LOCATION OF A CHAIN OF TWO COMPUTER SERVICES AT KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY CAMPUS USING THE P-MEDIAN MODEL

by

Agbitor Kwadzo Nicholas, BSc. (Hons) Mathematics

A Thesis submitted to the Department of Mathematics, Kwame Nkrumah University of Science of Technology

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Faculty of Physical Sciences,

College of Sciences

May, 2009

DECLARATION

I, Agbitor Kwadzo Nicholas, hereby declare that this submission is my own work towards the MSc. Mathematics degree, and that, to the best of my knowledge, it contains no material previously published by another person nor material which has been accepted for the award of any degree elsewhere, except where due acknowledgement has been made in the text.

AGBITOR K. NICHOLAS		
……………………………….	…………………….	………………...
Student	Signature	Date

Certified by:

DR. S. K. AMPONSAH		
……………………………….	…………………….	………………...
	Signature	Date
MR. K. F. DARKWAH		
……………………………….	…………………….	………………...
Supervisors	Signature	Date

Certified by:

DR. S. K. AMPONSAH		
……………………………….	…………………….	………………...
Head of Dept	Signature	Date

ABSTRACT

One of the greatest problems facing both the public and the private sector enterprises is how to locate facilities. In public service oriented siting problems, decision makers have to decide about the location of public services (offices, schools, hospitals, ambulance services, banks, etc.), while emphasizing on the accessibility of public services to people. In the case of most medical emergencies, the risk of lost of life increases with response time. Consequently, one reasonable objective is to minimize response time given a limited budget.

Industrial firms must determine locations for fabrication and assembly plants as well as warehouses. Retail outlets must also locate stores. The cost of serving a retail establishment from a warehouse may depend on the time a driver must spend traveling from the warehouse to the retail store. In this case, the cost depends approximately linearly on the distance between the store and the warehouse. In all cases the right location is paramount.

This piece of work attempts to use the *p*-median model (2-median) to find suitable location for setting up a chain of two computer services at Kwame Nkrumah University of Science and Technology (KNUST) campus.

Different methods were used to locate the suitable sites, but the main one was the Lagrangian algorithm. The result obtained using the Lagrangian suggested that, the two facilities be located at the Republic Hall (node *E*) and the University Hall (node *B*). The optimal objective function value was 1719170 metres. This value gave the demand-weighted distance. It resulted in the average distance of 207 metres. (i.e. the average distance = the demand-weighted distance divided by the total demand; 1719170 m / 8310 $\cong$ 207 m). It implies that, on average, each student would travel a distance of 207 metres from a hall / hostel to the nearby facility.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

DEDICATION

This work is dedicated to my lovely wife Mercy Serwaa, and my beloved children; Grace Worlali Ahiadormey, Vincent Nyamekor Ahiadormey, Benedict Edem Ahiadormey and Norbert Agbenyo Ahiadormey.

ACKNOWLEDGEMENT

I will first and foremost express my sincere gratitude to my Heavenly Father for bringing me this far and also for His love, mercy, grace and protection throughout the course.

I am also very grateful to my academic supervisors Dr. S. K. Amponsah and Mr. K. F. Darkwah for their guidance, useful suggestions, encouragement, support and positive criticism. They have been of great help and source of inspiration to me.

I would also like to thank all the lecturers in the Mathematics Department (KNUST), especially the Head of Department, Dr. E Prempeh for their encouragement.

I cannot forget my lovely wife Mercy Serwaa. My deepest gratitude goes to her for her spiritual and moral support, encouragement and cooperation throughout the course.

My profound gratitude also goes to my dear mother, Madam Odilia Nenemse, and all my siblings for their prayers and moral support.

I would like to sincerely thank all those who provided me with the data for this work; Hall/Hostel officials and Head of Geomatic Department. Without their cooperation, this work would have merely been a dream.

Finally, my sincerest gratitude goes to those who helped me in diverse ways in making this work a success. God richly bless you all.

CHAPTER ONE

1.1 Introduction

When a rhetorical question beloved by both property-conscious tycoons and academic geographers was asked: “What are the three most important things in the success of a new supermarket (or gas station, or housing)?” Quite likely you would hear the answer: “Location, location, location” (Goldman, 2006).

If you ask what to look for in buying a house, any realtor will tell you that there are three things that are important: location, location and location. The theory behind the answer is that the community one chooses to live and the location within that community are likely to affect ones quality of life at least as much as the amenities within ones house. For instance, if ones house is too close to a factory, noise, traffic, and pollution from the factory may degrade ones quality of life. If one lives near a community centre, one may be able to avoid involvement in car pools taking children to and from activities. If one lives within walking distance of the local basic school, ones children need not be bused to school (Daskin, 1952).

In most location problems, we are interested in locating desirable facilities. In other words, value increases, the closer the facilities are to the people or goods being served. Ambulances, fire stations, schools, hospitals, post offices, warehouses, and production plants are all considered desirable facilities in this sense. Some facilities, however, are considered undesirable in the sense that most people want them located as far away as possible. Typically, such facilities are either noxious (i.e. posing a health or welfare hazard to people), or obnoxious (i.e. posing a threat to people's lifestyles). Hazardous waste sites, landfills, incinerators, nuclear missile silos, and prisons fall into this category (Erkut et al, 1989).

Location problem is concerned with the locating of one or more facilities, in some space (i.e. at the node or on the edge of a network), so as to optimize some specific criteria. Often these criteria are linked with distribution costs or providing optimal access for the customers of the facility in question. This does not necessarily follow however, when facilities produce some undesirable or obnoxious effect. Here, the risk to the local population far outweighs any benefits of close sitement of the facility. This therefore, causes the location formulation to change to that of minimizing risk or equivalently maximizing some distance function to the population centers (Amponsah, 2003).

The problem of siting a single facility on a network, so as to maximize the minimum Euclidean distance along the arcs/edges of the network, from the nodes present (representing population centres or existing facilities) is a trivial case of the obnoxious location question. The problem becomes more difficult, when these distances do not have to lie on the arcs/edges of the network. This allows for the spread of any pollution that is emitted, across the plane in which the network lies. The underlying assumptions of this formulation, lies in the fact that the pollution decreases with distance uniformly about the facility from which it originates (Amponsah, 2003).

In the next chapter, the sitement of facilities will be discussed. This will commence with the study of non-obnoxious (desirable) facility location, within three possible solution spaces namely, a discrete sets of points, the network and the plane. For each case, different objective functions will be proposed based on the minisum and minimax principles and their solution techniques discussed.

In today's modern society, the quality of life is often defined by the number of facilities available to the populace. From dry-cleaners to garages, from fire-stations to football stadia, all provide a service and so can be defined as facilities. In other words, a facility can be considered as a physical entity that provides a service. These facilities can be classified into three categories; desirable (non-obnoxious), semi-obnoxious or obnoxious (non-desirable). Most services are provided by desirable or non-obnoxious facilities.

Non-obnoxious (desirable) facilities may be supermarkets, warehouses, shops, garages, banks, libraries, etc. As the customer needs access, of some sort, to the facility providing the service, it is beneficial if these facilities are sited close to the customers that they will be serving. This implies that the customer has better access to the facility. Sometimes, a facility may produce a negative or undesirable effect. This effect may be present even though a high degree of accessibility is required of the facility. For example, a stadium provides entertainment and so requires a large amount of access to enable supporters to attend a game. On the other hand, on match days, local non-football funs would have to contend with the noise and the traffic generated. This generation of noise is unpleasant for locals and therefore undesirable. The combination of the two makes this facility a semi-obnoxious. Another example is the waste disposal sites. Here, access is needed to deposit the waste produced by local population. Conversely, the disposal site may be offensive to look at, and also it emits offensive odour. These two contradicting points cause the disposal site to be defined as a semi-obnoxious facility. Other examples of semi-obnoxious facilities are ambulance and fire stations. A facility is defined as obnoxious facility if its undesirable effect far outweighs its accessibility. Some examples are nuclear power stations, military installations and pollution-producing industries. Although necessary to society, these facilities are undesirable and often dangerous to the surrounding inhabitants so lowering local house prices and quality of life (Amponsah, 2003).

1.2 Background Study

Computer services at the KNUST Campus comprise the internet services, photocopying, typing, printing and repair services. These services are in short supply compare to the number of people (students) they serve. The total number of resident students of KNUST in 2008/2009 academic year is 8310.

Each of the six halls of the university has an internet café which has an average of thirty (30) computers. The main Information and Communication Technology (I.C.T.) department also has two hundred and fifty (250) computers. In addition, few departments of the university have a limited number of computers for use by students.

There is only one typing and printing facility at the Unity Hall. There are also two undeveloped repair facilities at the Independence Hall the University Hall.

The above data showed that the facilities available at KNUST campus are woefully inadequate, and therefore there is the need to establish more.

1.3 Statement of Problem

One of the greatest problems facing both the public and the private sector enterprises is how to locate facilities. People site their facilities anywhere, anyhow without first considering how close that facility will be to people in the community.

This work therefore seeks to find the optimal sites to locate a chain of two computer services at the Kwame Nkrumah University of Science and Technology (KNUST) Campus using the p-median model.

1.4 Purpose of the Study

The computer, and for that matter, the internet is an indispensable tool as far as research work is concerned. In addition, computers are needed for typing and printing of project works. Those who have computers also might need repair services sometimes. Unfortunately, these facilities are in short supply on our university campuses. Due to the ever-increasing population of students at KNUST Campus, there is the need to establish more combined services of typing/printing, repairs and internet services to enable students do their research with ease.

The main aim of this work is to find the optimal sites to establish a chain of two computer services at KNUST Campus using the p-median model.

1.5 Objectives

The objectives of this study:

1. To locate a chain of two computer services at suitable sites at KNUST Campus for combined services of typing/printing and repairs.
2. To locate the facilities at suitable sites so that the average distance covered by students from the halls/hostel to the nearest of the *p* facilities be minimized.
3. To recommend to computer service providers on the best locations for expansion of their enterprise.

1.6 Methodology

The location problem was modeled as 2-median problem. Below are the steps used to solve it:

- Data on students population of the halls/hostel, and the inter-hall/hostel road distances were collected and used.

- Dijkstra’s algorithm was used to find the distance matrix, d(i, j) for all pairs shortest path.
- Myopic algorithm was used to estimate the demand-weighted distance which was then used as the upper bound (UB) for the Lagrangian algorithm.
- Lagrangian algorithm was used in optimal location to find the two sites for the computer services.
- Materials were obtained from the KNUST library, Mathematics library and Internet.

1.7 Thesis Organization

Chapter one: Introduction

Chapter two: Literature Review

Chapter three: Network Location Models

Chapter four: Data Collection and Analysis

Chapter five: Conclusion and Recommendation.

CHAPTER TWO

2.0 Literature Review

2.1 Desirable Facilities

The problem of siting *p* facilities ($p \in \mathrm{N}$) in some universe, so as to satisfy a given set of criteria has posed many questions over the last few decades:

i. What is the universe to be considered?

ii. What assumptions can be made to simplify the problem without distorting the solution set radically?

iii. What objective(s) is/are to be optimized?

These questions have resulted in the emergence of many different formulations to the fundamental location problems. As one would expect, the more accurately a model reflects 'real life' situations, the more complex the problem becomes. In this chapter, three different universes will be addressed. The whole essence of the siting problem is to locate several facilities (e.g. supermarkets, sports centres or industrial parks) within an environment so as to optimize their location. This optimization may vary depending on the particular objective function chosen. This function could be any one of the following:

i. minimize average travel time or cost,

ii. minimize average response time,

iii. minimize maximum travel time or cost, and

iv. maximize net income.

The basic question presented here is that of minimizing the weighted distance between *p* facilities ($p \in N$) and the set of customers, so as to maximize their availability to users or minimize transportation costs from the source node (population centre) to the destination node (facility).

2.2 The Universe to be Considered

The first universe to be considered is that of the entire plane, entitled the planar location problem. Here the set of points making up the entire plane is the set of feasible solutions. For this basic formulation, the planar model assumes direct distance metric e.g. Euclidean or rectilinear. Unfortunately, in such a problem, potential customers will normally travel along the arcs/edges of the network, road or rail. This prompts the formulation of the network location model, where the facilities may be positioned on the network. Distances are then reformulated to be the shortest path linking facilities and customers. Again this causes a problem, and what happens if the optimal location is not feasible, which may be highly probable in a densely populated area? This leads to a discrete formulation where the set of possible locations is a finite set of points.

2.2.1 Planar Location Models

A planar location model involves the location of p new facilities ($p \in \mathrm{N}$) within a feasible plane, so as to minimize some cost or distance from each new facility to the other new facilities and any existing locations within the plane.

Assumptions

Before any formulation of the above can be established a set of assumptions must be made:

i. Any point in the plane can be a member of the feasible solution.

ii. Each facility can be approximated by a point, i.e., it has no area.

iii. A subset of the earth's surface can be approximated by a plane.

The above assumptions immediately raise several questions about accuracy. Assumption (i) does not allow for the occurrence of infeasible area within the plane, such as property owned by other organizations, natural barriers are inaccessible sites. In these cases the model assumes that a site close to the optimal may be chosen with no loss of satisfaction. Assumption (ii) states that the feasible plane is infinitely bigger than the area taken by a facility. This is obviously unrealistic and may affect the results if the feasible area is on a very local scale and the potential facilities require large site area. Assumption (iii) assumes that the feasible set is small enough so that the spherical curve of the sphere does not alter the shortest distance.

2.2.2 Network Location Models

In the planar model, the set of feasible location points is made up of the entire plane and a distance metric is used as a measure of the accessibility to and from each facility. This may be a road or rail system, or a set of flight or shipping routes. It may therefore be preferred for sitement of the facilities to occur on the links or nodes of this network, thus implying the replacement of the distance metric, with actual network distances.

Assumptions

To adopt the model, the set of assumptions made above must first be modified:

i. Each facility can be approximated by a point i.e. it has no area.

ii. Network distances between points are defined as shortest path distances which can be computed using Disjkstra's algorithm or Floyd's algorithm.

iii. Any point in the network can be a member of the feasible solution.

These assumptions are very similar to those of the planar model and result in very similar formulations to those presented previously. However, if an additional assumption is introduced, that is, assumption (iv), each facility is of similar type and so a customer will travel to only the closest facility, a subset of the minimax /minisum formulation is addressed.

2.2.3 Discrete Location Models

Planar and network location models have several limitations, in that:

i. Every point in the plane or network is a candidate solution.

ii. Fixed costs for siting individual facilities at a particular point are ignored (assumed to be independent of the location chosen and so does not affect the optimal solution).

These limitations are confronted when the solution set is reduced to that of a finite number of candidate solutions. Then each candidate can be assigned an individual location cost, which can, in turn, be incorporated into the objective function.

So far it has been assumed that the facilities to be sited give a particular service and are in some sense desirable. However, there are facilities which give services to the public but they are undesirable i.e., closeness to the public is not wanted.

In the next section we shall put forward some location methods.

2.3.0 Some Location Methods

Location refers to a place or a position where something can be found. There are many factors, both quantitative and qualitative, to consider in choosing a location. Some of these factors are more important than others, so people use weighting to make the decision process more objective. There are three main location methods. These include the factor rating method, the centre of gravity method and the location break-even analysis.

2.3.1 The Factor Rating Method

The factor rating method is popular because a wide variety of factors, from education to labour skills to recreation, can be objectively included. The factor rating method has six steps:

i. develop a list of relevant factors,

ii. assign a weight to each factor to reflect its relative importance in the company's objectives,

iii. develop a scale for each factor (e.g. 1 to 10 or 1 to 100),

iv. assign a score to each location for each factor, using the scale in step (iii),

v. multiply the score by the weights for each factor and total the score for each location,

vi make a recommendation based on the maximum point score, considering the results of quantitative approaches as well.

When a decision is sensitive to minor changes, further analysis of either the weighting or the points assigned may be appropriate. Alternatively, management may conclude that these intangible factors are not the proper criteria on which to base a location decision. Managers therefore place primary weight on the more quantitative aspects of the decision (Amponsah, 2006).

2.3.2 The Centre of Gravity Method

The centre of gravity method is a mathematical technique used for finding the location of a distribution centre that will minimize distribution costs. The method takes into account the location of facilities, the volume of people moved to and from those facilities, and transportation costs in finding the best location for a distribution centre. The first step in the centre of gravity method is to place the location on a coordinate system. The origin of the coordinate system in the skill used is arbitrary, just as long as the relative distances are correctly represented. This is done easily by placing a grid over an ordinary map of the location in question. The centre of gravity is determined by equations (1) and (2):

$$C_x = \frac{\sum d_{ix} W_i}{\sum W_i} \quad \text{………………………..} \quad (1)$$

$$C_y = \frac{\sum d_{iy} W_i}{\sum W_i} \quad \text{…………………………} \quad (2)$$

Where C_x = x - coordinate of the centre of gravity

C_y = y - coordinate of the centre of gravity

d_{ix} = x - coordinate of location *i*

d_{iy} = y - coordinate of location *i*

W_i = volume of people moved to and from location *i*

2.3.3 The Location Break-Even Analysis

The location break-even analysis is the use of cost-volume analysis to make an economic comparison of location alternatives. By identifying fixed and variable costs and graphing them for each location. Location break-even analysis can be done mathematically or graphically. Both methods can help us determine the lowest cost. The graphical approach has the advantage of providing the range of volume over which each location is preferable. There are three steps in location break-even analysis. These are:

i. determine the fixed and variable costs for each location,

ii. plot the cost for each location, with cost on the vertical axis of the graph and production volume on the horizontal axis,

iii. select the location that has the lowest total cost for the expected production volume.

2.4 The Uncapacitated Facility Location Problem

The uncapacitated facility location problem, UFL problem, is used to model many applications. Some applications are: bank account allocation, clustering analysis, lock-box location, location of offshore drilling platforms, economic lot sizing, machine scheduling and inventory management, portfolio management and the design of communication networks.

For example, the bank account location problem arises from the fact that the clearing time for a cheque depends on a city *i* where it is cashed and the city *j* where the paying bank is located. A company that pays bills by cheque to clients in several locations finds it useful to open accounts in several strategically located banks. It pays the bill to the client in city *i* from a bank in city *j* that maximizes the clearing time.

The UFL problem consists of:

i. a set J = { 1, 2, 3, …, n } of potential sites for locating facilities,

ii. a set I = { 1, 2, 3, …, m } of clients whose demands need to be served by the facilities,

iii a profit c_{ij} for each facility $j \in J$ and client $i \in I$. The profit is made by satisfying the demand of client *i* from a facility *j*,

iv. a fixed nonnegative cost f_j for each facility $j \in J$ to be used in setting up the facility *j*.

The uncapacitated facility location problem is to select a given number *S* of the *J* facilities $(S \subseteq J)$ And to assign each client to exactly one facility such that the difference between the profits from *S* and the fixed costs is maximized.

The integer linear programming (ILP) formulation is given by

$$\text{Maximized} \quad Z = \sum_{i \in I} \sum_{j \in J} c_{ij} y_{ij} - \sum_{j \in J} f_j x_j \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

Subject to : $\sum_{j \in J} y_{ij} = 1 \quad \forall i \in I$ …………………… (2)

$y_{ij} \le x_j \quad \forall i \in I, \forall j \in J$ ………………… (3)

$x_j \in \{0,1\} \quad \forall j \in J$ ……………………..... (4)

$y_{ij} \in \{0,1\} \quad \forall i \in I, \forall j \in J$ …………………… (5)

Represent each potential site $j \in J$ by x_j and a facility is opened at site $j \in J$ if $x_j = 1$ otherwise, $x_j = 0$. Represent the satisfaction of demand of client $i \in I$ from facility $j \in J$ by y_{ij}. Demand of client *i* is served by the facility at site *j* if $y_{ij} = 1$ otherwise, $y_{ij} = 0$.

Equation (3) may be relaxed to obtain $\sum_{i \in I} y_{ij} \le m x_j \quad \forall j \in J$ …………… (6).

This is the result of summing the *m* constraints $y_{ij} \le x_j$. In such case the optimization problem is called weak integer programming (WIP).

When linear programming relaxation method is used to solve the ILP, the problem is called strong linear programming relaxation (SLPR). Also, when linear programming relaxation method is used to solve the WIP, the problem is known as weak linear programming relaxation (WLPR). In such case $x_j \in \{0,1\}$ is replaced by $0 \le x_j \le 1$ and $y_{ij} \in \{0,1\}$ replaced by $y_{ij} \ge 0$.

SLPR is formulated as

(SLPR) Max $Z = \sum_{i \in I} \sum_{j \in J} c_{ij} y_{ij} - \sum_{j \in J} f_j x_j$ ………………………. (7)

Subject to $\sum_{j \in J} y_{ij} = 1 \quad \forall i \in I$ ………………………. (8)

$y_{ij} \le x_j \quad \forall i \in I, \forall j \in J$ ……………… (9)

$x_j \le 1 \quad \forall j \in J$ …………………….. (10)

$$x_j \geq 0 \qquad \forall j \in J \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \quad (11)$$

$$y_{ij} \geq 0 . \qquad \forall i \in I, \forall j \in J \quad \ldots\ldots\ldots\ldots\ldots... \quad (12)$$

The dual formulation to the SLPR is obtained by using arguments for the simplex of LP method. The coefficients of the objective function of the dual are obtained from equations (8), (9) and (10). They are *m* (from *I*) and *n* (from *J*). Introducing new variables u_i and t_j we have the objective function;

$$\text{Min} \quad W = \sum_{i \in I} u_i + \sum_{j \in J} t_j$$

The RHS of the corresponding equations (8) and (9) are f_j and c_{ij}. We thus have the constraints

$$t_j - \sum_{i \in I} w_{ij} \geq - f_j \qquad \forall j \in J$$

$$u_i + w_{ij} \geq c_{ij} \qquad \forall i \in I, \forall j \in J$$

$$u_i \quad \text{free} \qquad \forall i \in I$$

$$w_{ij} \geq 0 \qquad \forall i \in I, \forall j \in J$$

$$t_j \geq 0$$

The dual of SLPR is, therefore, given as;

$$\text{(Dual SLPR)} \quad \text{Min} \quad W = \sum_{i \in I} u_i + \sum_{j \in J} t_j \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots. \quad (13)$$

$$\text{Subject to} \quad t_j - \sum_{i \in I} w_{ij} \geq - f_j \qquad \forall j \in J \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots. \quad (14)$$

$$u_i + w_{ij} \geq c_{ij} \qquad \forall i \in I, \forall j \in J \quad \ldots\ldots\ldots\ldots\ldots. \quad (15)$$

$$u_i \ \text{free} \qquad \forall i \in I \quad \ldots\ldots\ldots\ldots\ldots. \quad (16)$$

$$w_{ij} \geq 0 \qquad \forall i \in I, \forall j \in J \quad \ldots\ldots\ldots\ldots\ldots \quad (17)$$

$$t_j \geq 0 \qquad \forall j \in J \qquad \ldots\ldots\ldots\ldots\ldots\ldots \quad (18)$$

Instead of solving the relaxed integer programming problem we reduce the dual problem to condensed form and solve by heuristic methods. Suppose all the variables u_i have been given fixed values then from equation (15) and (17) $u_i + w_{ij} \geq c_{ij}$ and $w_{ij} \geq 0$.

It implies $w_{ij} = (c_{ij} - u_i)^+ \qquad \forall i \in I, \forall j \in J \quad \ldots\ldots\ldots\ldots\ldots \quad (19),$

Where, for any expression a, $a^+ = \max(a, 0)$.

Since the u_i's have been fixed and we are minimizing, we must assign the t_j minimum values such that from equation (14) and (18)

$t_j - \sum_{i \in I} w_{ij} \geq -f_j \quad (\forall j \in J) \quad \ldots\ldots\ldots\ldots\ldots\ldots$ (20a) and $t_j \geq 0 \quad (\forall j \in J)$ are satisfied.

This implies that $t_j \geq (\sum_{i \in I} w_{ij} - f_j)^+ \qquad \forall j \in J \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad$ (20b).

Substituting for w_{ij} we get

$$t_j = (\sum_{i \in I} (c_{ij} - u_i)^+ - f_j)^+ \qquad \forall j \in J.$$

We thus have the first condensed form;

$$\text{(CD1)}\ W = \min_{u_1,\ldots,u_m} \{\sum_{i \in I} u_i + \sum_{i \in I} (\sum_{i \in I} (c_{ij} - u_i)^+ - f_j)^+\} \quad \ldots\ldots\ldots\ldots\ldots \quad (21)$$

Suppose a u_q in equation (21) is increased by a small number $\varepsilon > 0$, then the increased effect in the $\sum_{i \in I} u_i$ in the first term is cancelled by the effect of $\sum_{i \in I} u_i$ in the bracket and the objective is not changed. But the increase of ε in u_q decreases the second term which is equal to t_j by the same amount. Under such circumstances there will always exist an optimal solution to the dual SLPR with $t_j \leq 0 \qquad \forall j \in J \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots. \qquad (22)$

we use this information to change u_i until t_j is at zero.

Equation (21) is reformulated by splitting the two terms and adding the constraint $u_i \le \max_{j\in J}\{c_{ij}\}$ we have;

$$\text{(CD2)}\ \min W = \sum_{i\in I} u_i \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (23)$$

$$\text{Subject to} \quad \sum_{i\in I}(c_{ij}-u_i)^{+} - f_j \le 0 \qquad \forall j\in J \quad \ldots\ldots\ldots\ldots\ldots \qquad (24)$$

$$u_i \le \max_{j\in J}\{c_{ij}\} \qquad \forall i \in I \quad \ldots\ldots\ldots\ldots\ldots.. \qquad (25)$$

2.5 Greedy Heuristic

We extract heuristic function from the relaxed linear programming and compute for the optimal solution directly and not through solving for the decision variables. The heuristic simultaneously find a candidate solution $x_j \in \{0,1\}$, $y_{ij} \in \{0,1\}$ to the UFLP problem instance.

Start with an empty set *S* of opened facilities and add to *S* a new facility $j \in J \setminus S$ at each step so that there is maximum improvement in the objective value when a set of S facilities have been opened is given by $Z(S) = \sum_{i\in I} u_i(S) - \sum_{j\in S} f_j$.

Current opened facilities are in the set *S*, where $S \subseteq J$ and $u_i\,(S)$ is the maximum profit obtained from serving client *i* using facilities in *S*, $i \in I$.

$p_j(S) = z(S \cup \{j\}) - z(S) = \sum_{i\in I}(c_{ij} - u_i(S))^{+} - f_j$, $\quad j \in J \setminus S$, where $p_j(S)$ is the change in objective value when new facility is opened at *j* that could be added to *S*.

For each computation step, put $u_i(S) = \max_{j\in S}\{c_{ij}\}$ with $u_i(\phi) = 0$ (Cornuejols et al, 1990).

Example: It is proposed that facilities be opened at a maximum of six (6) potential sites to serve clients at four (4) locations. The cost of setting up the facilities on the different sites are f = [3, 2, 2, 2, 3, 3]. The gain in serving the demand of client *i* by an opened facility *j* is given by $C = [C_{ij}]$.

$$C = \begin{bmatrix} 6 & 6 & 8 & 6 & 0 & 6 \\ 6 & 8 & 6 & 0 & 6 & 6 \\ 5 & 0 & 3 & 6 & 3 & 0 \\ 2 & 3 & 0 & 2 & 4 & 4 \end{bmatrix}$$

Iteration 1: S is currently an empty set.

It implies $S_0 = \phi$, $z(S_0) = 0$, $u(S_0) = [u_1, u_2, u_3, u_4] = [0, 0, 0, 0]$

Addition of a facility to S_0, we have;

$$p_j(S_0) = \sum_{i \in I} (c_{ij} - u_i(S_0))^+ - f_j = \sum_{i \in I} (c_{ij})^+ - f_j \;, \quad j \in J \setminus S_0$$

$p_1(S_0)$	$p_2(S_0)$	$p_3(S_0)$	$p_4(S_0)$	$p_5(S_0)$	$p_6(S_0)$
19-3 =16	17-2 =15	17-2 =15	14-2 =12	13-3 =10	16-3 =13

$w(u(S_0)) = \sum_{j \in J} p_j$ = 0 + 81 = 81, since $z(S_0) = 0$ (sum of u_i is zero).

S_1 = { 1 }, since $p_1(S_0)$ = 16 > 0 is the maximum among the lot.

Iteration 2: S_1 = { 1 }, $z(S_1) = 16$, $u(S_1) = [u_1, u_2, u_3, u_4]$ = [6, 6, 5, 2]

Addition of another facility to S_1 = { 1 }

$$p_j(S_1) = \sum_{i \in I} (c_{ij} - u_i(S_1))^+ - f_j, \quad j \in J \setminus S_1, j = 1 \text{ is left out.}$$

$p_2(S_1)$	$p_3(S_1)$	$p_4(S_1)$	$p_5(S_1)$	$p_6(S_1)$
3- 2 = 1	2- 2 = 0	1- 2 = -1	2- 3 = -1	2- 3 = -1

$p_2(S_1)$, implies second column for each row (i) and corresponding u_i. $g_j = \sum_{i \in I}(c_{ij} - u_i)^+$
$c_{12} - u_1 = 6 - 6 = 0$, $c_{22} - u_2 = 8 - 6 = 2$, $c_{32} - u_3 = 0 - 5 = -5$, $c_{42} - u_4 = 3 - 2 = 1$

Neglect the negative values and add non-negative values.

It implies 0 + 2 + 1 = 3, $g_2 - f_2 = 3 - 2 = 1$

$p_3(S_1)$, implies third column. $g_3 = \sum_{i \in I}(c_{ij} - u_i)^+ \Rightarrow c_{13} - u_1 = 8 - 6 = 2$, $c_{23} - u_2 = 6 - 6 = 0$

$c_{33} - u_3 = 3 - 5 = -2$, $c_{43} - u_4 = 0 - 2 = -2$, now $g_3 = 2$, $\Rightarrow$ $g_3 - f_3 = 2 - 2 = 0$.

$w(u(S_1))$ = 19 + 1 = 20, because $\sum_{i \in I} u_i$ = 19 and $p_2(S_1)$ = 1.

S_2 = {1, 2} since $p_2(S_1)$ = 1 > 0 is maximum.

Iteration 3: S_2 = {1, 2}, $z(S_2)$ = 16 + 1 = 17, $u(S_2)$ = [6, 8, 5, 3]

Addition of another facility to S_2 = {1, 2}

$p_j(S_2) = \sum_{i \in I}(c_{ij} - u_i(S_2))^+ - f_j$, $j \in J \setminus S_2$, j = 1, 2 are left out.

$p_3(S_2)$	$p_4(S_2)$	$p_5(S_2)$	$p_6(S_2)$
2 - 2 = 0	1-2 = -1	1 – 3 = - 2	1 – 3 = - 2

$p_3(S_2)$, implies third column. $g_3 = \sum_{i \in I}(c_{ij} - u_i)^+ \Rightarrow c_{13} - u_1 = 8 - 6 = 2$, $c_{23} - u_2 = 6 - 8 = -2$

$c_{33} - u_3 = 3 - 5 = -2$, $c_{43} - u_4 = 0 - 3 = -3$, now $g_3 = 2$, $\Rightarrow$ $g_3 - f_3 = 2 - 2 = 0$.

$w(u(S_2)) = 22 + 0 = 22$, because $\sum_{i \in I} u_i = 22$ and $p_2(S_2) = 0$.

Stop since $p_j(S_2) \le 0$, for j = 3, 4, 5, 6. That is, $\forall j \in J \setminus S_2$

The solution is $S = \{1, 2\}$ with objective $z(S) = 17$. The best upper bound is given by the dual greedy value $w(u) = w(u(S_1)) = 20$ (Cornuejols et al, 1990).

2.6 Network-Based Algorithms

2.6.1 Shortest Path Problems

Shortest path problems are the most basic and the most commonly encountered problems in the study of transportation and communication networks. There are many types of shortest-path problems. For instance, we may be interested in determining the shortest path (i.e. economical path or fastest path or minimum-fuel-consumption path) from one specific node in the network to another specific node. We may also need to find the shortest path from a specific node to all other nodes. The shortest paths between all pairs of nodes in a network are required in some problems. Sometimes, one wishes to find the shortest path from one given node to another given node that passes certain specified intermediate nodes.

There are instances where the actual shortest path is not required, but only the shortest distance. In the next section, we shall look at the two most important shortest-path problems:

i. how to determine distance (a shortest path) from a specific node s to another specific node t,

ii. how to determine distances (a path) from every node to every other node in the network.

2.6.2 Dijkstra's Algorithm

The Dijkstra's algorithm finds the shortest path from a source *s* to all other nodes in the network with nonnegative lengths. It maintains a distance label *d(i)* with each node *i,* which is an upper bound on the shortest path length from the source node to any other node *j.* At any intermediate step, the algorithm divides the nodes of the network under consideration into two groups: those which it designates as permanently labeled (or permanent), and those which it designates as temporarily labeled (or temporal). The distance label to any permanent node represents the shortest distance from the source node to that node. The fundamental idea of the algorithm is to find out from source node *s* and permanently labeled nodes in the order of their distances from the node *s*. Initially, node *s* is assigned a permanent label of zero (0) and each other node *j* a temporary label equal to infinity.

At each iteration, the label of a node *i* is its shortest distance from the source node along a path whose internal nodes (i.e. nodes other than s or the node *i* itself) are all permanently labeled. The algorithm selects a node *i* with the minimum temporary label (breaking ties arbitrarily), makes it permanent and reaches out from that node (i.e. it scans all the edges coming out from the node *i* to update the distances label of adjacent nodes). The algorithm terminates when it has designated all nodes permanent.

2.6.3 All-Pair Shortest Path Problem

The shortest path between two nodes might not be a direct edge between them, but instead involve a detour through other nodes. The all-pairs shortest path problem requires that we determine shortest path distances between every pair of nodes in a network.

2.6.4 Floyd-Warshall Algorithm

The Floyd-Warshall algorithm obtains a matrix of shortest path distance within $0\{n^3\}$ computations. The algorithm is based on inductive arguments developed by an application of a dynamic programming technique.

Let $d^k(i, j)$ represent the length of the shortest path from node *i* to node *j* subject to the condition that this path uses the nodes 1, 2, …, k-1 as internal nodes. Clearly, $d^{k+1}(i, j)$ represent the actual shortest path distance from the node *i* to *j*. The algorithm first computes $d^1(i, j)$ for all node pairs *i* and *j*. Using $d^1(i, j)$, it then computes $d^2(i, j)$ for all node pairs *i* and *j*. It repeats this process until it obtains $d^{k+1}(i, j)$ for all node pairs *i* and *j*, then it terminates. Given $d^k(i, j)$, the algorithm computes $d^{k+1}(i, j)$ using $d^k(i, j) = \min\{d^k(i,k), d^k(k, j)\}$. The Floyd-Warshall algorithm remains of interest because it handles negative weight edges correctly.

2.7 Exact Verses Heuristic Method

Here, a brief discussion would be made about some exact and some heuristic search methods commonly used in Operational Research (OR)/Management Science (MS).

Real-world problems are difficult to solve for the following reasons:

i. The size of the search space: The number of possible solutions in the search space is so large as to forbid as exhaustive search for the best answer. For instance, a ten-city Traveling Salesman's Problem (TSP) has about 181,000 possible solutions, a twenty-city TSP has about 10^{16} possible solutions and a fifty-city TSP has about 10^{62} possible solutions.

ii. Modeling the problem: Whenever a problem is solved, we realize that we are, in reality, finding the solution to a model of the problem. Most models could represent a specification of a real-world problem; otherwise they would be as complex and unwieldy as the natural setting itself. The process of problem solving consists of two separate general steps: (a) creating the model of the problem, and (b) using that model to generate a solution. The 'solution' is only a solution in terms of the model. If the model has a high degree of accuracy, we can have confidence that the solution would be meaningful. In contrast, if the model has too many unfulfilled assumptions and rough approximations, the solution may be meaningless. In this case to get any solution, we have to introduce simplifications that make the problem tractable.

iii. Change over time: real-world problems often do change over time. Some may change before modeling, or while the solution is being derived, or after the execution of the solution. We need to be sure that the model reflects the current knowledge about the problem.

iv. Almost all real-world problems pose constraints and if we violate the constraints we cannot implement our solution. There are two types of constraints: namely hard constraints (these are impossible to violate, as the solution becomes redundant) and soft constraints (desirable but could be violated). After getting the right constraints for a problem, we are then left with the problem of searching for the best assignment (i.e. the solution that is feasible and minimizes the evaluation function for the soft constraints.

Suppose we have found a feasible solution which does not do well with regards to the soft constraints, we apply some variation operators to this solution with regards to the soft constraints, but in so doing, we generate a solution that violates at least one hard constraints. We must choose to discard the solution for it is infeasible, or we might see if we can repair it to generate infeasible solution that still handles the soft constraints as well. Either way, it is typically a difficult job.

2.8 Exact Methods

There are many classic algorithms that are designed to search spaces for an optimum solution. The classic methods of optimization fall into two disjoint classes:

i. Algorithms that only evaluate complete solutions.

ii. Algorithms that require the evaluation of partially constructed or approximate solutions. Whenever an algorithm treats complete solutions, we can stop it at any time and will always have at least one potential answer that we can try. In contrast, if we interrupt an algorithm that works with partial solutions, we might not be able to use any of the results at all. We can often decompose the original problem into a set of smaller and simpler problems. The idea is that in solving each of these simpler problems, we can eventually combine the partial solutions to get answer for the original problem. This is the concept used in dynamic programming.

In the next sections we will look at some of the exact methods like exhaustive search, integer programming (cutting plane, branch-and bound) and dynamic programming.

2.9 Exhaustive Search

Exhaustive search checks each and every solution in the search space until the best solution has been found. That means if we do not know the value that corresponds to the evaluated worth of the best solution, there is no way to be sure that we have found the best solution using the exhaustive search unless we examine every solution. Note that the size of the search space of real-world problems of even modest size can be too large to deal with. But exhaustive algorithms are simple; the only requirement is to generate every possible solution to the problem systematically.

2.10 Integer Programming-Based Techniques

Problems in which the decision variables are discrete, where the solution is a set or sequence of integers or other discrete objects are known as combinatorial problems. Examples are the assignment problem, 0-1 knapsack problem, the set covering problem and the vehicle routing problem. Combinatorial problems have close links with linear programming (LP) and most of the early attempts to solve them used developments of LP methods.

Integer programming deals with the solution of mathematical programming problems in which some or all the variables can assume non-negative integer values only. An integer program is called mixed or pure depending on whether some or all the variables are restricted to integer values. Integer programming is quite similar to LP except for the restriction that variables take only integer values. One might therefore suppose that such an integer program could be solved by simply ignoring the integral requirement, solving the LP and rounding of any non-integer solution component to the nearest integer. The LP that results from ignoring the integer constraint is called the linear programming relaxation (continuous) of the integer program. The solution of the LP provides a lower bound on the optional objective value for the integer problem. Several algorithms have been developed for the integer program, but none of these methods is uniformly efficient from the computational perspective, particularly as the size of the problem increases.

Many integer programming problems that arise in practical settings have the special property that, some or all of their variables are restricted to take on only values 0 and 1. Such variables are called 0-1 variables; they often arise naturally in the formation of problems that involve yes-or no decisions.

2.11 Cutting Plane Algorithm

One of the methods used in solving integer programming problems is the cutting plane method. This method, which is primarily developed for integer linear problems, starts with the continuous optimum. By systematically adding special “secondary” constraints, which essentially represent necessary conditions for integrality, the continuous solution space is gradually reduced until its associated continuous optimum extreme point satisfies the integer conditions.

This method cuts (eliminates) certain parts of the solution space that do not contain feasible integer solutions of the original problem.

The idea of the cutting plane algorithm is to change the convex set of the solution space so that the appropriate extreme point becomes all integers. Such changes in the boundaries of the solution space should result still in a convex set. If a cutting plane algorithm fails to solve a given instance, we are left with several options. One option is to use the solution cost of the final LP relaxation, which is a (typically good) lower bound on the optimal value, to assess the quality of a known feasible solution found by any heuristic method. Another option is to feed the final (typically strong) linear relaxation into a classical Branch-and-Bound algorithm for integer problems, which is described below (Amponsah, 2003).

2.12 Branch-and-Bound

The Branch-and-Bound method solves the integer problem by considering its continuous version. This method applies directly to both the pure and the mixed problems. In general, the idea of the method is first to solve problem as a continuous model (linear program). Suppose that x_r is an integer constrained variable whose optimum continuous value x_r^* is fractional, it can be shown that the range $[x_r^*] < x_r < [x_r^*] + 1$ cannot include any feasible integer solution. Consequently, a feasible integer value of x_r must satisfy one of the following conditions: $x_r \leq [x_r^*]$ or $x_r \geq [x_r^*] + 1$. These two conditions when applied to the continuous model results in two mutually exclusive LP problems. In this case it is said that the original problem is branched into two sub-problems. Actually the branching process deletes parts of the continuous space that do not include integer points by enforcing necessary conditions for integrality. Each sub-problem may be solved as linear program (using the same objective function of the original problem). If its optimum is feasible with respect to the integer problem, its solution is recorded as the best one so far available.

In this case it will be unnecessary to further "branch" this sub-problem since it cannot yield a better solution. Otherwise, the sub-problem must be partitioned into sub-problems by again imposing the integer conditions on one of its integer variables that currently has a fractional optimal value. Naturally, when a better integer feasible solution is found for the sub-problem, it should replace the one at hand. This process of branches continues, where applicable, until each sub-problem terminates; either as an integer solution or there is evidence that it cannot yield a better solution. In this case the feasible solution at hand, if any, is the optimum. The efficiency of the computation can be enhanced by introducing the concept of bounding. This concept indicates that if the continuous optimum solution of the sub-problem yields a worse objective value than the one associated with the best available integer solution, it does not pay to explore the sub-problem any further. In this case the sub-problem is said to be fathomed and may henceforth be deleted. The importance of acquiring a good bound at the early stages of the calculations cannot be overemphasized (Aidoo, 2007).

2.13 Dynamic Programming

Dynamic programming works on the principle of finding an overall solution by operating on an intermediate point that lies between where we are now and where we want to go. The procedure is recursive in that each next intermediate point is a function of the point already visited.

A prototypical problem that is suitable for dynamic programming has the following properties:

i. The problem can be decomposed into a sequence of decisions made at various stages.
ii. Each stage has a number of possible states.
iii. A decision takes one from a state at one stage to some state at the next stage.
iv. The best sequence of decision also known as policy at any stage is independent of the decisions made at prior stages.
v. There is a well-defined cost for traversing from state to state across stages. Moreover, there is a recursive relationship for choosing the best decision.

One drawback of dynamic programming, however, is that it can be computationally intensive. The method can however be extended to handle a variety of optimization problems. Dynamic programming algorithms tend to be somewhat complicated to understand. This is because, in practice, the construction of a dynamic program depends on the problem. It is a sort of "artistic" intellectual activity depending, in part, on the specific structure of the sequential decision problem.

2.14 Heuristics

Heuristics is derived from the Greek word "heuriskein" which means to find or discover. A heuristic is a technique which seeks (near optimal) solutions at a reasonable computational cost without being able to guarantee either feasibility, or even in many cases to state how close to optimality a particular feasible solution is. A naïve approach to solving an instance of a combinatorial problem is simply to list all feasible solutions of a given problem, evaluate their objective functions and pick the best. This approach of complete enumeration is likely to be grossly inefficient. It is possible, in principle, to solve any problem in this way but, in practice, it is not because of the large number of possible solutions to any problem of a reasonable size.

In the early days of Operational Research, the emphasis was mostly on finding the optimal solution to a problem, or rather, to a model of a problem which occurred in the real world.

Various exact algorithms were devised which would find the optimal solution to a problem much more efficiently than a complete enumeration. One of the famous methods is the Simplex Algorithm for LP problems. Such exact algorithms may not be able to find optimal solutions to NP-hard problems in a reasonable amount of computing time.

When approaching complex real life problems, four commonly applied methodologies exist. However, this list is not exhaustive as combinations do always exist but are usually difficult to explicitly define:

i. an exact method to the exact (true) problem;

ii. a heuristic method to the exact problem;

iii. an exact method to the (approximate) modified problem;

iv. a heuristic method to the approximation problem.

These rules are put in priority ordering, however, the degree of modification of the problem is a crucial point when dealing with practical problems. The idea is to keep the characteristics of the problem as close as possible to the true problem and try to implement (i) and (ii) (Aidoo, 2007).

2.15 Need for Heuristic

Heuristics are used only when exact methods, which guarantee optimal solutions, are intractable due to (a) either the excessive computational effort, (b) or the risk of being trapped at a local optimum. For such reasons, heuristics become the only way to a company to find reasonably acceptable solutions.

The reasons for accepting and promoting heuristic include:

i. They can be the way forward to producing concrete solutions to large combinatorial problems.

ii. Heuristics can be supported by a graphical interface to help the user in assessing the results more easily.

iii. Management and less specialized users find them reasonably easy to understand and therefore are able to comment and interact with the system.

iv. These are not difficult to w rite, validate and implement.

v. Management can introduce some unquantifiable measures indirectly to see their effect as solutions can be generated reasonably fast.

vi. These methods are suitable for producing several solutions, and not only a single one, from which the user feels more relaxed to choose one or two solutions for further investigation.

vii. The design of heuristics can be considered as an art with a proper insight of a problem.

CHAPTER THREE

3.0 Network Location Models

3.1 Some Models

1. Set Covering Model: this model finds a minimum set of facilities from among a finite set of candidate facilities so that every demand node is covered by at least one facility. For instance, if we wish to locate ambulances so that the maximum response time is under four minutes, then our objective might be to minimize the number of ambulances needed so that all demand nodes are within four minutes (the service standard). Demands are said to be covered if the nearest ambulance is located not more than four minutes away.
2. Maximum Covering Model: this model maximizes the number of demands that can be covered within a specified service standard using a given number of vehicles. It must be noted that, in the covering model, the covering distance/time between a demand and the nearest facility is specified.
3. *P*-Center Problem: this model minimizes the maximum response time (the time between a demand site and the nearest facility/ambulance), using a given number *p*, of vehicles. The *p*-center problem is also known as minimax problem, because we minimize the maximum response time/distance between a demand and the nearest facility to the demand.
4. *P*-Median Problem: this model minimizes the average response time/distance between a demand site and the nearest ambulance, using a given number *p*, of vehicles. (Hakimi, 1964; Hakimi, 1965).

The *p*-median problem is the problem of locating *p* "facilities" relative to a set of "customers" such that the sum of the shortest demand weighted distance between "customers" and "facilities" is minimized.

The model considered in this piece of work is the *p*-median. This is because the objective of this work is to minimize the average distance/time that students would travel from the halls/hostel to the nearest of the *p* facilities.

3.2 The Median Problem

The *p*-median problem is the problem of locating *p* facilities (medians) on a network so as to minimize the sum of all the distances from each demand point to its nearest facility.

This problem may be formulated using the following notation:

Inputs

h_i = demand at node *i*

d_{ij} = distance between demand node *i* and candidate site *j*

P = number of facilities to be located

Decision variables

$$X_j = \begin{cases} 1, \text{ if we locate at candidate site } j \\ 0, \text{ if not} \end{cases}$$

$$Y_{ij} = \begin{cases} 1, \text{ if demands at node } i \text{ are served by a facility at node } j \\ 0, \text{ if not} \end{cases}$$

With this notation, the *P*- median problem may be formulated as follows:

$$\text{Minimize} \qquad \sum_i \sum_j h_i d_{ij} Y_{ij} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \qquad (1)$$

Subject to $\sum_j Y_{ij} = 1 \quad \forall i$ …………………………….. (2)

$\sum_j X_j = P$ …………………………….. (3)

$Y_{ij} - X_j \leq 0 \quad \forall i, j$ ……………………… (4)

$X_j = 0, 1 \quad \forall j$ ……………………… (5)

$Y_{ij} = 0, 1 \quad \forall i, j$ ………………………. (6)

The objective function (1) minimizes the total demand-weighted distance between each demand node and the nearest facility. Constraint (2) requires that, each demand node i be assigned to exactly one facility *j*. Constraint (3) states that exactly *P* facilities are to be located. Constraints (4) link the location variables, X_j and the allocation variables, Y_{ij}. They state that demands at node *i* can only be assigned to a facility at location *j* ($Y_{ij} = 1$) if a facility is located at node *j* ($X_j = 1$). Constraints (5) and (6) are the standard integrality conditions.

The *p*-median formulation given above assumes that facilities are located on the nodes of the network. For *p*-median problem, at least one optimal solution consists of locating *p* facilities on the network's nodes (Hamiki, 1965).

To prove the above assumption, we consider a solution in which at least one facility is located at point P on link (*i*, *j*), a distance of α units from node *i* ($0 < \alpha < d_{ij}$). Let H_i represent total demand that enters link (*i*, *j*) through node *i*, and H_j represent total demand which enters link (*i*, *j*) through node *j*. Assume that $H_i \geq H_j$. By moving the facility from P to node *i* (without altering any of the demand allocations), we change the objective function by $(H_j - H_i)\,\alpha$. Since $H_i \geq H_j$, this quantity is non positive. Thus, making this change in the location of the facility will not degrade the solution. This is sufficient to prove that at least one optimal solution consists of locating only on the nodes of the network. In what follows, we argue that such a change is likely to result in improvement in the objective function. If $H_i \geq H_j$, the quantity $(H_j - H_i)\ \alpha$ will be negative this will improve the objective function. However, this quantity does not account for improvements in the objective function that may be obtained by reallocating demands after the facility is moved. In particular, some demands may now be closer to node i than they were to whatever node to which they had previously been allocated. Also, some of the demands that were originally allocated to the facility located between nodes *i* and *j* and that entered the facility via node *j* may now be closer to some other facility. Reallocating these demands can further reduce the objective function.

3.3 The 1-Median Problem on a Tree

If we focus on locating a single median on a tree, we can find a very efficient algorithm for solving the problem optimally. It must be noted that if a node has at least half of the total demand ($\sum_i h_i = H$), then locating at that node is the optimal location. To prove this, let us consider any solution in which the facility is located at a point on the network (either a node or anywhere on a link) other than the node at which at least half of the total demand occurs. Let the node with half of the total demand be node *A*, with demand h_A. Now consider moving the facility from its current (supposedly optimal) position a distance δ toward node *A*. But since we are moving closer to node *A*, we will be getting closer to at least half of the total demand, *H*. The objective function will, therefore, decrease by at least $\delta\,(2\,h_A - H) \geq 0$. If $h_A > H/2$, such a move will strictly decrease the objective function. Thus, we have shown that at least one optimal solution consists of locating the facility at the node which has at least half of the total demand. This property is independent of whether or not the underlying network is a tree or not. That is, if one node has at least half of the total demand, it is optimal to locate at that node (Daskin, 1952).

Suppose that half or more of the demand does not exist at any single node, as shown in the Fig 3.1 below. Consider locating at some point *X* halfway between *B* and *C*. At such a location 15 demands ($h_A + h_B$) are to the left of the facility and 33 demands ($h_C + h_D + h_E$) are to the right of the facility. Moving the facility δ distance units towards node *C* would reduce the objective function (the demand-weighted total distance) by 18δ units [(33-15) δ =18δ]. Again consider moving the facility from node *C* towards either node *E* or node *D*. Moving the facility a distance δ from node *C* towards node *D* on the link *CD*, would move the facility further from 36 demands and closer to only 12 demands. Thus, the objective function would increase by 24δ. Such a move would not be optimal. Similarly, moving the facility δ from *C* towards *E* on link *CE* would also increase the objective function by 20δ.

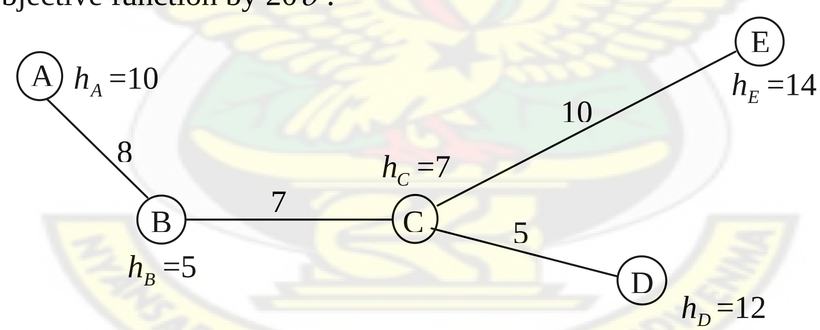


Figure 3.1: A tree network.

Using the qualitative argument outlined above, we can show that an optimal solution to the 1-median problem on a tree may be found by "folding" the demand at a tip node onto the node that is incident on the tip node and deleting the tip node (this means that we add the demand at the tip node to that of the node that is incident on the tip node and then remove the tip node and the link connecting it to the node on which it is incident from the tree). This process is repeated until a node of the new tree contains at least half of the total demand of the tree. A tip node is any node of the tree that is incident on only one other node. Thus nodes *A*, *D* and *E* are tip nodes in Figure 3.1. If the tip node *A* is folded onto node *B*, and the tip node *D* followed by node *E* folded onto node *C*, the corresponding networks, as shown in Figures 3.2 and 3.3.

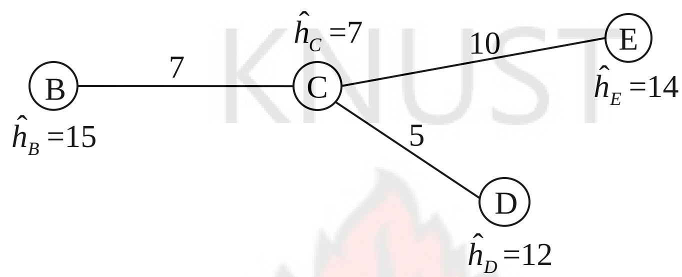


Figure 3.2: Effect of folding tip node A onto node B.

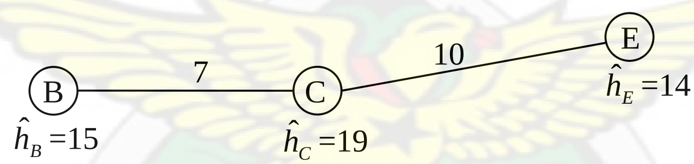


Figure 3.3: Effect of folding tip node D onto node C.

As shown in Figure 3.4, the revised demand at node *C* is over half of the total demand (H = 48). Thus, the optimal location for the 1-median on the tree is node *C*.

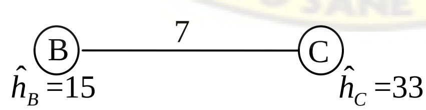


Figure 3.4: Effect of folding tip node *E* onto node *C*.

3.4 Algorithms for the 1-Median Problem on a Tree (Goldman, 1971)

Step 1: Set $\hat{h}_i = h_i$ for all nodes *i*.

Step 2: Select any tip node *i*. If $\hat{h}_i \geq \sum_j h_j / 2 = H / 2$, then locate at node *i* and go to the next step. If not, add $\hat{h}_i$ to the value of $\hat{h}_k$, where *k* is the unique node that is incident on node *i*; delete node *i* and the points on the link between *i* and *k* and repeat step 2.

Step 3: Compute the objective function.

3.5 Heuristic Algorithms for the *P*-Median Problem

The heuristic algorithms for the solution of the *p*-median problem are a myopic algorithm, an exchange algorithm and a neighborhood search algorithm. These algorithms fall into two broad classes of algorithms: construction algorithms and improvement algorithms (Golden et al, 1980).

The myopic algorithm is a construction algorithm in which we start to build good solution from scratch. Both the exchange and the neighborhood search algorithms are improvement algorithms. If we were to locate only a single facility on a network, we could find the optimal location by enumerating all possible locations and choosing the best (i.e. by total enumeration). For the fact that at least one optimal solution to any *P*-median problem consists of locating only on the demand nodes, we could evaluate the 1-median objective function, $Z_j = \sum_j h_i d_{ij}$, that would result if we locate at demand node *j*, for each demand node. We would then choose the location that results in the smallest value of Z_j. If we only want to locate a single facility, it is clear that this approach would give an optimal solution.

Now suppose that we are given the location of *p*-1 facilities. Let X_{P-1} denote the set of locations of these *p*-1 facilities. Also, let *d* (*i*, X_{P-1}) be the shortest distance between demand node *i* and the closest node in the set X_{P-1}. Similarly, we let $d(i, j \cup X_{P-1})$ be the shortest distance between demand node *i* and the closest node in the set X_{P-1} augmented by candidate location *j*. the best place to locate a single new facility, given that the first *p*-1 facilities are located at the sites given in the set X_{P-1}, is at the location *j* that minimizes $Z_j = \sum_i h_i d(i, j \cup X_{P-1})$. This approach leads to the myopic algorithm for constructing a solution to the *p*-median problem.

3.6 Myopic Algorithm for the P-Median Problem

Step 1: Initialize *k* = 0 (*k* will count the number of facilities we have located so far) and $X_k = \phi$, the empty set (X_k will give the location of *k* facilities that we have located at each stage of the algorithm).

Step 2: Increment *k*, the counter on the number of facilities located.

Step 3: Compute $Z_j^k = \sum_i h_i d(i, j \cup X_{k-1})$ for each *j* which is not in the set X_{k-1}. (Z_j^k gives the value of the *p*-median objective function if we locate the k^{th} facility at node *j*, given that the first *k*-1 facilities are at the locations given in the set X_{k-1} and node *j* is not in that set).

Step 4: Find the node $j^*(k)$ that minimizes Z_j^k, that is $j^*(k)$ = argmin$_j$ (Z_j^k). [$j^*(k)$ gives the best location for the k^{th} facility, given the location of the first *k*-1 facilities). Add node $j^*(k)$ to the set X_{k-1} to obtain the set X_k; that is, set $X_k = X_{k-1} \cup j^*(k)$.

Step 5: If *k* = *P* stop; the set X_p is the solution to the myopic algorithm. If *k* < *P* go to step 2.

The solution obtained using the myopic algorithm will not necessarily be optimal but this algorithm is appealing for a number of reasons:

- It is very simple to understand and to implement.
- In practice, many decisions are made this way. We are often given the location of some number of facilities which cannot be moved. We are then asked to find the location of a few (often only one or two) new facilities (Daskin, 1952).

If we are only required to locate one additional facility and the existing ones cannot be relocated, this approach will clearly be optimal.

To illustrate the approach, we consider the network of figure 3.5. Numbers in boxes next to the nodes are demands, h_i .

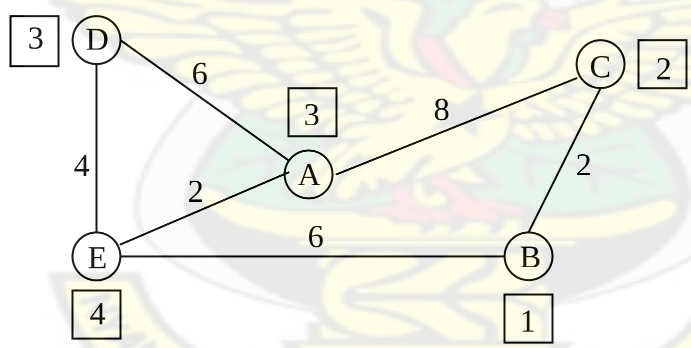


Figure 3.5: Sample network for *p*-median problem.

By using either the Floyd's algorithm or the Dijkstra's algorithm, we obtained shortest paths matrix (distance matrix) $d(i, j)$ for the above network.

Table 3.1: Distance matrix, d(i, j).

$d(i, j) =$

From \ To	A	B	C	D	E
A	0	8	8	6	2
B	8	0	2	10	6
C	8	2	0	12	8
D	6	10	12	0	4
E	2	6	8	4	0

Next we find demand times distance [$h_i .d(i, j)$]. Here, we multiply *A* row of $d(i, j)$ by h_A, *B* row of $d(i, j)$ by h_B, and so on. By summing the entries in each column, the values of Z_j^1 are obtained. The smallest value of Z_j^1 gives the solution to 1-median problem.

Table 3.2: First Myopic Median, [h_i . d(i, j)].

Node *i* \ Node *j*	A	B	C	D	E
A	0	24	24	18	6
B	8	0	2	10	6
C	16	4	0	24	16
D	18	30	36	0	12
E	8	24	32	16	0
Total	50	82	94	68	40

The smallest Z_j^1 value corresponds to $j = E$, with a value of 40. Thus, the optimal total demand weighted distance if we locate only one median for the network is 40, resulting in an average distance of 40/13 or 3.077. The total demand is 13.

To locate a second median, we compute $h_i.\min\{d(i,E);d(i,j)\}$ for each node/candidate location pair (*i, j*). The column totals correspond to Z_j^2.

Table 3.3: Second Myopic Median, [h_i . min{d(i. j) ; d(i, E)}].

Node *j*

Node *i*	A	B	C	D	E
A	0	6	6	6	6
B	6	0	2	6	6
C	16	4	0	20	16
D	12	12	12	0	12
E	0	0	0	0	0
Total	34	22	20	32	40

It is best to add a facility at *C*. The total demand-weighted distance is now 20 resulting in an average distance of 30 / 13 or 1.538.

To locate a third facility, we compute $h_i.\min\{d(i,E);d(i,C);d(i,j)\}$ for each node / candidate location pair (*i, j*). The column totals correspond to Z_j^3.

The third facility must, therefore, be located at *D*. The total demand-weighted distance is 8 resulting in an average distance of 8 / 13 or 0.615.

Table 3.4: Third Myopic Median, [h_i . min{d(i, E) ; d(i, C) ; d(i, j)}].

Node *j*

Node *i*	A	B	C	D	E
A	0	6	6	6	6
B	2	0	2	2	2
C	0	0	0	0	0
D	12	12	12	0	12
E	0	0	0	0	0
Total	14	18	20	8	20

The table 3.5 shows the results for the first three myopic medians for the network of figure 3.1 .

Table 3.5: Results for the First Three Myopic Medians

Median number	Location	Total demand-weighted distance	Average distance
1	E	40	3.077
2	C	20	1.538
3	D	8	0.615

The results for the first three median is given on the network diagram below.

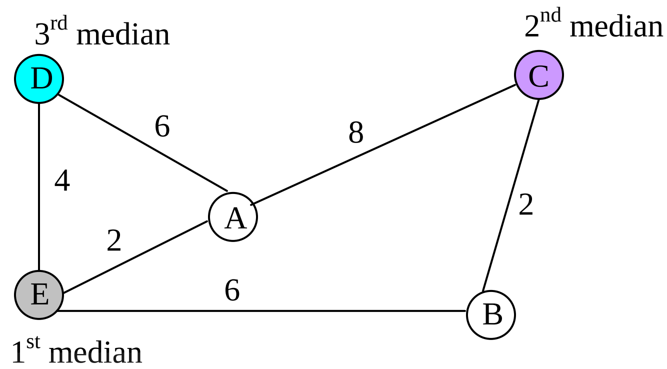


Figure 3.6: Network diagram for result.

3.7.0 Improvement Algorithm

3.7.1 Neighborhood Search Algorithm

Given the locations of some facilities (whether the locations are optimal or not), each demand node should be assigned to the nearest facility, since the facilities are uncapacitated, and the demand-weighted distance minimized. This creates sets of nodes such that all nodes in the same set are assigned to the same facility. Nodes within a set are referred to as being in the neighborhood of the facility to which they are assigned. Figure 3.7 displays the neighborhood associated with the three myopic medians of the original network.

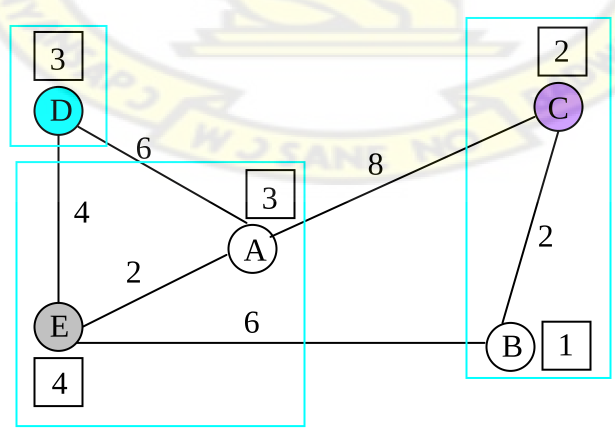


Figure 3.7: Neighborhoods associated with myopic 3-median.

Within each neighborhood, it is expected that the median would be located optimally. That is, the facility serving each neighborhood would be located at the optimal 1-median site for the nodes within the neighborhood. This leads to the neighborhood search algorithm.

The neighborhood search algorithm can begin with any set of P facilities sites. For instance, we could start with the P sites identified by the myopic algorithm. For each facility site, the algorithm identifies the set of demand nodes that constitute the neighborhood around the facility site. Within each neighborhood the optimal 1-median is found. If any site changes, the algorithm reallocates demands to the nearest facility and forms new neighborhoods. If any of the neighborhoods changes, the algorithm again finds the 1-median within each neighborhood, and so on.

Consider the Fig.3.7 above, which displays the neighborhoods associated with the 3-myopic median. If, for example, the facility at node E is moved to node A, the total demand-weighted distance would increase from 8 to 10. Since the objective function is being minimized, this movement would worsen the solution. Similarly, if the facility at node C is moved to node B, the total demand-weighted distance would increase from 8 to 10. It means that the solution obtained earlier using the myopic algorithm is optimal.

Figure 3.8 is the flowchart of neighborhood search algorithm [Maranzana (1964) was the first to propose such an algorithm].

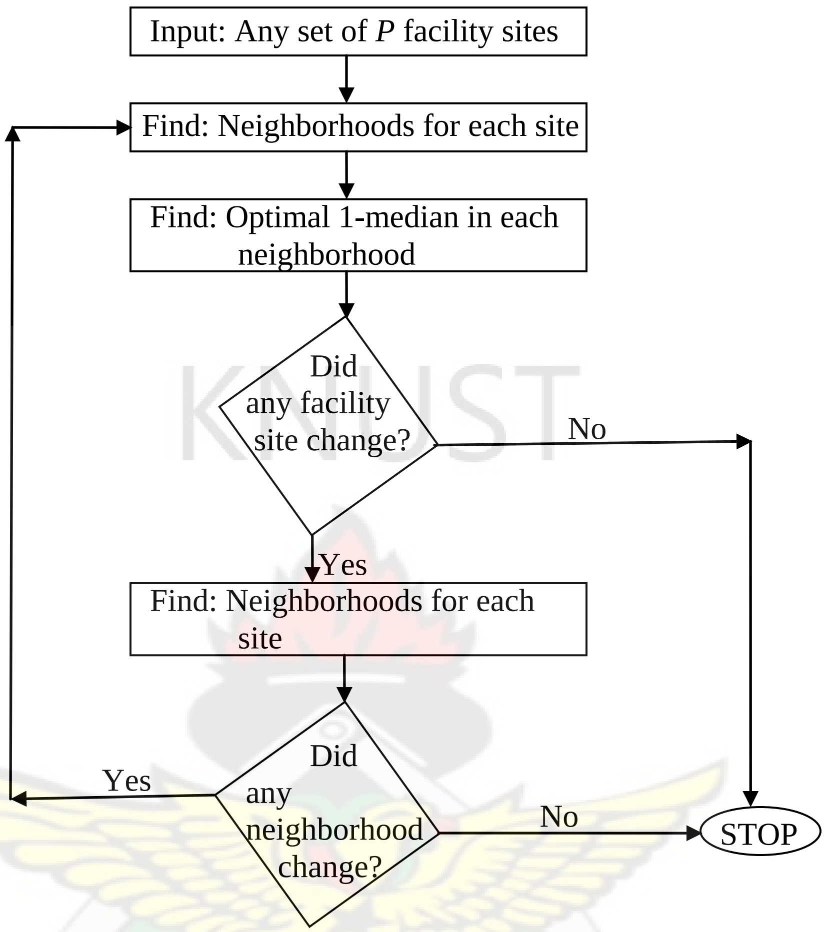


Figure 3.8: Flowchart of neighborhood search algorithm (Maranzana, 1964).

3.7.2 Exchange Algorithm

In evaluating the impact of any relocation decision using the neighborhood search algorithm, only the effects on those nodes in the neighborhood are considered. The potential benefit to nodes outside of the neighborhood is not considered in deciding whether or not relocation should be made. This is a limitation on the part of the neighborhood search algorithm, hence the need for exchange algorithm as an alternative improvement algorithm. Figure 3.9 shows the flowchart of the exchange algorithm.

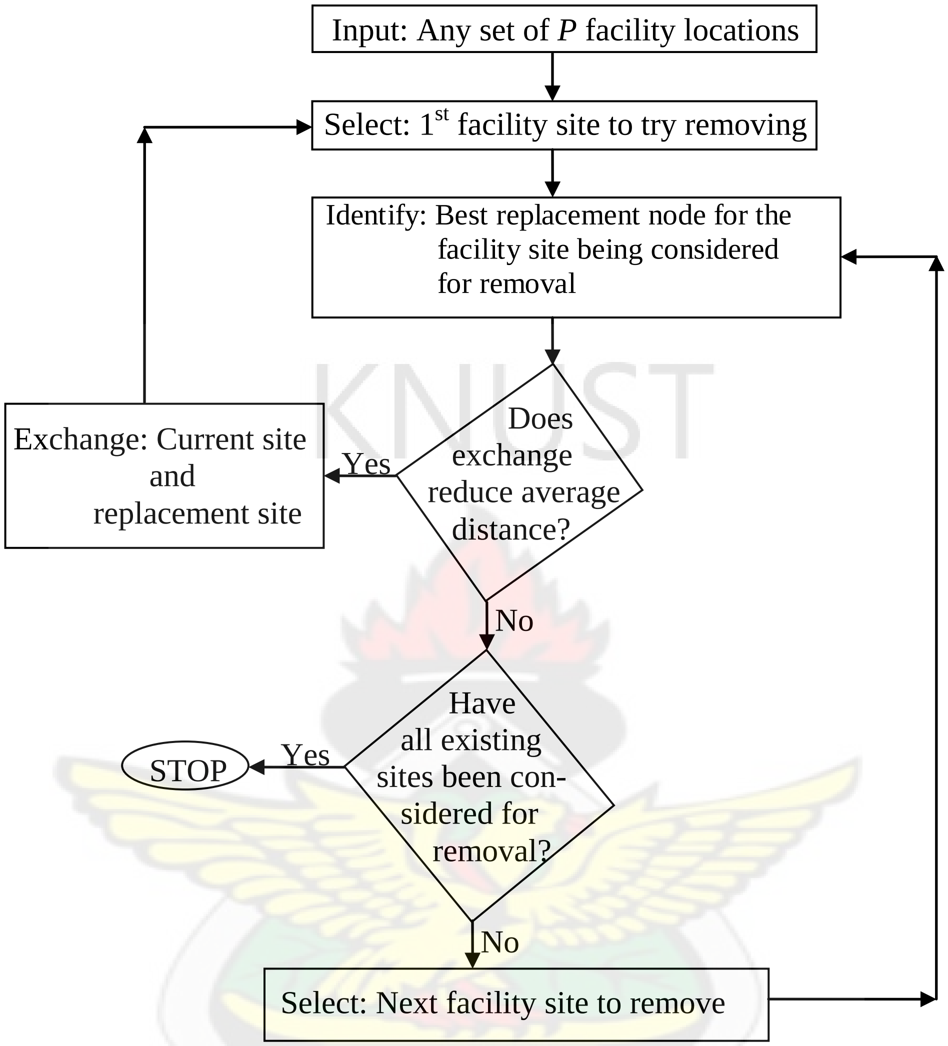


Figure 3.9: Flowchart of the exchange algorithm.

Consider the figure 3.6, if the facility at node *D* should be moved to node *A*, and the demand at node *D* allocated to the facility at node *E*, then the objective function value becomes $3(4) + 1(2) = 14 > 8$ (myopic solution). In the same way, if either of the facilities at nodes *C* or *E* should be moved to another node, the objective function value would be more than 8. Thus the solution found under the myopic algorithm is optimal.

3.8 An Optimization-Based Lagrangian Algorithm for the *P*-Median Problem

Many heuristic algorithms have been used successfully in solving applied *p*-median problems. However, the problem with these algorithms and any heuristic approaches is that we do not know how good the solution is. In some cases, the solution may be optimal or very close to optimal. In other cases, the solution may be quite far from optimal. This calls for an algorithm which can give us a clue as to whether the solution is optimal or close to optimal. Here, we consider an optimization-based Lagrangian algorithm for the *p*-median problem. Lagrangian relaxation is an approach to solving difficult problems such as integer programming problems.

The Lagrangian relaxation is base on the premise that removing constraints from a problem makes the problem easier to solve. The relaxation removes a constraint but introduces a penalty for violating the removed constraint.

The Lagrangian relaxation approach under the *p*-median problem involves the following steps:

1. Relax one or more constraint(s) by multiplying through by Lagrange multiplier(s) and bring the constraint(s) into the objective function.
2. Solve the resulting relaxed problem to find the optimal values of the original decision variables (in the relaxed problem).
3. Use the resulting decision variables from the solution to the relaxed problem found in step (2) to find a feasible solution to the original problem. Update the lower bound (LB) on the best feasible solution known for the problem.
4. Use the solution obtained in step (2) to compute a lower bound on the best value of the objective function.

5. Examine the solution obtained in step (2) and determine which of the relaxed constraints are violated. Use, for example, the subgradient optimization method to modify the Lagrange multipliers in such a way that the violated constraints are less likely to be violated on the subsequent iteration.

3.9 Termination of the Lagrangian Algorithm

The Lagrangian algorithm is terminated when one/more of the following conditions is/are true:

1. When a number of specified iterations is done.
2. The lower bound equals the upper bound (i.e. $L^n = UB$), or L^n is close enough to UB.
3. α^n becomes very small. When α^n is very small, the changes are not likely to help solve the problem (Daskin, 1952).
4. When there is no violation of the relaxed constraints (i.e., Q = $\sum_i \{\sum_j Y_{ij}^n - 1\}^2 = 0$).

   (Refer to Table 4.8 in page 64 and Table B2 in page 76).

3.10 Lagrangian Formulations

We begin by restating the formulation of the original *p*-median problem.

$$\text{Minimize} \qquad \sum_i \sum_j h_i d_{ij} Y_{ij} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots. \qquad (1)$$

$$\text{Subject to} \qquad \sum_j Y_{ij} = 1 \qquad \forall i \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots. \qquad (2)$$

$$\sum_j X_j = P \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots. \qquad (3)$$

$$Y_{ij} - X_j \leq 0 \qquad \forall i, j \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots... \qquad (4)$$

$$X_j = 0, 1 \qquad \forall j \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (5)$$

$$Y_{ij} = 0, 1 \qquad \forall i, j \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (6)$$

To solve the above *p*-median problem using Lagrangian relaxation, either constraint (2) or (4) can be relaxed. Now, if constraint (2) is relaxed, we have

$$\underset{\lambda}{MAX}\{\underset{X,Y}{MIN} \quad \sum_i \sum_j h_i d_{ij} Y_{ij} + \sum_i \lambda_i (1 - \sum_j Y_{ij})\} \text{ or}$$

$$\underset{\lambda}{MAX}\{\underset{X,Y}{MIN} \quad \sum_i \sum_j (h_i d_{ij} - \lambda_i) Y_{ij} + \sum_i \lambda_i\} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (1a)$$

$$SUBJECT\ TO: \quad \sum_j X_j = P \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (3)$$

$$Y_{ij} - X_j \leq 0 \quad \forall i, j \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (4)$$

$$X_j = 0, 1 \qquad \forall j \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (5)$$

$$Y_{ij} = 0, 1 \qquad \forall i, j \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (6)$$

For fixed values of the Lagrange multipliers, λ_i (estimated demand-weighted distance), we want to minimize the objective function. With the values of λ_i fixed, the second terms of the objective function is a constant. To minimize the objective function, we set Y_{ij} = 1 if its coefficient $h_i d_{ij} - \lambda_i < 0$, and Y_{ij} = 0 otherwise. But setting Y_{ij} = 1 means already X_j = 1 by constraint (4), and from constraint (3), we can set X_j = 1 for *p* number of *j* values. Thus, to minimize the objective function for fixed values of the Lagrange multipliers, we begin by computing the value of setting each value of the X_j values to 1. This value is given by $V_j = \sum_i \min(0, \ h_i d_{ij} - \lambda_i)$ for each candidate location *j*. We then find *p* smallest values of V_j and set the corresponding values of X_j =1 and all other values of X_j = 0. We then set

$$Y_{ij} = \begin{cases} 1 \text{ if } X_j = 1 \text{ and } h_i d_{ij} - \lambda_i < 0 \\ 0 \text{ if not} \end{cases}$$

If we relax constraint (4), we get

$$\underset{\lambda}{MAX}\{\underset{X,Y}{MIN} \quad \sum_i \sum_j h_i d_{ij} Y_{ij} + \sum_i \sum_j \lambda_{ij} \underset{i}{(Y_{ij} - X_j)}\} \text{ or}$$

$$\underset{\lambda}{MAX}\{\underset{X,Y}{MIN} \quad \sum_i \sum_j (h_i d_{ij} + \lambda_{ij}) Y_{ij} - \sum_j (\sum_i \lambda_{ij}) X_j \quad \ldots\ldots\ldots\ldots\ldots \quad (1b)$$

SUBJECT TO: $\sum_j Y_{ij} = 1 \qquad \forall i \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \qquad (2)$

$\sum_j X_j = P \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \qquad (3)$

$X_j = 0, 1 \qquad \forall j \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (5)$

$Y_{ij} \geq 0 \qquad \forall i, j \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (6)$

$\lambda_{ij} \geq 0 \qquad \forall i, j \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (7)$

Again, for fixed values of the Lagrange multipliers, λ_{ij}, we want to minimize the objective function. In this case, the problem breaks into two separate subproblems; one in the allocation variables, Y_{ij}, and the other in the location variables, X_j. Below are these subproblems:

Problem in the allocation variables Y_{ij} for fixed values of λ_{ij}

Minimize $\sum_i \sum_j (h_i d_{ij} + \lambda_{ij}) Y_{ij} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (1c)$

Subject to $\sum_j Y_{ij} = 1 \qquad \forall i \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \qquad (2)$

$Y_{ij} \geq 0 \qquad \forall i, j \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (6)$

Problem in the Location Variables X_j for fixed values of λ_{ij}

Maximize $\sum_j (\sum_i \lambda_{ij}) X_j$ ………………………… (1d)

Subject to $\sum_j X_j = P$ ………………………….. (3)

$X_j = 0, 1 \quad \forall j$ ……………………… (5)

It is worth noting that the problem in X_j for fixed values of λ_{ij} becomes a maximization problem because the objective function (1b) is being minimized.

To solve the problem in Y_{ij}, we identify the facility location $j_i^* = \text{argmin}_j\{h_i d_{ij} + \lambda_{ij}\}$ for each demand location i. That is, j_i^* is the facility location that minimizes $h_i d_{ij} + \lambda_{ij}$ for demand node i. Set $Y_{ik} = 1$ if $k = j_i^*$ and $Y_{ik} = 0$ for all other facility locations k.

To solve for the optimal values of the location variables X_j, for fixed values of λ_{ij}, we find the P largest values of $\sum_i \lambda_{ij}$. We then set the corresponding X_j values to 1 and all other X_j values to 0.

In either relaxation, we find a primal feasible solution related to the Lagrangian solution by ignoring the allocation variables, Y_{ij}, and siting the facilities at those sites for which X_j =1. We then let $S = \{ j \backslash X_j = 1\}$; that is, S is the set of facility locations. For each demand node i, we then find $\hat{j}_i = \arg\min_{j \in S}\{d_{ij}\}$; that is, $\hat{j}_i$ is the open facility that is closest to node i. We then set $\hat{Y}_{ik} = 1$ if $k = \hat{j}_i$ and $\hat{Y}_{ik} = 0$ for all other locations k as before. We evaluate the P- median objective function, $\sum_i \sum_j h_i d_{ij} \hat{Y}_{ij}$. This value is an upper bound on the solution. The best (smallest) value over all iterations of the Lagrangian relaxation procedure is used as the upper bound.

The Lagrange multipliers are revised using a standard subgradient optimization procedure. When constraint (2) is relaxed, a step size, t^n, at the n^{th} iteration of the Lagrangian procedure is computed as follows:

$$t^n = \frac{\alpha^n (UB - L^n)}{\sum_i (\sum_j Y_{ij}^n - 1)^2} \quad \text{……………………} \quad (8)$$

Where, t^n = the stepsize at the n^{th} iteration of the Lagrangian procedure

α^n = a constant on the n^{th} iteration, with α^1 generally set to 2

UB = the best (smallest) upper bound on the P-median objective function

L^n = the objective function of the Lagrangian function on the n^{th} iteration

Y_{ij}^n = the optimal value of the allocation variable, Y_{ij} on the n^{th} iteration.

The Lagrange multipliers are updated using the equation below:

$$\lambda_i^{n+1} = \max\{0, \lambda_i^n - t^n (\sum_j Y_{ij}^n - 1)\} \quad \text{……………………} \quad (9)$$

When the constraint (4) is relaxed, equations (8) and (9) are modified as follows:

$$t^n = \frac{\alpha^n (UB - L^n)}{\sum_i \sum_j (Y_{ij}^n - X_j^n)^2} \quad \text{……………………………} \quad (10)$$

$$\lambda_i^{n+1} = \max\{0, \lambda_i^n + t^n (Y_{ij}^n - X_j^n)\} \quad \text{……………………..} \quad (11)$$

Where all notations is as defined above and,

X_j^n = the optimal value of the location variable, X_j, on the n^{th} iteration.

It is worth noting that the Lagrange multipliers are initialized, and one way is to initialize them to constant value.

CHAPTER FOUR

4.1 Data Collection and Analysis

The six halls of residence of KNUST and a hostel are considered in this piece of work. Students' population in the various halls / hostel was collected from the hall / hostel officials. Table 4.1 shows the students population in the halls / hostel.

Table 4.1: Students population in the halls/hostel.

Name of Hall / Hostel	Node i	Number of Students
GUSS Hostel	A	935
University Hall	B	1190
Independence Hall	C	1176
Unity Hall	D	1925
Republic Hall	E	1208
Queens Hall	F	1164
Africa Hall	G	712
		Total = 8310

The set of distances of roads linking the halls / hostel was collected from the Geomatic Department of the Kwame Nkrumah University of Science and Technology. This has been presented in the Table 4.2 below.

Table 4.2: Inter-halls/hostel distances.

Edges (i, j)	Distance [d(i, j), in metres]
(A, B)	306
(B, C)	1050
(B, F)	950
(C, D)	340
(C, E)	210
(D, E)	380
(D, G)	400
(E, F)	100
(F, G)	375

The above data has been developed into a network of Figure 4.1 below. Numbers in boxes next to the nodes are the number of students in the halls / hostel. These numbers represent the demand (h_i).

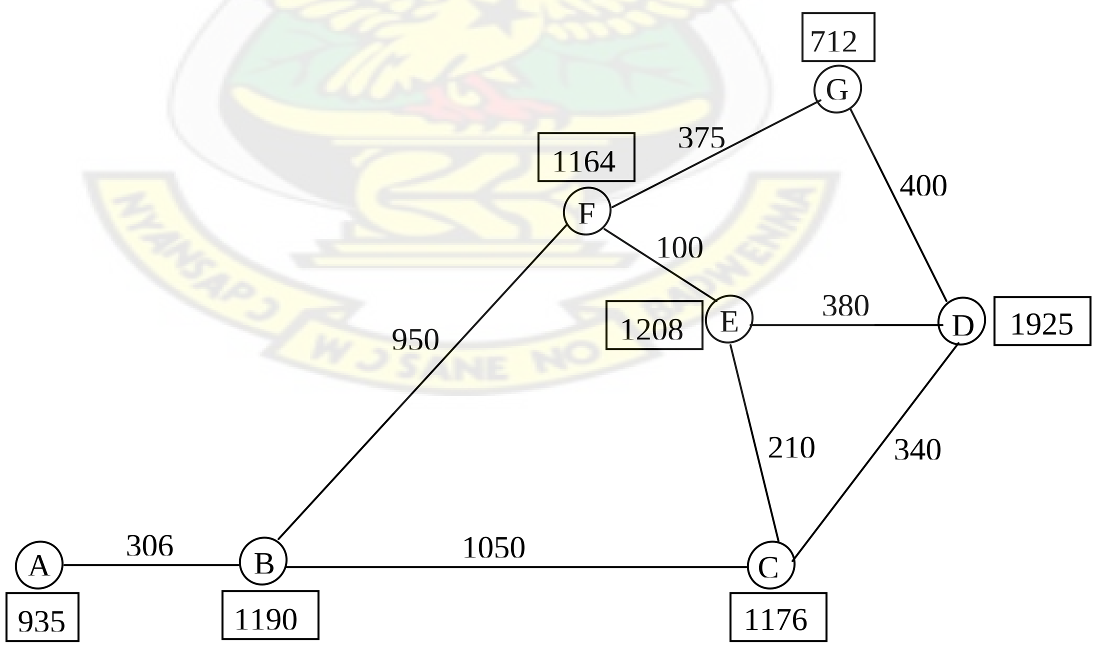


Figure 4.1: Network of the data.

By using the Dijkstra's algorithm, the shortest path matrix or distance matrix, d(i, j) for the above network was obtained as shown in Table 4.3.

Table 4.3: Shortest path distance matrix, d(i, j).

d(i, j) =

From	To: A	B	C	D	E	F	G
A	0	306	1356	1636	1356	1256	1631
B	306	0	1050	1390	1050	950	1325
C	1356	1050	0	340	210	310	685
D	1636	1390	340	0	380	480	400
E	1356	1050	210	380	0	100	475
F	1256	950	310	480	100	0	375
G	1631	1325	685	400	475	375	0

Table 4.4 shows the shortest path distance matrix together with the demands at the various nodes.

Table 4.4: Demand (h_i) and shortest path distance matrix d(i, j).

		To						
h_i	From	A	B	C	D	E	F	G
935	A	0	306	1356	1636	1356	1256	1631
1190	B	306	0	1050	1390	1050	950	1325
1176	C	1356	1050	0	340	210	310	685
1925	D	1636	1390	340	0	380	480	400
1208	E	1356	1050	210	380	0	100	475
1164	F	1256	950	310	480	100	0	375

712	G	1631	1325	685	400	475	375	0

4.2 Solution Using the Myopic Algorithm

We begin by using the myopic algorithm to find the first two myopic medians. Here, we multiply the demand at node *i* by the distance between node *i* and node *j*, ($h_i . d_{ij}$). That is, we multiply the row A through by the demand at node A, h_A (i.e. h_A . d(A, j)), the row B by the demand at node B, h_B (i.e. h_B . d(B, j)), and so on. The result is shown in Table 4.5. Sums of entries in each column are found, and the smallest total gives the solution to 1- median problem. This corresponds to the node where the first facility would be located. From the Table 4.5 below, the first facility would be located at node E.

Table 4.5: First Myopic Median, [h_i . d(i, j)]

	Node j						
Node i	A	B	C	D	E	F	G
A	0	286110	1267860	1529660	1267860	1174360	1524985
B	364140	0	1249500	1654100	1249500	1130500	1576750

C	1594656	1234800	0	399840	246960	364560	805560
D	3149300	2675750	654500	0	731500	924000	770000
E	1638048	1268400	253680	459040	0	120800	573800
F	1461984	1105800	360840	558720	116400	0	436500
G	1161272	943400	487720	284800	338200	267000	0
Total	9369400	7514260	4274100	4886160	3950420	3981220	5687595

For the second median, we compute [h_i . min{d(i, j) ; d(i, E)}]. The column with the least sum gives the solution to the 2 – median problem. It means that the second facility would be located at node B, as shown in the Table 4.6 below.

Table 4.6: Second Myopic Median, [h_i . min{d(i, j) : d(i, E)}]

	Node j						
Node i	A	B	C	D	E	F	G
A	0	286110	1267860	1267860	1267860	1174360	1267860
B	364140	0	1249500	1249500	1249500	1130500	1249500
C	246960	246960	0	246960	246960	246960	246960
D	731500	731500	654500	0	731500	731500	731500
E	0	0	0	0	0	0	0
F	116400	116400	116400	116400	116400	0	116400
G	338200	338200	338200	284800	338200	267000	0
Total	1797200	1719170	3626460	3165520	3950420	3550320	3612220

From the two myopic median tables above, the facilities should be located at nodes B and E representing the University hall and the Republic hall respectively.

4.3.0 The Lagrangian Algorithm

At this point, we use the Lagrangian algorithm to solve the problem. We begin by formulating the p-median problem as:

Minimize

$$\begin{aligned}
&0Y_{AA} + 286110Y_{AB} + 1267860Y_{AC} + 1529660Y_{AD} + 1267860Y_{AE} + 1174360Y_{AF} + 1524985Y_{AG} \\
&+ 364140Y_{BA} + 0Y_{BB} + 1249500Y_{BC} + 1654100Y_{BD} + 1249500Y_{BE} + 1130500Y_{BF} + 1576750Y_{BG} \\
&+ 1594656Y_{CA} + 1234800Y_{CB} + 0Y_{CC} + 399840Y_{CD} + 246960Y_{CE} + 364560Y_{CF} + 805560Y_{CG} \\
&+ 3149300Y_{DA} + 2675750Y_{DB} + 654500Y_{DC} + 0Y_{DD} + 731500Y_{DE} + 924000Y_{DF} + 770000Y_{DG} \\
&+ 1638048Y_{EA} + 1268400Y_{EB} + 253680Y_{EC} + 459040Y_{ED} + 0Y_{EE} + 120800Y_{EF} + 573800Y_{EG} \\
&+ 1461984Y_{FA} + 1105800Y_{FB} + 360840Y_{FC} + 558720Y_{FD} + 116400Y_{FE} + 0Y_{FF} + 436500Y_{FG} \\
&+ 1161272Y_{GA} + 943400Y_{GB} + 487720Y_{GC} + 284800Y_{GD} + 338200Y_{GE} + 267000Y_{GF} + 0Y_{GG}
\end{aligned} \tag{4.1a}$$

Subject to:

$$Y_{AA} + Y_{AB} + Y_{AC} + Y_{AD} + Y_{AE} + Y_{AF} + Y_{AG} = 1$$

$$Y_{BA} + Y_{BB} + Y_{BC} + Y_{BD} + Y_{BE} + Y_{BF} + Y_{BG} = 1$$

$$Y_{CA} + Y_{CB} + Y_{CC} + Y_{CD} + Y_{CE} + Y_{CF} + Y_{CG} = 1$$

$$Y_{DA} + Y_{DB} + Y_{DC} + Y_{DD} + Y_{DE} + Y_{DF} + Y_{DG} = 1$$

$$Y_{EA} + Y_{EB} + Y_{EC} + Y_{ED} + Y_{EE} + Y_{EF} + Y_{EG} = 1$$

$$Y_{FA} + Y_{FB} + Y_{FC} + Y_{FD} + Y_{FE} + Y_{FF} + Y_{FG} = 1$$

$$Y_{GA} + Y_{GB} + Y_{GC} + Y_{GD} + Y_{GE} + Y_{GF} + Y_{GG} = 1 \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (4.2)$$

$$X_A + X_B + X_C + X_D + X_E + X_F + X_G = 2 \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (4.3)$$

$$Y_{AA},\ Y_{AB},\ Y_{AC},\ Y_{AD},\ Y_{AE},\ Y_{AF},\ Y_{AG} \leq X_A$$

$$Y_{BA},\ Y_{BB},\ Y_{BC},\ Y_{BD},\ Y_{BE},\ Y_{BF},\ Y_{BG} \leq X_B$$

$$Y_{CA},\ Y_{CB},\ Y_{CC},\ Y_{CD},\ Y_{CE},\ Y_{CF},\ Y_{CG} \leq X_C$$

$$Y_{DA},\ Y_{DB},\ Y_{DC},\ Y_{DD},\ Y_{DE},\ Y_{DF},\ Y_{DG} \leq X_D$$

$$Y_{EA},\ Y_{EB},\ Y_{EC},\ Y_{ED},\ Y_{EE},\ Y_{EF},\ Y_{EG} \leq X_E$$

$$Y_{FA},\ Y_{FB},\ Y_{FC},\ Y_{FD},\ Y_{FE},\ Y_{FF},\ Y_{FG} \leq X_F$$

$$Y_{GA},\ Y_{GB},\ Y_{GC},\ Y_{GD},\ Y_{GE},\ Y_{GF},\ Y_{GG} \leq X_G \quad \ldots\ldots\ldots\ldots\ldots \quad (4.4)$$

$$X_A,\ X_B,\ X_C,\ X_D,\ X_E,\ X_F,\ X_G \in \{0, 1\} \quad \ldots\ldots\ldots\ldots\ldots \quad (4.5)$$

$$Y_{AA}, Y_{AB}, Y_{AC}, Y_{AD}, Y_{AE}, Y_{AF}, Y_{AG}, Y_{BA}, Y_{BB}, Y_{BC}, Y_{BD}, Y_{BE}, Y_{BF}, Y_{BG},$$
$$Y_{CA}, Y_{CB}, Y_{CC}, Y_{CD}, Y_{CE}, Y_{CF}, Y_{CG}, Y_{DA}, Y_{DB}, Y_{DC}, Y_{DD}, Y_{DE}, Y_{DF}, Y_{DG},$$
$$Y_{EA}, Y_{EB}, Y_{EC}, Y_{ED}, Y_{EE}, Y_{EF}, Y_{EG}, Y_{FA}, Y_{FB}, Y_{FC}, Y_{FD}, Y_{FE}, Y_{FF}, Y_{FG},$$
$$Y_{GA}, Y_{GB}, Y_{GC}, Y_{GD}, Y_{GE}, Y_{GF}, Y_{GG} = \{0, 1\} \quad \ldots\ldots\ldots\ldots\ldots \quad (4.6)$$

We want to relax the constraint (4.2). This process is in two steps; we first multiply the constraints through by the Lagrange multipliers, λ_i, and then bring them into the objective function. The end result, as shown below, is the Lagrangian objective function.

$$\underset{\lambda}{MAX}\ \underset{X,Y}{MIN}$$

$$(0 - \lambda_A)Y_{AA} + (286110 - \lambda_A)Y_{AB} + (1267860 - \lambda_A)Y_{AC} + (1529660 - \lambda_A)Y_{AD} + (1267860 - \lambda_A)Y_{AE} + (1174360 - \lambda_A)Y_{AF} + (1524985 - \lambda_A)Y_{AG}$$

$$+ (364140 - \lambda_B)Y_{BA} + (0 - \lambda_B)Y_{BB} + (1249500 - \lambda_B)Y_{BC} + (1654100 - \lambda_B)Y_{BD} + (1249500 - \lambda_B)Y_{BE} + (1130500 - \lambda_B)Y_{BF} + (1576750 - \lambda_B)Y_{BG}$$

$+ (1594656 - \lambda_C)Y_{CA} + (1234800 - \lambda_C)Y_{CB} + (0 - \lambda_C)Y_{CC} + (399840 - \lambda_C)Y_{CD} + (246960 - \lambda_C)Y_{CE} + (364560 - \lambda_C)Y_{CF} + (805560 - \lambda_C)Y_{CG}$

$+ (3149300 - \lambda_D)Y_{DA} + (2675750 - \lambda_D)Y_{DB} + (654500 - \lambda_D)Y_{DC} + (0 - \lambda_D)Y_{DD} + (731500 - \lambda_D)Y_{DE} + (924000 - \lambda_D)Y_{DF} + (770000 - \lambda_D)Y_{DG}$

$+ (1638048 - \lambda_E)Y_{EA} + (1268400 - \lambda_E)Y_{EB} + (253680 - \lambda_E)Y_{EC} + (459040 - \lambda_E)Y_{ED} + (0 - \lambda_E)Y_{EE} + (120800 - \lambda_E)Y_{EF} + (573800 - \lambda_E)Y_{EG}$

$+ (1461984 - \lambda_F)Y_{FA} + (1105800 - \lambda_F)Y_{FB} + (360840 - \lambda_F)Y_{FC} + (558720 - \lambda_F)Y_{FD} + (116400 - \lambda_F)Y_{FE} + (0 - \lambda_F)Y_{FF} + (436500 - \lambda_F)Y_{FG}$

$+ (1161272 - \lambda_G)Y_{GA} + (943400 - \lambda_G)Y_{GB} + (487720 - \lambda_G)Y_{GC} + (284800 - \lambda_G)Y_{GD} + (338200 - \lambda_G)Y_{GE} + (267000 - \lambda_G)Y_{GF} + (0 - \lambda_G)Y_{GG}$

$$+ \lambda_A + \lambda_B + \lambda_C + \lambda_D + \lambda_E + \lambda_F + \lambda_G \qquad (4.1b)$$

SUBJECT TO:

Constraints (4.3), (4.4), (4.5) and (4.6).

4.3.1 Algorithm

Steps:

1. Use the myopic algorithm to determine the upper bounds (UB).
2. Input λ_i, $\alpha = 2$, h_i, d_{ij} and UB for $i,j = A, B, C, D, E, F, G$.
3. For each j, compute $U_{ij} = \begin{cases} h_i d_{ij} - \lambda_i, & \text{if} \quad h_i d_{ij} < \lambda_i \\ 0, & \text{if} \quad h_i d_{ij} > \lambda_i \end{cases}$
4. Calculate $V_j = \sum_i U_{ij}$.
5. Pick the two least values of V_j.
6. For such j values, assign $X_{j_1} = 1$, $X_{j_2} = 1$ and $Y_{ij} = 1$ for $U_{ij} < 0$.
7. Calculate sum of square violation, Q = $\sum_i \{\sum_j Y_{ij}^n - 1\}^2$.
8. Calculate $L^n = \sum_i \sum_j (h_i d_{ij} - \lambda_i^n) Y_{ij} + \sum_i \lambda_i^n$
9. If the sum of square violation, Q = 0 then stop.

10. Otherwise test, if $L^n - L^{n-1} \leq 0$ then use $\alpha^n = \frac{1}{2}\alpha^{n-1}$, if not use $\alpha^n = \alpha^{n-1}$.

11. Calculate $t^n = \dfrac{\alpha^n (UB - L^n)}{\sum_i (\sum_j Y_{ij}^n - 1)^2}$

12. Compute $\lambda_i^{n+1} = \max\{0, \lambda_i^n - t^n (\sum_j Y_{ij}^n - 1)\}$

13. Return to step 2.

4.4 Lagrangian Solution

Each of the iterations is in four steps.

First Iteration

Step 1:

The first step is to compute $V_j = \sum_i U_{ij}$. But $U_{ij} = min\ \{0, h_i d_{ij} - \lambda_i\}$.

That is, $U_{ij} = \begin{cases} h_i d_{ij} - \lambda_i, & if \quad h_i d_{ij} < \lambda_i \\ 0, \ if \ h_i d_{ij} > \lambda_i \end{cases}$

We also let λ_i = 500 000, for i = *A, B, C, D, E, F* and *G*.

The computation was done in table 4.7 below. The column totals gave the V_j values, for j = *A, B,* C, *D, E, F* and *G*.

Table 4.7: Values of $V_j = \sum_i U_{ij}$.

	U_A	U_B	U_C	U_D	U_E	U_F	U_G
A	-500000	-213890	0	0	0	0	0

B	-135860	-500000	0	0	0	0	0
C	0	0	-500000	-100160	- 253040	- 135440	0
D	0	0	0	-500000	0	0	0
E	0	0	-246320	- 40960	- 500000	- 379200	0
F	0	0	-139160	0	- 383600	- 500000	- 63500
G	0	0	- 12280	-215200	- 161800	- 233000	-500000
V_j	-635860	-713890	-897760	-856320	-1298440*	-1247640*	-563500

If the demand at node *i* is allocated to a facility at node *j*, then $Y_{ij} = 1$. The V_j values above suggest that, if a facility is located at node *A*, then $Y_{AA} = Y_{BA} = 1$. This means that, demands at nodes *A* and *B* would be allocated to the facility at node *A*, if the facility is at *A*. Similarly, if the facility is at *B*, then $Y_{BA} = Y_{BB} = 1$. Also, if the facility is at *C*, then $Y_{CC} = Y_{EC} = Y_{FC} = Y_{GC} = 1$. Then again, if the facility is at *D*, then we have, $Y_{CD} = Y_{DD} = Y_{ED} = Y_{GD} = 1$. If the facility is at *E*, then $Y_{CE} = Y_{EE} = Y_{FE} = Y_{GE} = 1$. Also if the facility is at F, then $Y_{CF} = Y_{EF} = Y_{FF} = Y_{GF} = 1$. Finally, if the facility is sited at *G*, then $Y_{FG} = Y_{GG} = 1$.

Since we want to locate only $p = 2$ facilities, we choose nodes *E* and *F* (the two nodes with smallest V_j values). Thus we set $X_E = X_F = 1$, and the rest $X_j = 0$. It also means that $Y_{CE} = Y_{CF} = Y_{EE} = Y_{EF} = Y_{FE} = Y_{FF} = Y_{GE} = Y_{GF} = 1$, and the rest of $Y_{ij} = 0$.

Step 2:

In this step we find the first Lagrangian objective function value, L^1.

$$L^1 = \sum_i \sum_j (h_i d_{ij} - \lambda_i) Y_{ij} + \sum_i \lambda_i$$

$$= 7(500000) + (-\ 1298440 - 1247640) = 3500000 - 2546080 = 953920$$

Next, we find the sum of the squared violation of constraint (3.2), which is $\sum_j Y_{ij} - 1 = 0$.

$$\sum_i \{\sum_j Y_{ij}^1 - 1\}^2 = (0 - 1)^2 + (0 - 1)^2 + (2 - 1)^2 + (0 - 1)^2 + (2 - 1)^2 + (2 - 1)^2 + (2 - 1)^2$$

$$= 1 + 1 + 1 + 1 + 1 + 1 + 1 = 7.$$

Step 3:

Here we find the stepsize, t^n, by using the formula $t^n = \frac{\alpha^n (UB - L^n)}{\sum_i (\sum_j Y_{ij}^n - 1)^2}$ (Daskin, 1952).

UB = 1719170m (i.e. myopic optimal value), $L^n = L^1$ = 953920 (i.e. first Lagrangian optimal value), $\sum_i \{\sum_j Y_{ij}^1 - 1\}^2 = 7$ (sum of square violation of constraint 4.2), $\alpha^n = \alpha^1 = 2$, (Daskin, 1952). It must be noted that, α^n is halved if $L^{i+1} - L^i \leq 0$.

Therefore we have,

$$t^1 = \frac{\alpha^1 (UB - L^1)}{\sum_i (\sum_j Y_{ij}^1 - 1)^2} = \frac{2(1719170 - 953920)}{7} = 218642.86 \cong 218643.$$

Step 4:

The last step for the first iteration is to update the Lagrange multipliers, λ_i, by using the formula of Daskin, (1952); λ_i^{n+1} = max $\{0,\ \lambda_i^n - t^n (\sum_j Y_{ij}^n - 1)\}$. And we have,

λ_i^2 = max $\{0,\ \lambda_i^1 - t^n (\sum_j Y_{ij}^1 - 1)\}$

λ_A^2 = max {0, 500000 – 218643 (- 1)} = 718643

λ_B^2 = max {0, 500000 – 218643 (- 1)} = 718643

λ_C^2 = max {0, 500000 – 218643 (1)} = 281357

λ_D^2 = max {0, 500000 – 218643 (- 1)} = 718643

λ_E^2 = max {0, 500000 – 218643 (1)} = 281357

λ_F^2 = max {0, 500000 – 218643 (1)} = 281357

λ_G^2 = max {0, 500000 – 218643 (1)} = 281357

The results obtained from the various iterations of the Lagrangian algorithm are summarized in the Table 4.8 below.

Let Q = $\sum_i \{\sum_j Y_{ij}^n - 1\}^2$ (i.e. sum of square violation of constraints 4.2).

The V_j values with asterisks in each column, are the two minimum values chosen.

Table 4.8: Computational Results of the Various Iterations.

Variable	1st	2nd	3rd	4th	5th
V_A	-635860	-1073146*	-69178	-963414*	-881266
V_B	-713890	-1151176*	-772814	-1041444*	-959296*
V_C	-897760	-373177	-1050418	-647473	-811769
V_D	-856320	-718643	-1175758*	-824732	-907080
V_E	-1298440*	-480711	-1356916*	-821633	-1025026*
V_F	-1247640*	-4456271	-1129792	-755184	-891143
V_G	-563500	-281357	-642400	-373770	-496692
L^n	953920	1057035	1316226	1575412	1719170
Q	7	7	6	7	0
t^n	218643	189181	134315	41074	_
α^n	2	2	2	2	_

λ_A^{n+1}	718643	529462	663777	622703	_
λ_B^{n+1}	718643	529462	663777	622703	_
λ_C^{n+1}	281357	470538	336223	377297	_
λ_D^{n+1}	718643	907824	773509	814583	_
λ_E^{n+1}	281357	470538	336223	377297	_
λ_F^{n+1}	281357	470538	470538	511612	_
λ_G^{n+1}	281357	470538	336223	377297	_

4.5 Discussion

The total student population of the various halls / hostel is the demand allocated to the two facilities. The overall total demand is 8310 as given in Table 4.1. From the first myopic median, as it could be seen in Table 4.5, the column with the least sum corresponds to node *E*, with a value of 3950420 metres. Thus, the optimal total demand – weighted distance if only one facility were to be located is 3950420 metres, resulting in an average distance of 3950420 m / 8310 $\cong$ 475 metres. This result suggests that, if only one facility were to be located, then it should be located at node *E*. The average distance that each student would travel from any of the halls / hostel to the facility at node *E* is approximately equals to 475 metres.

For the second median, the facility is to be located at node *B*, as it could be seen in Table 4.6. The total demand – weighted distance is 1719170 metres, resulting in an average distance of 1719170 m / 8310 $\cong$ 207 metres. This result also means that, if the two facilities are located at nodes B and E, then the average distance that each student would travel from any of the halls / hostel to the nearby facility is approximately equals to 207 metres.

The result obtained from the myopic algorithm, therefore, suggested that the two facilities must be located at nodes B and E, representing the University Hall and the Republic Hall respectively.

Improvement algorithms (i.e. the neighborhood search and exchange algorithms) were also used. They all confirmed the result under the myopic algorithm.

It must be noted here that, the myopic algorithm also served as the stepping stone algorithm for the Lagrangian algorithm. It provided the upper bound (UB) value for the Lagrangian algorithm (UB = 1719170 metres).

To begin the iterations of the Lagrangian algorithm, two important choices were made;

1. The initial values of Lagrange multipliers, λ_i, ($i = A, B, C, D, E, F, G$) were chosen to be 500000, (i.e. $\lambda_A = \lambda_B = \ldots = \lambda_G = 500000$).
2. The constant, $\alpha^1 = 2$ (Daskin, 1952).

In Table 4.8, there has been a tremendous incremental jump of the values of L^n from iteration to iteration. For example, $L^2 - L^1 = 103115$, $L^3 - L^2 = 259191$, and so on. As a result, the values of α^n have been kept constant throughout the iterations. As it could also be seen in Table 4.8, the relaxed constraints (4.2) have been violated in the first four iterations. This means that, some of the demand nodes were not assigned to any facility, while others were assigned to both facilities. In the fifth and the last iteration, the violation was zero (0). This also means that, all the demand nodes were assigned to exactly one facility. The Lagrangian objective function value for the fifth iteration, L^5 = 1719170 metres. This value is also equal to the UB (i.e. myopic optimal value). The Lagrangian algorithm has, therefore, confirmed the result to be 1719170 metres. The optimal solution is therefore, X_B and X_E. Thus, the two facilities should be located at the Republic Hall (node *E*) and the University Hall (node *B*).

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The main objective of the study was to use the *p*-median model, ($p = 2$) to determine suitable locations at the KNUST Campus to establish a chain of two computer services. For the above objective to be realized, the sites must be located such that, the average distance travel by a student from a hall/hostel to the nearer of the two facilities be minimized (i.e. the average time taken is minimized).

Different methods were used to locate the suitable sites, but the main one was the Lagrangian algorithm. The result obtained using the Lagrangian suggested that, the two facilities be located at the Republic Hall (node *E*) and the University Hall (node *B*). The maximum of the lower bounds obtained was 1719170 metres. This value gave the demand-weighted distance. It resulted in the average distance of 207 metres. (i.e. the average distance = the demand-weighted distance divided by the total demand; 1719170 m / 8310 $\cong$ 207 m). It implies that, on average, each student would travel a distance of 207 metres from a hall / hostel to the nearby facility.

The myopic algorithm, which served as the stepping stone algorithm, also gave the same result. The value, 1719170 metres was the minimum of the column sums of the second myopic median. This was the value for the objective function, $\sum h_i d_{ij} Y_{ij}$. Thus, the myopic algorithm is a good approximation of the Lagrangian algorithm. Also, the neighborhood search and the exchange algorithms confirmed the result as found in the appendix.

The two facilities must, therefore, be located at the Republic Hall and the University Hall.

5.2 Recommendation

In view of the result obtained in this study, the following recommendations are made:

1 Corporate bodies as well as the individuals, who want to invest in combined computer services of typing, printing and/or repairs at KNUST Campus, are advised to establish them at the Republic Hall and the University Hall.

2 Facilities located at these halls would save the students much time and energy in traveling to them. And hence, patronage of the facilities at these halls would be higher, and as a result, investors would have value for their investment.

REFERENCES

Amponsah S. K. (2003), PhD Thesis, the University of Birmingham, U K.

Amponsah S. K. (2006) Lecture Note on Location Problems.

Aidoo E. (2007) M Sc. Thesis, the Kwame Nkrumah University of Science and Technology.

Christofides N. and Beasley J. E. (1982) A Tree Search Algorithm for p-Median Problem, *European Journal of Operation Research*, 10, 196-204.

Church R. L. and Garfinkel R. S. (1978) Locating an Obnoxious Facility on a Network, Transportation, 12, 107-118.

Cornuejols G., Nemhauser G. L and. Wolsey L. A (1990) The Uncapacitated Facility Location Problem, in Discrete Location Theory (P. B. Mirchandani and R. L. Francis, eds), Wiley, New York, Chapter 3, pages 119-171.

Daskin M. S. (1952) Network and Discrete Location, 200-229.

Daskin M. S. and Jones P. C. (1993) A New Approach to Solving Applied Location/Allocation Problem, Microcomputers in Civil Engineering, 8, 409-421.

Erkut E. and Neuman S. (1989) Analytical Models for Locating Undesirable Facilities, *European Journal of Operational Research,* 275-291.

Francis R. L., McGinnis L.F. and White J. A. (1992) Facility Location and Layout: An Analytical Approach, Prentice-Hall, *Eaglewood Cliffs, NJ.*

Ghosh A. and Harche F. (1993) Location-Allocation Models in the Private Sector: Progress, Problems and Prospects, Location Science, 1, 81-106.

Glover F., Klingman D. and Phillips N. V. (1992) Network Model in Optimization and their Applications in Practice, Wiley, New York.

Goldberg J. and Paz L. (1991) Locating Emergency Vehicle Bases when Service Time Depends on Call Location, Transportation Science, 25, 264-280.

Golden B., Bodin L., Doyle T. and Stewart W. Jr (1980) Approximate Traveling Salesman Algorithms, *Operations Research,* 13, 694-711.

Goldman A. J. (1971) Optimal Center Location in Simple Networks, *Transportation Science,* 5, 212-221.

Goldman A. J. (2006) Optimal Facility-Location, *Journal of the National Institute of Standards and Technology,* 111, 97-101.

Hakimi S. L. (1965) Optimum Distribution of Switching Centers in a Communication Network and some Related Graph Theoretic Problems, *Operational Research,* 13, 462-475.

Kariv O. and Hakimi S. L. (1979b) An Algorithmic Approach to Network Location Problems, 11; The p-Median, *SIAM Journal on Applied Mathematics,* 37, 539-560.

Krarup J. and Pruzan P. M. (1990) Ingredients of Locational Analysis, in Discrete Location Theory (Mirchandani P. B. and Francis R. L. eds.), Wiley, New York, Chapter 1, pages 1-54.

Love R. F., Morris J. G. and G. O. Wesolowsky J. G. (1988) Facility Location; Models and Methods, North-Holland, New York.

Maranzana F. E. (1964) On the Location of Supply Points to Minimize Transport Costs, *Operational Research Quarterly*, 15, 261-276.

Mirchandani P. B. (1990) The p-Median Problem and Generalizations, in Discrete Location Theory (Mirchandani P. B. and Francis R. L, eds.), Wiley, New York, Chapter 2, pages 55-117.

Mirchandani P. B. and Francis R. L. (1990) Discrete Location Theory, Wiley, New York.

ReVelle C., Mark S. D. and Liebman J. C. (1970) An Analysis of Private and Public Sector Location Models; Management Science, 16, 692-707.

Weaver J. R. and Church R. L. (1983b) A Median Facility Location Problem with Non-closest Facility Service, Working Paper 83-101, Department of Management Science and Statistics, College of Commence and Business Administration, University of Alabama, University, AL.

http://www.hyuan.com/java/how.html

http://www./ac.inpe.br/~lorena/pmed.PDF

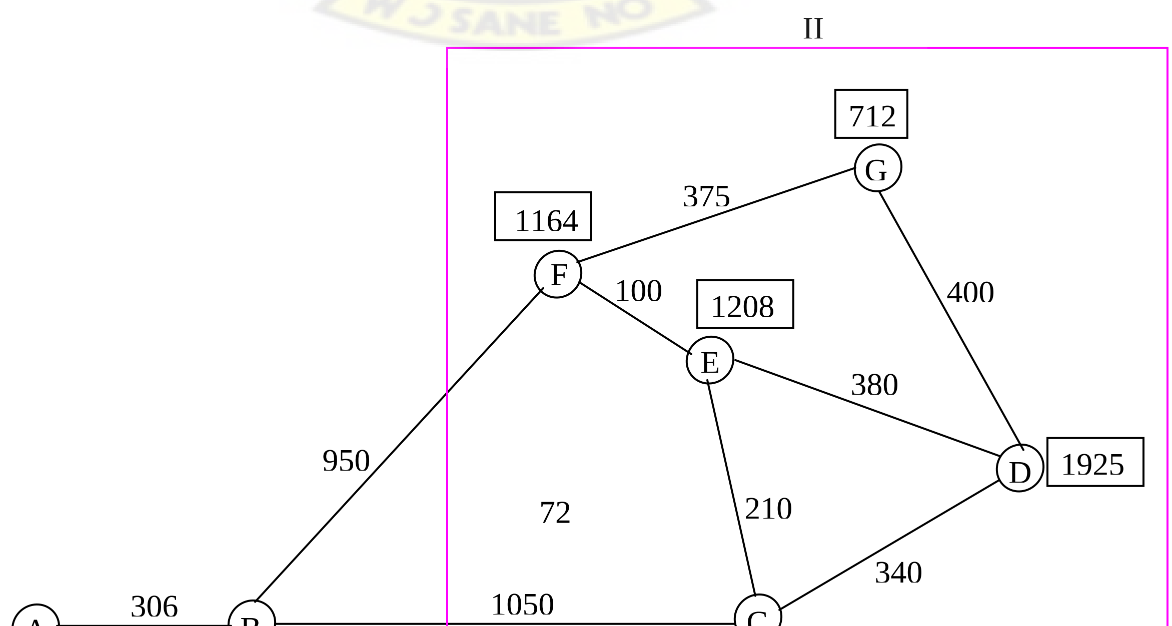
http://www.hyuan.com/java/

Appendix A

The objective function value under the myopic algorithm is 1719170m.

1. The Neighborhood Search Solution

Within neighborhood I, it is obvious that the facility would be located at node B. However, within neighborhood II, the first myopic median needs to be calculated.



I

1176

Figure A1: Network showing neighborhoods associated with myopic 2-median.

Table 1 below, shows the distance-matrix of neighborhood II, and Table 2 also shows the first myopic median of the neighborhood.

Table A1: Distance Matrix of Nodes in Neighborhood II.

From	C	D	E	F	G
C	0	340	210	310	685
D	340	0	380	480	400
E	210	380	0	100	475
F	310	480	100	0	375
G	685	400	475	375	0

Table A2: First Myopic Median of Neighborhood II.

Node i	C	D	E	F	G
C	0	399840	246960	364560	805560
D	654500	0	731500	924000	770000
E	253680	459040	0	120800	573800
F	360840	558720	116400	0	436500

G	487720	284800	338200	267000	0
Total	1756740	1702400	1433060	1676360	2585860

From the table, the facility would be located at node E. This shows that the neighborhood search algorithm confirmed the previous result. Thus, the facilities should be located at nodes B and E.

2. Exchange Solution

Now, if the facility at node E is moved to node C and the facility at node B is maintained, then, for the objective function value we have

935(306) + 1208(210) + 1164(310) + 1925(340) + 712(685)

= 286110 + 253680 + 360840 + 654500 + 487720 = 2042850 > 1719170.

Also, if the facility at node E is moved to node D and the facility at node B is maintained, then we have

935(306) + 1176(340) + 1208(380) + 1164(480) + 712(400)

= 286110 + 399840 + 459040 + 558720 + 284800 = 1988510 > 1719170.

Similarly, if the facility at node E is moved to node F and the facility at node B is maintained, then we have

935(306) + 1208(100) + 1176(310) + 1925(480) + 712(375)

= 286110 + 120800 + 364560 + 924000 + 267000 =1962470 > 1719170.

At a glance, it could be seen that, if the facility is moved from node E to node G, the objective function value would be greater than 1719170. Similarly, if the facility at node E is maintained and that at node B is move to any other node, the objective function value would still be greater than 1719170. Thus, the solution is optimal if the facilities are located at nodes B and E. These represent the University hall and Republic hall respectively.

Appendix B

Upper bound (UB) less than or greater than the one used from the myopic algorithm (1719170).

Table B1: Using the UB = 1519170.

Variable	1st	2nd	3rd	4th
V_A	-435860	-701712*	-511112	-537688
V_B	-513890	-779742*	-589142	-615718
V_C	-585480	-280468	-472602	-472602
V_D	-515360	-532926	-705800*	-692512*
V_E	-898440*	-437862	-747936*	-734648*
V_F	-847640*	-413422	-699322	-686034
V_G	-400000	-267074	-362374	-349086
L^n	1053928	1185620	1499238	1539102
Q	7	7	3	_

t^n	132926	95300	13288	_
α^n	2	2	2	_
λ_A^{n+1}	532926	437626	450914	_
λ_B^{n+1}	532926	437626	450914	_
λ_C^{n+1}	267074	362374	362374	_
λ_D^{n+1}	532926	628226	628226	_
λ_E^{n+1}	267074	362374	362374	_
λ_F^{n+1}	267074	362374	362374	_
λ_G^{n+1}	267074	362374	349086	_

Table B2: Using the UB = 1819170.

Variable	1st	2nd	3rd	4th	5th	6th	7th	8th	9th
V_A	-635860	-1130288*	-883074*	-801088	-859650*	-817810	-817810	-817810	-839952
V_B	-713890	-1208318*	-961099*	-879118	-937680*	-895840*	-895840	-895840	-917982*
V_C	-897760	-345500	-514659	-647738	-589176	-672840	-789541	-731191	-649653
V_D	-856320	-747214	-715200	-814732	-797186	-848199	-964900*	-906550*	-875227
V_E	-1298440*	-394998	-804012	-967984*	-850860	-934524*	-1034522*	-976172*	-942959*
V_F	-1247640*	-384772	-753212	-917184*	-800060	-883724	-883724	-883724	-861582
V_G	-563500	-252786	-376393	-417386	-388105	-409021	-470519	-405873	-397950
L^n	953920	914180	1532220	1614204	1672761	1702469	1650110	1708460	1719170
Q	7	_	7	7	7	1	_	5	0
t^n	247214	123607	40993	29281	20916	116701	58351	11071	_
α^n	2	1	1	1	1	1	0.5	0.5	_

λ_A^{n+1}	747214	623607	582614	611895	590975	590975	590975	602046	_
λ_B^{n+1}	747214	623607	582614	611895	590975	590975	590975	602046	_
λ_C^{n+1}	252786	376393	417386	388105	409021	409021	409021	397950	_
λ_D^{n+1}	747214	623607	664600	693881	714797	831498	773148	762077	_
λ_E^{n+1}	252786	376393	417386	388105	409021	409021	409021	409021	_
λ_F^{n+1}	252786	376393	417386	388105	409021	409021	409021	409021	_
λ_G^{n+1}	252786	376393	417386	388105	409021	409021	409021	397950	_