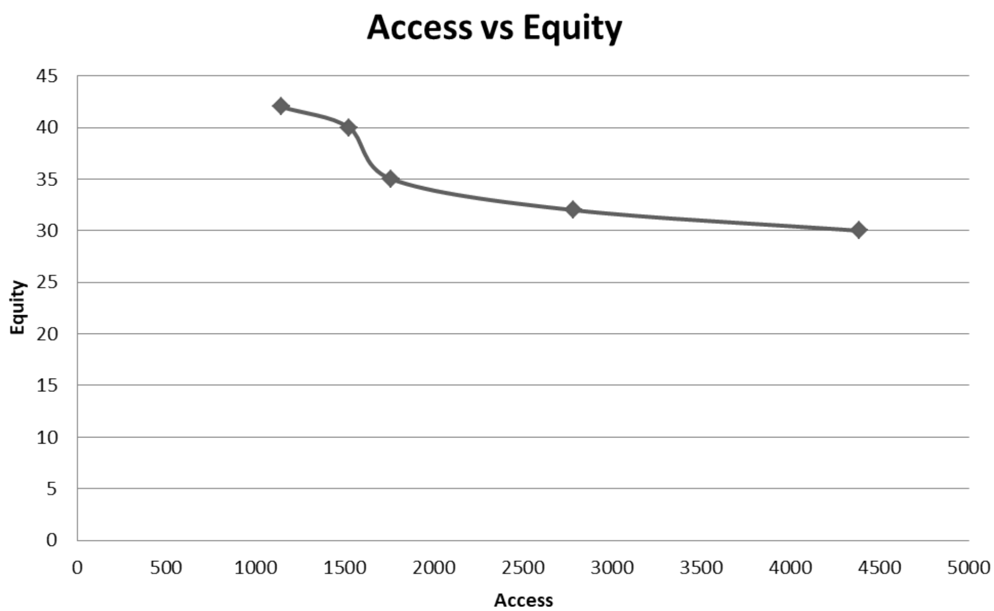

Figure F.1: Graph showing the Pareto frontier for DS1

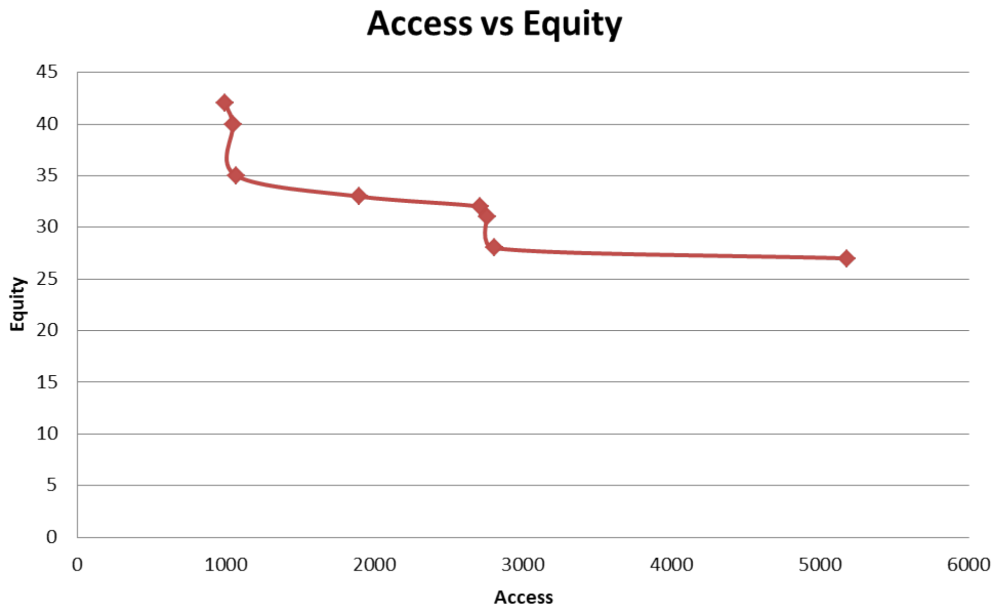

Figure F.2: Graph showing the Pareto frontier for DS3

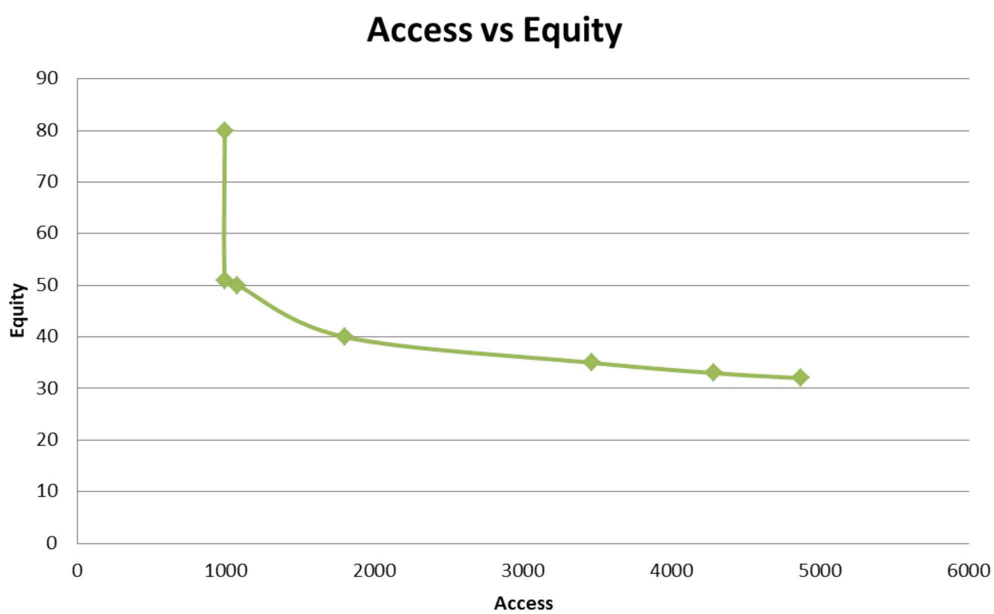

Figure F.3: Graph showing the Pareto frontier for DS4

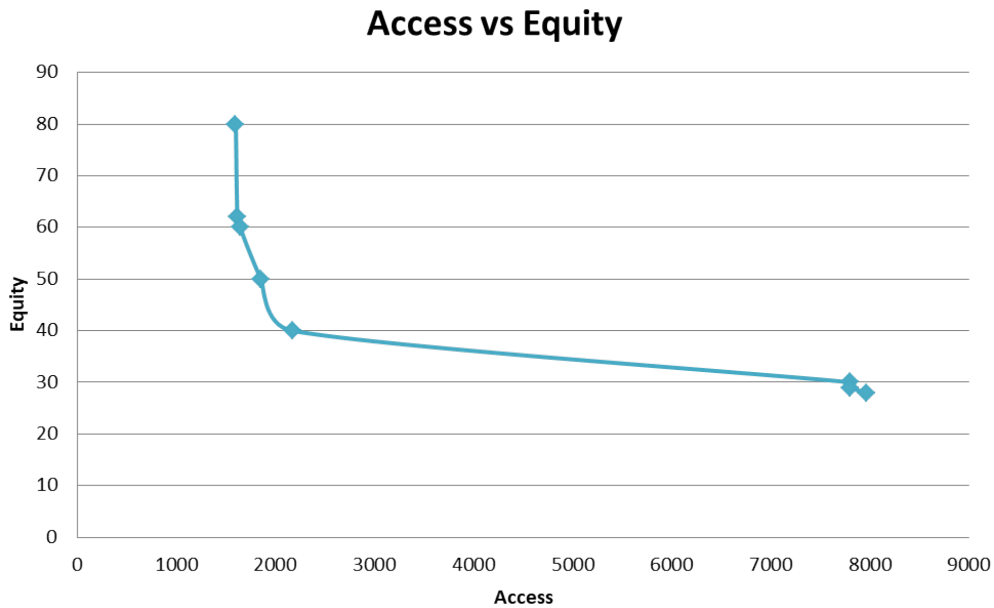

Figure F.4: Graph showing the Pareto frontier for DS6

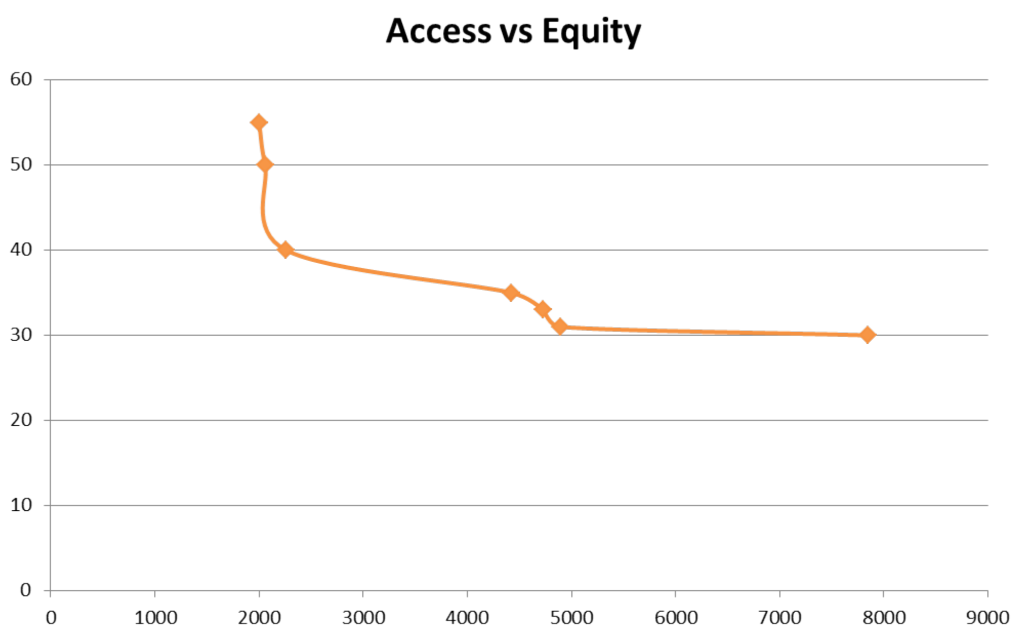

Figure F.5: Graph showing the Pareto frontier for DS7

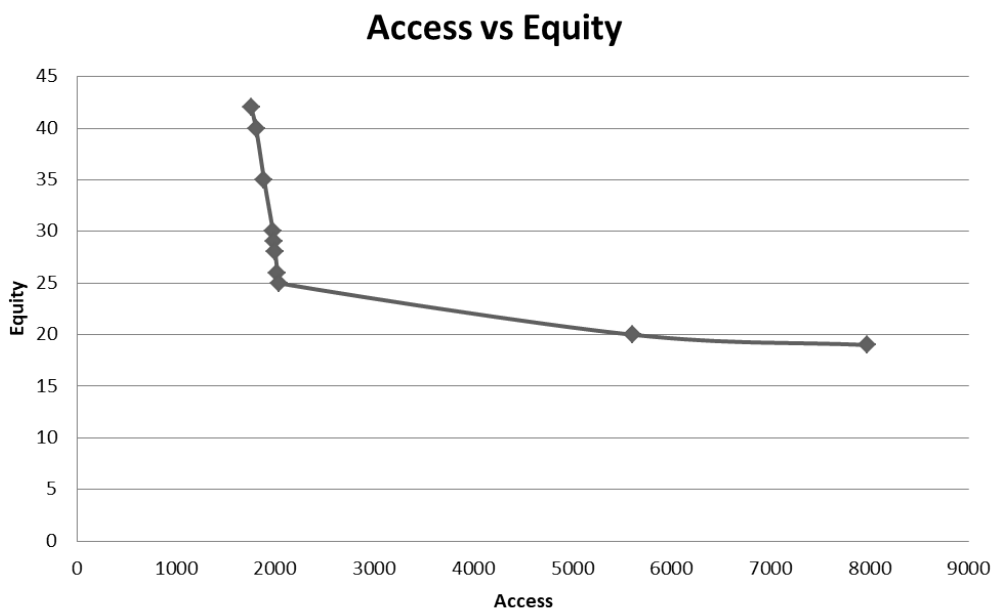

Figure F.6: Graph showing the Pareto frontier for DS11

Appendix G

Model A Results

This section provides the tables showing the results of Model A for DS1 to DS11. Each table shows the accessibility and equity objective achieved as well as the facilities selected by the model for each dataset.

Iteration	Access	Equity	Facilities opened
Iteration 1	1145	120	3,6,9,44,47
Iteration 2	1145	100	3,6,9,44,47
Iteration 3	1145	80	3,6,9,44,47
Iteration 4	1145	60	3,6,9,44,47
Iteration 5	1145	45	3,6,9,44,47
Iteration 6	1524	40	6,9,13,44,47
Iteration 7	1762	37	6,9,13,47,50
Iteration 8	1762	36	6,9,13,47,50
Iteration 9	1762	35	6,9,13,47,50
Iteration 10	2268	34	6,9,13,22,44
Iteration 11	2782	33	3,9,22,25,44
Iteration 12	2782	32	3,9,22,25,44
Iteration 13	4386	31	27,43,44,45,48
Iteration 14	4386	30	27,43,44,45,48
Combined	63.94	41	3,6,9,44,47

Figure G.1: Model A's Results for DS1

Iteration	Access	Equity	Facilities opened
Iteration 1	995	119	8,27,38,43,45
Iteration 2	995	100	8,27,38,43,45
Iteration 3	995	80	8,27,38,43,45
Iteration 4	995	60	8,27,38,43,45
Iteration 5	1047	40	12,27,38,43,45
Iteration 6	1072	35	1,27,38,43,45
Iteration 7	1896	33	12,38,40,43,45
Iteration 8	2714	32	10,12,38,40,43
Iteration 9	2757	31	10,12,38,40,43
Iteration 10	2807	30	12,15,38,45,48
Iteration 11	2807	29	12,15,38,45,48
Iteration 12	2807	28	12,15,38,45,48
Iteration 13	5178	27	12,15,26,45,48
Combined	21.44	35	1,27,38,43,45

Figure G.2: Model A's Results for DS3

Iteration	Access	Equity	Facilities opened
Iteration 1	993	122	2,10,14,30,39
Iteration 2	993	100	2,10,14,30,39
Iteration 3	993	80	2,10,14,30,39
Iteration 4	995	60	10,14,30,36,39
Iteration 5	1081	50	10,14,30,38,39
Iteration 6	1800	40	2,10,30,39,48
Iteration 7	3461	35	10,28,30,34,49
Iteration 8	4283	33	7,10,30,34,49
Iteration 9	4871	32	7,30,34,49,50
Combined	32.28	42	6,9,29,31,43

Figure G.3: Model A's Results for DS4

Iteration	Access	Equity	Facilities opened
Iteration 1	1601	115	13,20,40,42,46
Iteration 2	1601	100	13,20,40,42,46
Iteration 3	1601	80	13,20,40,42,46
Iteration 4	1617	70	13,15,40,42,48
Iteration 5	1645	60	13,15,40,42,48
Iteration 6	1857	50	13,15,20,40,42
Iteration 7	2171	40	13,15,20,24,42
Iteration 8	7802	30	9,20,21,45,49
Iteration 9	7802	29	9,20,21,45,49
Iteration 10	7966	28	9,20,21,45,48
Combined	46.78	36	13,15,20,24,42

Figure G.4: Model A's Results for DS6

Iteration	Access	Equity	Facilities opened
Iteration 1	1999	120	5,17,30,32,35
Iteration 2	1999	80	5,17,30,32,35
Iteration 3	1999	60	5,17,30,32,35
Iteration 4	2058	50	5,17,24,30,32
Iteration 5	2259	40	5,12,17,24,35
Iteration 6	4424	35	5,8,11,19,29
Iteration 7	4728	33	5,8,11,19,29
Iteration 8	4894	31	5,8,11,19,29
Iteration 9	7844	30	3,8,17,18,43
Combined	46.08	37	5,12,17,35,43

Figure G.5: Model A's Results for DS7

Iteration	Access	Equity	Facilities opened
Iteration 1	1759	123	3,5,13,29,42,61,79,80,81,82
Iteration 2	1759	100	3,5,13,29,42,61,79,80,81,82
Iteration 3	1759	80	3,5,13,29,42,61,79,80,81,82
Iteration 4	1759	60	3,5,13,29,42,61,79,80,81,82
Iteration 5	1759	50	3,5,13,29,42,61,79,80,81,82
Iteration 6	1759	45	3,5,13,29,42,61,79,80,81,82
Iteration 7	1808	40	3,5,13,29,42,54,69,80,81,82
Iteration 8	1890	35	3,5,23,29,42,61,80,81,82,100
Iteration 9	1979	30	1,3,5,42,44,61,80,81,82,100
Iteration 10	1990	29	1,3,5,42,44,69,80,81,82,100
Iteration 11	2001	28	1,3,5,42,44,69,80,81,82,100
Iteration 12	2024	26	1,3,5,42,63,69,80,81,82,100
Iteration 13	2038	25	1,3,5,42,63,69,80,81,82,100
Iteration 14	5612	20	6,18,22,42,51,56,63,80,83,90
Iteration 15	7974	19	6,18,56,57,68,80,83,88,90,100
Combined	20.38	25	1,3,5,42,63,69,80,81,82,100

Figure G.6: Model A's Results for DS11

Appendix H

Model B Results

This section provides the tables showing the results of Model A for DS1 to DS11. Each table shows the accessibility and coverage objective achieved as well as the facilities selected and demand points excluded by the model for each dataset.

Iteration	"r" value	Access	Coverage	Facilities opened	Demand points not covered
Iteration 1	31	11349	509	2,25,28,34,45	10,18
Iteration 2	31	3473	509	2,25,28,34,45	10,18
Iteration 3	30	3898	509	2,25,28,34,45	10,18
Iteration 4	29	2520	498	2,6,28,38,45	10,18,25
Iteration 5	28	2999	498	2,6,28,38,45	10,18,25
Iteration 6	27	6411	498	2,6,20,28,34	10,18,25
Iteration 7	26	4924	489	2,6,22,28,38	10,18,25,36
Iteration 8	25	4533	487	2,18,34,35,45	6,25
Iteration 9	24	4536	487	7,12,23,30,34	6,25
Iteration 10	23	8338	486	3,6,12,22,35	10,11,12,18,25
Iteration 11	22	4792	477	3,7,12,23,30	6,14,25,36
Iteration 12	21	8942	469	3,4,12,22,33	6,10,14,18,25,36
Iteration 13	20	6832	469	3,4,12,22,30	6,10,14,18,25,36
Iteration 14	19	5805	455	3,4,11,12,22	6,10,13,14,15,18,25,36
Iteration 15	18	5281	423	3,4,14,33,30	2,6,10,12,15,18,21,36,49
Iteration 16	17	3980	414	3,4,8,11,50	2,6,10,12,15,18,20,25,31,36,45,49

Figure H.1: Model B's Results for DS4

Iteration	"r" value	Access	Coverage	Facilities opened	Demand Points not covered
Iteration 1	27	8757	541	9,18,21,37,48	5,40
Iteration 2	27	6460	541	9,30,37,45,48	5,40
Iteration 3	26	8256	541	23,30,37,45,48	5,40
Iteration 4	26	6307	541	9,30,37,45,48	5,40
Iteration 5	24	7691	515	22,23,30,45,48	5,35,40,47
Iteration 6	23	9335	495	11,12,17,41,48	2,4,5,13,28,44
Iteration 7	22	4166	458	14,17,24,28,48	2,5,13,35,36,37,38,41
Iteration 8	21	5816	445	15,17,19,38,49	2,5,13,18,27,28,33,36,41,42,44
Iteration 9	20	5596	444	17,19,22,45,46	2,5,13,15,33,35,39,41,44
Iteration 10	19	3487	402	7,17,22,32,45	2,3,5,10,13,15,33,39,40,41,43,48
Iteration 11	18	3343	399	7,15,17,32,45	2,3,4,5,10,13,14,20,22,33,35,36,37,40,41,43,50
Iteration 12	17	2988	371	15,17,24,32,45	2,3,4,5,10,13,14,18,20,22,29,33,35,36,37,40,41,43,47,50

Figure H.2: Model B's Results for DS6