LOCATION OF AN ELECTRIC SUBSTATION IN THE KOOTOKROM TOWNSHIP, SUNYANI MUNICIPALITY
(AN APPLICATION OF PRIM'S ALGORITHM AND P-MEDIAN METHOD)

A THESIS SUBMITTED TO THE DEPARTMENT OF MATHEMATICS, IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF A MASTER OF SCIENCE (MSc.) DEGREE IN INDUSTRIAL MATHEEMATICS.

BY

TUAH AFRAM

MATHEMATICS DEPARTMENT
INSTITUTE OF DISTANCE LEARNING

NOVEMBER 2013

DECLARATION

I, the undersigned, hereby declare that the work contained in this thesis is my original work, and that any works done by others have been acknowledged and referenced accordingly.

.... ……………………….

Tuah Afram Date

Certified by:

.... ………………………..

Mr. F. K. Darkwah Date

(Supervisor)

.... ………………………

Prof. S. K. Amponsah Date

(Head of Department)

ABSTRACT

In Ghana, electricity consumption has been growing at 10 to 15 percent per annum for the last two decades. It is projected that the average demand growth over the next decade will be about six percent per year. As a result, consumption of electricity will considerably increase. The projected electricity growth assumption has profound economic, financial, social and environmental implications for the country. The aspirations of developing countries for higher living standards can only be satisfied through sustained development of their electric power markets as part of their basic infrastructure.

There is therefore the need to establish an elaborate and extensive transportation system to efficiently and effectively move electricity from producing regions to consumption regions.

The aim of this thesis is to locate an electric substation at the appropriate site in the Kootokrom Township and to model optimally, the connection of electric cables.

Prim's algorithm and P-median method have been used to realize an electric cable length of 1,642,027m and node 21 as the minimum connector and the best location respectively.

ACKNOWLEDGEMENT

My greatest appreciation goes to God Almighty whose grace and loving kindness has brought me to the dawn of another era of my life.

I am very much indebted to my parents whose strenuous effort and love has seen me through all these years of my academic career.

I want to thank Mr. K. F. Darkwah for giving me the chance to work on this topic. I owe great gratitude to the supervision, helpful comments, discussions and support he readily gave me, I am grateful to all the lecturers and workers in the Mathematics Department.

I am also very thankful to my family and friends for all the love, support and encouragement that I have received from them over the years.

DEDICATION

This study is dedicated to my brother Jonathan Tuah Badu and my beloved wife Adwoa Pomaa Danquah.

LIST OF TABLES

Table **Page**

TABLE OF CONTENT

CHAPTER ONE

1.0 INTRODUCTION

Electricity is a general term encompassing a variety of phenomena resulting from the presence and flow of electric charge. These include many easily recognizable phenomena, such as lightning, static electricity, and the flow of electrical current in an electrical wire.

Three of the most important goals of electricity distribution network design are optimality, reliability and efficiency. When a light is turned on, electricity must be readily available. Since it is not stored anywhere on a power network, electricity must somehow be dispatched immediately. With long distances of interconnected cables, electricity may travel miles out of any direct path to get where it is needed. As a result, a well designed and effective network system must be employed to reduce electricity supply problems and the associated cost in the Ghana power industry

A power supply system typically includes:

(i) Generation

(ii) Transmission

(iii) Distribution

1.0.1 GENERATING ELECTRICITY

There are many fuels and technologies that can generate electricity. Usually a fuel like coal, natural gas, or fuel oil is ignited in the furnace section of a boiler. Water piped through the boiler in large tubes is superheated to produce heat and steam. The steam turns turbine blades which are connected by a shaft to a generator. Nuclear power plants use nuclear reactions to produce heat while wind turbines use the wind to turn the generator. A generator is a huge electromagnet surrounded by coils of wire which produces electricity when rotated. Electricity generation ranges from 13,000 to 24,000 volts. Transformers increase the voltage to hundreds of thousands of volts for transmission. High voltages provide an economical way of moving large amounts of electricity over the transmission system.

1.0.2 TRANSMITTING ELECTRICITY

Once electricity is given enough voltage to travel long distances, it can be moved onto the wires or cables of the transmission system. The transmission system moves large quantities of electricity from the power plant through an interconnected network of transmission lines to many distribution centers called substations. These substations are generally located long distances from the power plant. Electricity is stepped up from lower voltages to higher voltages for transmission.

Most transmission systems use overhead lines that carry alternating current (AC). There are also overhead direct current (DC) lines, underground lines, and even under water lines. All AC transmission lines carry three phase current - three separate streams of electricity travelling along three separate conductors. Lines are designated by the voltage that they can carry. Voltage ratings are usually 345 kilovolt (kV) for primary transmission lines and 138 kV and 69 kV for sub transmission lines. Transmission voltages in Ghana are presently 69,000 volts, 161,000 volts and 220,000 volts.

1.0.3 DISTRIBUTING ELECTRICITY

The distribution system is made up of poles and wires seen in neighborhoods and underground circuits. Distribution substations monitor and adjust circuits within the system. The distribution substations lower the transmission line voltages to 34,500 volts or less. Substations are fenced yards with switches, transformers and other electrical equipment. Once the voltage has been lowered at the substation, the electricity flows to homes and businesses through the distribution system. Conductors called feeders reach out from the substation to carry electricity to customers. At key locations along the distribution system, voltage is lowered by distribution transformers to the voltage needed by customers or end-users.

1.0.4 SOURCE OF ELECTRICITY GENERATION IN GHANA

There are three sources of electricity generation in Ghana

1. Hydro
2. Thermal
3. Imports

Akosombo and Kpong dams have an effective combined capacity of 970 megawatts with an installed combined capacity of 1180 megawatts providing up to 55% of Ghana's electricity demand from the hydro sector.

Takoradi Power Company, Takoradi International Company and Tema Diesel Station which make up the thermal sector also provide up to 33% of the country's power demands and then, imports make up to about 12%

The cost of operating the complex system of power supply industries has an impact on the economic importance of any village, town or city in Ghana. Nevertheless, electricity is necessary for human survival in every society.

The chapter gives the background to the study, states the problem which the researcher is researching into and the reason for carrying out the research. It also states the justification of the study and the scope of the study.

1.1 BACKGROUND TO THE STUDY

Electricity, like other forms of energy, is a vital ingredient in the economic development of countries the world over. Not only is it a critical factor and cornerstone of the accelerated development and growth of any nation, it is also a measure of the standard and quality of life of a people. Without safe, sustained, reliable and reasonably affordable supply of electricity to meet demand, a country can hardly make progress in its economic and social development.

Population growth and economic expansion are the major factors driving growth in the electric power industry. As the number of households grow and as more and more households add on electric appliances, electric power generation capacity has to keep up. At the same time, electricity is an essential input for industrial and economic performance, although there are other things that are equally or more important.

The direct effects come from demand for electricity as households are added or businesses and industries expand. The indirect effects come from the contribution that electricity makes to our life-styles and quality of life and technological development.

The use of electricity spans through agricultural, household, industrial, recreational and environmental activities. All these human exploits require electric

1.2 PROFILE OF VOLTA RIVER AUTHORITY (VRA) AND GHANA GRID COMPANY (GRIDCO)

The Volta River Authority (VRA) was established on April 26, 1961 under the Volta River Development Act, Act 46 of the Republic of Ghana, as a body corporate with the mandate to operate mainly as a power generation, transmission and distribution utility.

In 2005, following the promulgation of a major amendment to the VRA Act in the context of the

Ghana Government Power Sector Reforms, the VRA's mandate has now been largely restricted to generation of electricity. The transmission function has been separated into an entity, designated Ghana Grid Company (GRIDCo). The VRA will continue to operate its distribution agency, the Northern Electricity Department (NED) as a subsidiary company and NED eventually merge with the Electricity Company of Ghana (ECG) into a single distribution utility after the transition period. The amendment has a key function of creating the requisite environment to attract independent power producers (IPPs) onto the Ghana energy market.

VRA has other responsibilities in furtherance of its corporate mandate. These include Lake Transportation, Tourism, and Maritime administration of the Volta Lake, Lakeside Health and Management of the Akosombo Township.

The Authority operates a total installed electricity generation capacity of 1,730MW. This is made up of two hydroelectric plants on the Volta River, with installed capacities of 1,020MW and 160MW at the Akosombo and Kpong Generating Stations, respectively, and complemented by a 330MW Combined Cycle Thermal Plant at Aboadze, near Takoradi. A further 220MW Thermal Plant, Takoradi International Company (TICO) is owned as a joint venture with TAQA, from Abu Dhabi in the United Arab Emirates. The plant, which is being operated in a simple cycle mode is being converted into a 330MW combined cycle plant, and thus bring the installed thermal generation capacity at Takoradi to 660MW.

As a short-to-medium term strategy for energy sufficiency, the VRA is developing a number of plants in Tema. These include a 110MW Tema Thermal 1 Power Plant, an 80MW Mines Reserve Plant, both commissioned in 2008; a 49.5 MW Tema Thermal 2 Power Plant, scheduled for commissioning by the end of 2009; and a 220MW Tema Thermal Power Plant, which is being developed. This will bring the total installed thermal generation capacity to 1,119.5MW

The VRA, through its distribution agency, the Northern Electricity Department (NED), is the sole distributor of electricity in the Brong-Ahafo, Northern, Upper East, Upper West, and parts of Ashanti and Volta Regions of Ghana. NED was developed as an integral part of a larger scheme, designated the Northern Electrification & System Reinforcement Project (NESRP) to extend the national electricity grid to northern Ghana. NED has a customer population close to 300,000 and a load demand of about 120MW.

GRIDCo was established in accordance with the Energy Commission Act, 1997 (Act 541) and the Volta River Development (Amendment) Act, 2005 Act 692, which provides for the establishment and exclusive operation of the National Interconnected Transmission System by an independent Utility and the separation of the transmission functions of the Volta River Authority (VRA) from its other activities within the framework of the Power Sector Reforms.

GRIDCo was incorporated on December 15, 2006 as a private limited liability company under the Companies Code, 1963, Act 179 and granted a certificate to commence business on December 18, 2006. The company became operational on August 1, 2008 following the transfer of the core staff and power transmission assets from VRA to GRIDCo

GRIDCo's main functions are to:

- Undertake economic dispatch and transmission of electricity from wholesale suppliers (generating companies) to bulk customers, which include the Electricity Company of Ghana (ECG), Northern Electricity Department (NED) and the Mines;
- Provide fair and non-discriminatory transmission services to all power market participants;

- Acquire, own and manage assets, facilities and systems required to transmit electrical energy;
- Provide metering and billing services to bulk customers;
- Carry out transmission system planning and implement necessary investments to provide the capacity to reliably transmit electric energy; and manage the Wholesale Power Market.

1.3 PROBLEM STATEMENT

The cost of electricity production for the power industry in Ghana has been increasing due to the increasing cost of operations and imported materials, and since locating a facility decides production technology and cost structure, it is important that the enormous benefits of electricity is made available to consumers through cost effective mathematical methods. It is obvious that the cost of imported materials is a big part of the rather huge cost of operations in the power industry in Ghana but before we become capable of producing some of these imported materials we must optimize the use of the imported ones.

Optimum location reduces transportation cost, labor cost and affects the industry's ability to serve customers quickly and efficiently. Accessibility to the larger populace is widely enhanced and the insecure style of living due to lack of electricity or high tariffs is minimized.

The problem at hand is to locate electrical substations appropriately, so as to optimize the cable network system in other to make electricity easily accessible and if possible a little more affordable.

1.4 OBJECTIVES

A. MAIN OBJECTIVES

From the problem statement above, the researcher seeks to address the following main objectives:

(i) To use Prim's Algorithm to determine a tree and the minimum connector for the electrical cable distribution layout for a proposed electricity company.

(ii) To use p-median method to locate on a determined tree, a distribution substation for a proposed electricity company

B. SPECIFIC OBJECTIVES

i. Data will be collected from Town and Country Planning and The Statistical Services(Sunyani Metropolitan Assembly)

ii. The problem of distribution cables will be modeled as a network problem.

iii. The Prim's algorithm will be used to determine a tree and a minimum connector for the modeled network

iv. The p-median method will be used to locate an electrical substation on one of the nodes on the modeled network

1.5 SIGNIFICANCE OF THE STUDY

VRA and GRIDCo are the companies in charge of electricity in the Sunyani municipality. Electricity from this source serves the Sunyani municipality and its environs. As the municipality grows electricity must be made available for people to utilize. Lack of electricity in the emerging suburbs will therefore affect the livelihood of the people.

Application of findings from this research will minimize the cost of operations of GRIDCo and help improve their level of efficiency. Accessibility of electricity to the general population will also be enhanced; electricity will reach out to most communities and propel people to stop resorting to acquiring benefits from electricity from other sources, such as candles, which can be dangerous sometimes.

This in effect will help to

- reduce governments spending on some of the effects of the lack of electricity such as poor academic performance, insecurity, disasters etc
- increase productivity in other sectors of the economy like health
- serve as a basis for more research to be made on this area.

1.6 ORGANIZATION

Chapter 1 presents the general introduction of the study. Chapter 2 puts a spotlight on review of the relevant literature and discusses both the theoretical and empirical background literature. Chapter 3 deals with methodology. Chapter 4 deals with data collection, solution, analysis and discussion of results. The concluding chapter 5 summarizes the findings and also provides conclusions and recommendations.

CHAPTER TWO

LITERATURE REVIEW

2.0 INTRODUCTION

We shall review some relevant literature on location methods and other forms of shortest path algorithms in this chapter.

2.1 LITERATURE REVIEW

The primary aim of the study is to locate an electrical substation to ensure the largest possible total market share for its entire chain and to optimize its network.

Schrijver (1996) cited Boruvka as the first to consider the MST. The MST algorithms that are commonly used are the Prim's (1957) and Kruskal's (1956) algorithms (cited in Jayawant and Glavin, 2009). However, Kruskal algorithm finds the minimum spanning forest if the network is not connected (Agarwal, 2010). Zachariasen (1998) noted that all known exact algorithms for the Euclidean Steiner tree problem require exponential time. The general consensus is to use heuristics and approximation algorithms. However, one of the first and easiest methods involves the use of minimal spanning trees as approximation to the Steiner tree algorithm. The Euclidean Steiner tree problem has its roots in Fermat problem whereby one finds in the plane a point, the sum of whose distances from three given points is minimal (Ivanov and Tuzhilin, 1994). The steiner ratio compares the solution of the Steiner Minimum Tree (SMT) problem to that of the MST.

Moore in (Gilbert and Pollack, 1968) put the lower bound of the Steiner ratio to be 0.5 while Gilbert and Pollack (1968) conjectured the upper bound to be 0.866. This means that the SMT can be shorter than the MST by at most 13.4% (Bern and Graham, 1989). Dott (1997) reduced the 952 km Palliser pipeline network to 832 km using the haMSTer program, which is based on Prim algorithm. This was a 13% reduction over the original length. The author conjectured that if the SMT is used in addition to the haMSTer then 721 km will be realized. Brimberg *et al*. (2003), presented the optimal design of an oil pipeline network for the South Gabon oil field. The original design covered thirty-three (33) nodes with hundred and twenty-nine (129) possible arcs having total distance of 191.1 km. Using a variation of Prim's algorithm, this reduced the connection to 121.6 km, which was a reduction of 36.4% of the total distance to be covered. Dott (1997) obtained an Optimal Design of Natural Gas Pipeline of Amoco East Cross field Gas pipeline project, (Alberta, Canada). The pipeline, which covers a distance of 66 km was reduced to 48.9 km with the use of the haMSTer program software. Steiner tree algorithm was later used to reduce the MST created by the hamster to 48.84 km. This was 1% reduction over the MST. Arogundade and Akinwale (2009) used Prim's algorithm to find the shortest distance between 88 villages connected by 96 roads of Odeda local government map, Nigeria, and arrived at an MST of 388,270 m. Nie *et al*. (2000) combined an algorithm of rectilinear Steiner tree and the constraints and connectivity reliability of road network to obtain a new method of rural road network layout designing in a county area. The method is applied to the rural road network layout designing of Shayang County, China.

2.1.1 NETWORK SYSTEMS AND SHORTEST PATH ALGORITHMS

The visualization of network has become one of the essential characteristics of urban cable network management system (Xu et al, 2008). But for the distribution of related resources (e.g., data storage, modeling platform, etc.) and complex structure, the design and implementation of a cable network visualization application is always a challenge. Barab (2002), writes, "The diversity of networks in business and the economy is astounding. There are policy networks, ownership networks, collaboration networks, organizational networks, network marketing--you name it. It would be impossible to integrate these diverse interactions into a single all-encompassing web. Yet no matter what organizational level we look at, the same robust and universal laws that govern nature's webs seem to greet us."

In graph theory, the shortest path problem is the problem of finding a path between two vertices (or nodes) such that the sum of the weights of its constituent edges is minimized. An example is finding the quickest way to get from one location to another on a road map; in this case, the vertices represent locations and the edges represent segments of road and are weighted by the time needed to travel that segment.

Formally, given a weighted graph (that is a set V of vertices, a set E of edges, and a real-valued weight function $f : E \rightarrow R$), and one element v of V, find a path P from v to a v' of V so that

$$\sum_{P=p} f(p)$$

is minimal among all paths connecting v to v'

According to Mouli et al., (2010), Shortest Path problems are among the most studied network flow optimization problems with interesting applications in a wide range of fields. One such application is in the field of GPS routing systems. These systems need to quickly solve large shortest path problems but are typically embedded in devices with limited memory and external storage. Conventional techniques for solving shortest paths within large networks cannot be used as they are either too slow or require huge amounts of storage. In their project they tried to reduce the runtime of conventional techniques by exploiting the physical structure of the road network and using network pre-processing techniques. Their algorithms may not guarantee optimal results but can offer significant savings in terms of memory requirements and processing speed. Their work used heuristic estimates to bind the search and directs it towards a destination. They also associated a radius with each node that gives a measure of importance for roads in the network. The farther they got from either the origin or destination the more selective they became about the roads they traveled with greater importance (i.e. roads with larger radii). By using these techniques they were able to dramatically reduce the runtime performance compared to conventional techniques while still maintaining an acceptable level of accuracy.

In recent years, the static shortest path (SP) problem has been well addressed using intelligent optimization techniques e.g., artificial neural networks, genetic algorithms (GAs), particle swarm optimization, etc. However, with the advancement in wireless communications, more and more mobile wireless networks appear, e.g., mobile networks.

2.1.2 PRIM'S ALGORITHM

Prim's algorithm is an algorithm that finds a minimum spanning tree for a connected weighted undirected graph. It finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. Prim's algorithm is an example of a greedy algorithm. The algorithm was developed in 1930 by Czech mathematician Vojtech Jarnik and later independently by computer scientist Robert C. Prim in 1957 and rediscovered by Edsger Dijkstra in 1959. Therefore it is also sometimes called the DJP algorithm, the Jarnik algorithm, or the Prim-Jarnik algorithm (Prim, 1957).

Gonina et al., (2007) described parallel implementation of Prim's algorithm for finding a minimum spanning tree of a dense graph. Their algorithm uses a novel extension of adding multiple vertices per iteration to achieve significant performance improvements on large problems (up to 200,000 vertices). They described several experimental results on large graphs illustrating the advantages of our approach on over a thousand processors.

Gloor et al., (1993) described a system for visualizing correctness proofs of graph algorithms. The system has been demonstrated for a greedy algorithm, Prim's algorithm for finding a minimum spanning tree of an undirected weighted graph. They believed that their system is particularly appropriate for greedy algorithms, though much of what they discuss can guide visualization of proofs in other contexts. While an example is not a proof, our system provides concrete examples to illustrate the operation of the algorithm. These examples can be referred to by the user interactively and alternatively with the visualization of the proof where the general case is portrayed abstractly. Martel et al., (2002) studied the expected performance of Prim's minimum spanning tree (MST) algorithm implemented using ordinary heaps. We show that this implementation runs in linear or almost linear expected time on a wide range of graphs. This helps to explain why Prim's algorithm often beats MST algorithms which have better worst-case run times.

2.1.3 DIJKSTRA'S ALGORITHM

This algorithm solves the single-source shortest-path problem when all edges have non-negative weights. It is a greedy algorithm and similar to Prim's algorithm. Algorithm starts at the source vertex, s, it grows a tree, T, that ultimately spans all vertices reachable from S. Vertices are added to T in order of distance i.e., first S, then the vertex closest to S, then the next closest, and so on.

Dijkstra's algorithm, conceived by Dutch computer scientist Edsger Dijkstra in 1959, is a graph search algorithm that solves the single-source shortest path problem for a graph with nonnegative edge path costs, producing a shortest path tree. This algorithm is often used in routing. An equivalent algorithm was developed by Moore, (1959).

For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex. It can also be used for finding costs of shortest paths from a single vertex to a single destination vertex by stopping the algorithm once the shortest path to the destination vertex has been determined. For example, if the vertices of the graph represent cities and edge path and costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities. As a result, the shortest path first is widely used in network routing protocols, most notably OSPF (Open Shortest Path First)

The literature makes is abundantly clear that the procedure commonly known today as Dijkstra's Algorithm was discovered in the late 1950s, apparently independently, by a number of analysts. There are strong indications that the algorithm was known in certain circles before the publication of Dijkstra's famous paper. It is therefore somewhat surprising that this fact is not manifested today in the "official" title of the algorithm (http://en.wikipedia.org).

Dijkstra, (1959) submitted his short paper for publication in Numerische Mathematik June 1959. In November 1959, Pollack and Wiebenson (1960) submitted a paper entitled Solutions of the shortest- route problem - a review to the journal Operations Research. This review briefly discusses and compares seven methods for solving the shortest path problem.

Sniedovich, (2006) described Dijkstra's Algorithm as one of the most popular algorithms in computer science. It is also popular in operations research. It is generally viewed and presented as a greedy algorithm. In their paper they attempted to change this perception by providing a dynamic programming perspective on the algorithm.

In Open Shortest Path First (OSPF), Dijkstra's shortest path algorithm, which is the conventional one, finds the shortest paths from the source on a program counter-based processor. The calculation time for Dijkstra's algorithm is 0 (N2) when the number of nodes is *N*. When the network scale is large, calculation time required by Dijkstra's algorithm increases rapidly.

2.1.4 KRUSKAL ALGORITHM

Kruskal's algorithm is an algorithm in graph theory that finds a minimum spanning tree for a connected weighted graph. This means it finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. If the graph is not connected, then it finds a minimum spanning forest (a minimum spanning tree for each connected component). Kruskal's algorithm is an example of a greedy algorithm.

It maintains a set of partial minimum spanning trees, and repeatedly adds the shortest edge in the graph whose vertices are in different partial minimum spanning trees.

This algorithm first appeared in Proceedings of the American Mathematical Society, pp. 48–50 in 1956, and was written by Joseph Kruskal.

Description

- ✓ create a forest F (a set of trees), where each vertex in the graph is a separate tree
- ✓ create a set S containing all the edges in the graph
- ✓ while S is nonempty and F is not yet spanning
    - i. remove an edge with minimum weight from S
    - ii. if that edge connects two different trees, then add it to the forest, combining two trees into a single tree
    - iii. otherwise discard that edge.

At the termination of the algorithm, the forest has only one component and forms a minimum spanning tree of the graph (Cormen et al., 2001)

This minimum spanning tree algorithm was first described by Kruskal in 1956 in the same paper where he rediscovered Jarnik's algorithm. This algorithm was also rediscovered in 1957 by Loberman and Weinberger, but somehow avoided being renamed after them. The basic idea of the Kruskal's algorithms is as follows: scan all edges in increasing weight order; if an edge is safe, keep it

Kruskal's Algorithm is directly based on the generic minimum spanning tree (MST) algorithm. It builds the MST in forest.

Initially, each vertex is in its own tree in the forest. Then, the algorithm considers each edge in turn, ordering by increasing weight. If an edge (u, v) connects two different trees, then (u,v) is added to the set of edges of the MST, and two trees connected by an edge (u, v) are merged into a single tree on the other hand, if an edge (u, v) connects two vertices in the same tree, then edge (u, v) is discarded.

A little more formally, given a connected, undirected, weighted graph with a function

$w : E \to R$.

i. Starts with each vertex being its own component.
ii. Repeatedly merges two components into one by choosing the light edge that connects them (i.e., the light edge crossing the cut between them).
iii. Scans the set of edges in monotonically increasing order by weight.
iv. Uses a disjoint-set data structure to determine whether an edge connects vertices in different components. (Kruskal, 1956)

2.1.5 OTHER FORMS OF SHORTEST PATH ALGORITHMS

Bellman-Ford algorithm solves the single source problem if edge weights are negative.

- ✓ A search algorithm solves for single pair shortest path using heuristics to speed up the search.
- ✓ Floyd- Warshall algorithm solves all pair’s shortest paths.
- ✓ Johnson's algorithm solves all pair’s shortest paths, and may be faster than Floyd-Warshall on sparse graphs.
- ✓ Perturbation theory finds (at worst) the locally shortest path.

2.1.6 THE P-MEDIAN METHOD

The p-median problem is, with no doubt, one of the most studied facility location models. Basically, the p-median problem seeks the location of a given number of facilities so as to minimize some measure of transportation costs, such as distance or travel time. Therefore, demand is assigned to the closest facility.

The p-median problem is widely used in both public and private sector location decisions. Its uses include the original practical case suggested by Hakimi, which is locating a number of switching centers on a telephone network, as well as a large number of other applications, both geographical and not-geographical. Among the first, it is worth mentioning the location of public facilities so that the distance the public must travel to them is minimized: schools and hospitals are a typical example.

Non-geographical applications arise for example when there is the need of grouping or clustering objects, tasks, events, and so on. Why is it called p-*median*? The *median vertex* is the vertex of a network or graph for which the sum of the lengths of the shortest paths to all other vertices is the smallest. Locating a school on the median vertex of a network in which edges or arcs represent roads and each node represents a child, minimizes the total distance that children have to walk to go to that school. Or, if each node represents a customer, and a maintenance center housing a vehicle has to be located on some vertex of the network, the median vertex will be the location that minimizes the total distance traveled by the vehicle, if all customers have to be served, one at a time. On a network, finding the median vertex solves a problem similar to that posed by Fermat on a (Euclidean) plane in the 1600's, consisting of finding the location of the point on a plane which minimizes the sum of its distances to three points whose location is known. Weber, in the early 20th century, generalized this problem by adding weights to them, which could represent amount of demand or population aggregated at the points. If a facility is located at this weighted median, it will satisfy the demand of the three points with the minimal transportation cost. Later, the Weber problem was generalized to include more than three demand points, and to locate more than one facility. The 3 version with multiple facilities became known as the Multi-Weber problem. In the 20th century, Cooper (1963, 1964) provided heuristic solutions for it.

Although now it seems a natural step, Hakimi did not formulate the *p*-median as an integer programming problem. This was first done by ReVelle and Swain (1970) who, not being familiar with the results of Hakimi, assumed node-only location of what they called central facilities. This formulation opened a new line in the search of solution procedures for the *p*-median problem.

The *p*-median can be formally stated in words as:

"Given the location of *n* points that house known amounts of demand, designate *p* of these points as facilities and allocate each demand to a facility, in such a way as to minimize the total weighted distance between demands and facilities"

This problem can be solved using different methods. Total enumeration is always an alternative, although its complexity makes this method useless when the problem grows. The first methods that were proposed for solving the p-median were heuristic. Among these, Maranzana (1964) describes a heuristic that randomly locates the *p* facilities and then solves the allocation problem (which has a polynomial complexity). Each facility in this initial solution serves a set or cluster of demands. Once this solution is found, Maranzana iteratively relocates the facilities within each cluster if it improves the solution, and reallocates demands keeping fixed the locations of the facilities, which potentially changes the clusters. A stable solution is reached, which is the best, but not necessarily optimal. Teitz and Bart (1968) proposed a method called "vertex substitution", that, starting from a known solution, relocates facilities one by one (and reallocates demands), whenever this relocation improves the solution. When no more improvements are possible by this method, a good solution has been reached. The works of Maranzana (1964) and Teitz and Bart (1968) were known when ReVelle and Swain (1970) proposed an optimal procedure for the *p*-median, based on linear programming and branch and bound.

ReVelle and Swain (1970) observed that when branch-and-bound was required to resolve fractional variables produced by linear programming, the extent of branching and bounding needed was very small, always less than 6 nodes of a branch-and-bound tree. Therefore, the expanded form of the constraint makes integer solutions far more likely. Infact, in this formulation, only the location variables *yj* need to be declared binary, as ReVelle and Swain (1970) proved. Morris (1978) , solved 600 randomly generated problems of the very similar Simple Plant Location Problem with the extended form of the constraint and found that only 4% did require the use of branch-and-bound to obtain integer solutions. Rosing et al. (1979c) proposed several ways to reduce both the number of variables and constraints in order to make the P-Median Problem more tractable.

Since these early contributions, many methods have been proposed for solving this problem, as well as variations of the problem that consider additional constraints, cost functions and assignment policies.

Louis Hakimi was one of the first researchers addressing the problem on a network. In his 1964 paper, the best location of a facility was sought, considering that all demand must be attended. Similarly to the problem on a plane, the demand is distributed over the region of interest. In the network version of 6 the problem, demand is located only on vertices or nodes, each of them having a weight representing the total amount of demand that it houses. In Hakimi's version, the facility can be located on a node or at a point on an edge of the network, distinction that does not exist when the problem lies on the plane. Hakimi proved, however, that there is always an optimal solution at a node. The problem consists in finding this optimal location, in such a way that the sum of the distances between the facility and each demand node, weighted by the amount of demand, is minimum. Because of this minimization of a sum of terms, the problem has also been called "minsum", or "minisum" problem. In 1965, Hakimi was able to generalize his main result (node solution) to the case of multiple facilities. Now, the problem consists of finding the locations of *p* facilities, in such a way that the sum of the weighted distances between each demand node and its closest facility is the least. He called this problem the *p*-median. Note that the presence of more than one facility introduces an additional level of difficulty, since the solution must now answer to two questions: where to locate the *p* facilities – the "location" problem; and what demand node is assigned to which facility –the "allocation" problem. In the Hakimi (1965) version, the allocation problem is defined as assignment of demand nodes to their closest facilities. However, the location of multiple facilities allows different possibilities, including allocation of a demand to more than one facility, which could be optimal if facilities have a limited capacity, or if customers located at demand nodes can choose different facilities in different opportunities.

CHAPTER THREE

METHODOLOGY

3.0 INTRODUCTION

This chapter introduces the methods employed to form the basis for analysis in this study. The shortest path algorithms and location methods are used and the solution technique in the work is described in detail. The mathematical formulation of the location models and the total distances to be covered and calculations are outlined.

The solution to the model forms the basis of the results and analysis that will be presented in Chapter four (4). The purpose of this research is to develop network models to minimize the total cost in laying electrical cables and to allocate appropriately a substation, to enhance accessibility of electricity in the area.

The theory in this research is that an optimal location can be chosen for a substation and the length of cable for electricity distribution will be minimal in the area. Locations are selected from various suburbs in the area. With the help of the capital map, a set of points were selected to represent the location points.

3.1 GRAPH THEORY.

A graph theory is the study of points and lines. In particular, it involves the ways in which set of points, called vertices, can be connected by lines or edges, called edges. Graphs in this context differ from the more familiar coordinate plots that portray mathematical relations and functions.

3.2 UNDIRECTED GRAPHS

An undirected graph is a graph in which the nodes are connected by undirected arcs. An undirected arc is an edge that has no arrow. Both ends of an undirected arc are equivalent--there is no head or tail. Therefore, we represent an edge in an undirected graph as a set rather than an ordered pair:

An undirected graph is an ordered pair G = (V, E) with the following properties:

(i)The first component, V, is a finite, non-empty set. The elements of V are called the vertices of *G*.

(ii)The second component, E, is a finite set of sets. Each element of E is a set that is comprised of exactly two (distinct) vertices. The elements of E are called the edges of *G*.

3.3 DEGREE

Degree is the number of edges which connect a node.

In Degree is the number of edges pointing to a node.

Out Degree is the number of edges going out of a node.

3.4 WEIGHTED GRAPHS

A weighted graph is a graph in which each edge has a weight (some real numbers). The weight of a graph is the sum of all the weights of all edges.

3.5 CONNECTED GRAPHS

A graph is called connected if given any two vertices Vi, Vj, there is a path from Vi to Vj.

THEOREM: Every connected graph has a spanning tree.

3.6 MINIMUM SPANNING TREE

Let G = (V, E) be a simple, connected, undirected graph that is not edge-weighted.

A spanning tree of G is a free tree (i.e., a tree with no root) with | V | - 1 edges that connects all the vertices of the graph.

Thus a minimum spanning tree for G is a graph, T = (V', E') with the following properties:

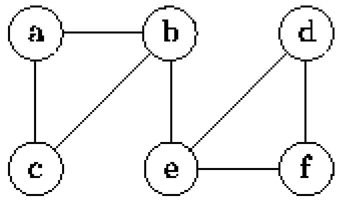
- V' = V
- T is connected
- T is acyclic.
- A spanning tree is called a tree because every acyclic undirected graph can be viewed as

a general, unordered tree. Because the edges are undirected, any vertex may be chosen to serve as the root of the tree.

3.7 CONSTRUCTING MINIMUM SPANNING TREES

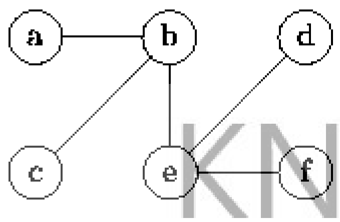
Any traversal of a connected, undirected graph visits all the vertices in that graph. The set of edges which are traversed during a traversal forms a spanning tree.

Fig 3.1



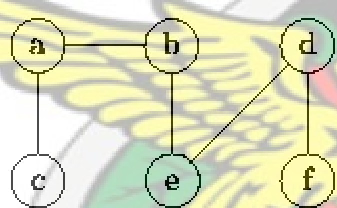
- For example, Fig 3.2 below shows the spanning tree obtained from a breadth-first traversal starting at vertex b.

Fig 3.2



Similarly, Fig 3.3 shows the spanning tree obtained from a depth-first traversal starting at vertex e.

Fig 3.3



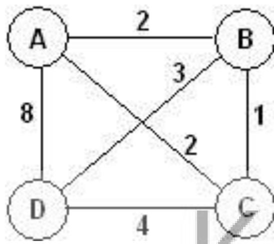
3.8 MINIMUM-COST SPANNING TREE

For an edge-weighted, connected, undirected graph, G, the total cost of G is the sum of the weights on all its edges.

A minimum-cost spanning tree for G is a minimum spanning tree of G that has the least total cost.

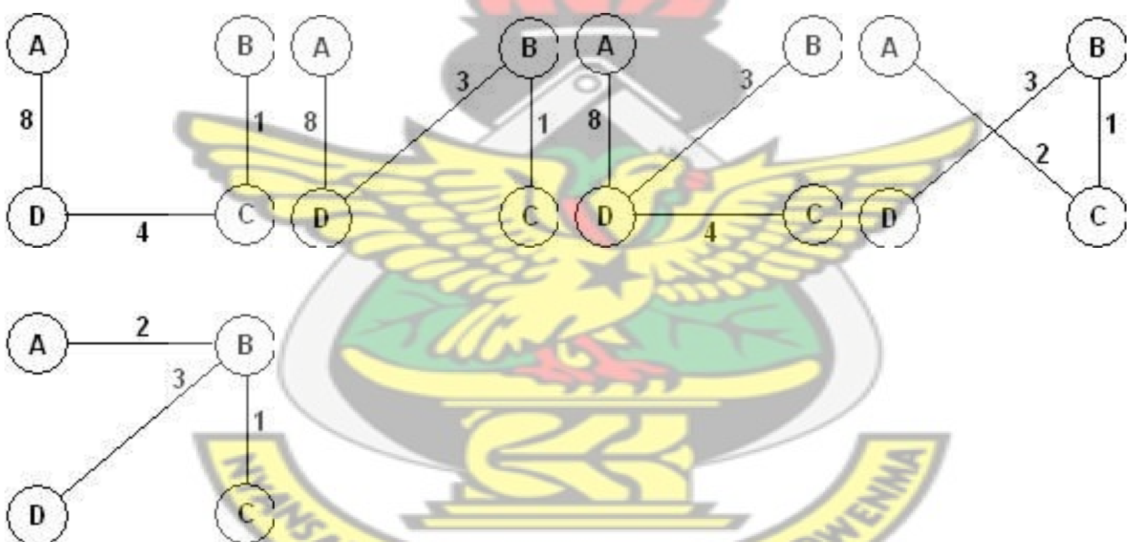
Example: The graph in Figure 3.4 below has 16 spanning trees.

Fig 3.4



Some of these spanning trees are indicated in Figure 3.5 below

Fig 3.5



The graph has two minimum-cost trees

Fig 3.6

These minimum-cost spanning trees are useful in building networks that join a number of neighborhoods with minimum cost.

3.9 CYCLES

A cycle in a graph is a collection of edges which make it possible to start a vertex and move to other vertices along edges, returning to the start vertex without repeating either edges or vertices(other than the start vertex)

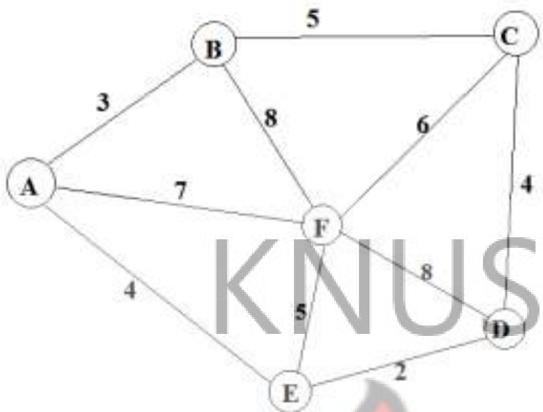
3.10 KRUSKAL'S ALGORITHM

1. Select the shortest edge in the network
2. Select the next shortest edge which does not create a cycle
3. Repeat step 2 until all vertices have been connected

If for instance, a cable company want to connect five villages to their network which currently extends to the market town of A then the problem at stake is to find the minimum length of cable required.

We model the situation as a network, and then the problem is to find the minimum connector using Kruskal's Algorithm.

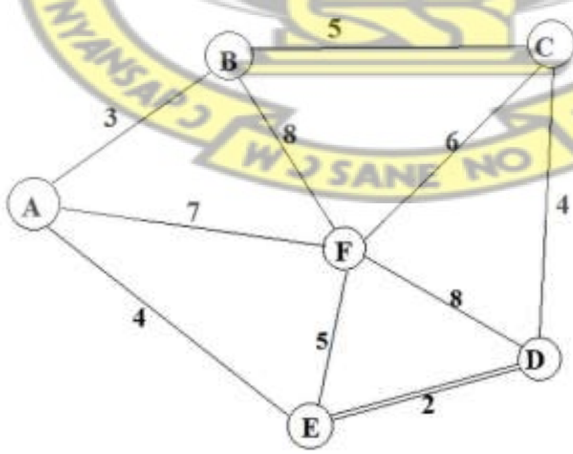
Fig 3.7



List the edges in order of size and select the shortest,

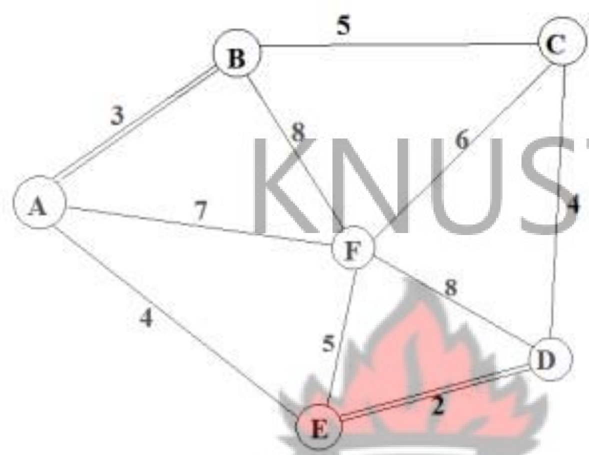
$ED = 2$ $AB = 3$ $AE = 4$ $CD = 4$ $BC = 5$ $EF = 5$ $CF = 6$ $AF = 7$ $BF = 8$ $CF = 8$. $ED = 2$ is selected

Fig 3.8



Select the next shortest edge which does not create a cycle

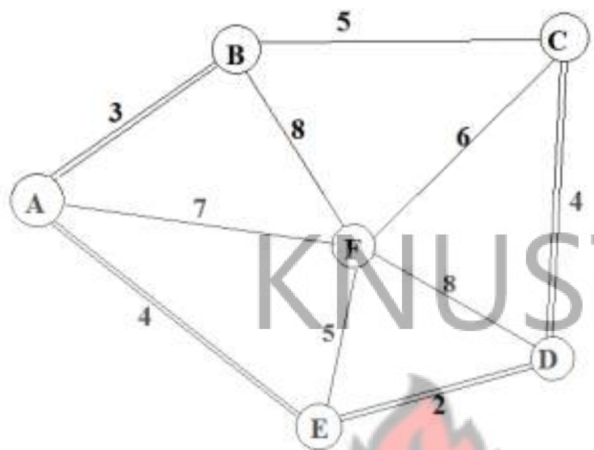
Fig 3.9



$ED = 2$ $AB = 3$ is selected

Select the next shortest edge which does not create a cycle

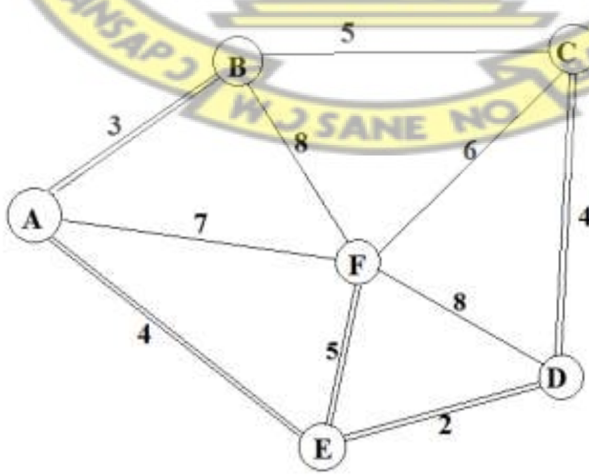
Fig 3.10



$ED = 2$ $AB = 3$ $CD = 4$ and $AE = 4$ are also selected.

The next shortest distances are $BC = 5$ and $EF = 5$ but BC will form a cycle when connected hence $EF = 5$ selected

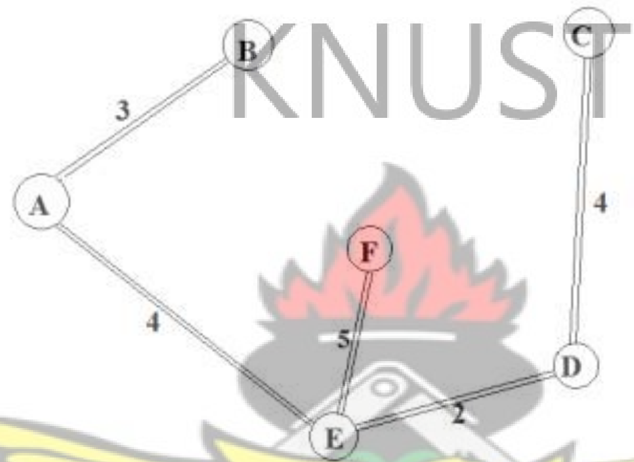
Fig 3.11



All vertices have been connected and a tree is formed as indicated in Figure 3.12

Total weight of tree $= 2 + 3 + 4 + 4 + 5 = 18$

Figure 3.12



3.11 PRIM'S ALGORITHM FORMULATION

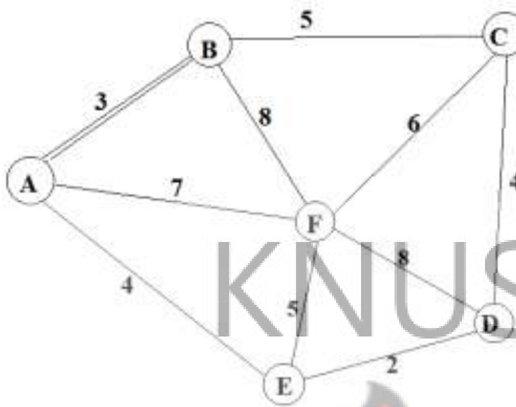
a) METHOD ONE

In method one of the solution using Prims Algorithm the following steps will be followed;

1. Select any vertex
2. Select the shortest edge connected to that vertex
3. Select the shortest edge connected to any vertex already connected
4. Repeat step 3 until all vertices have been connected

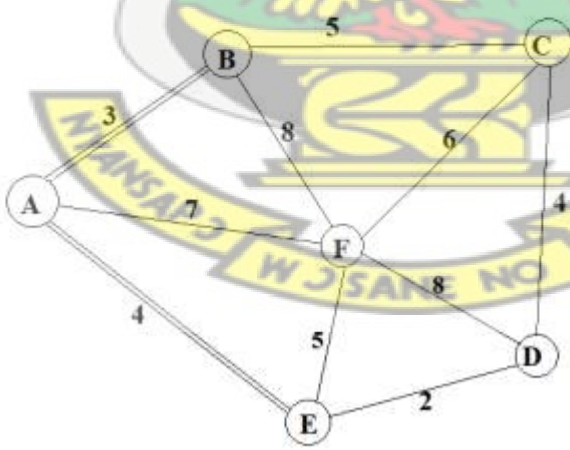
Select any vertex say **A** then select the shortest edge connected to that vertex $AB = 3$.

Fig 3.12



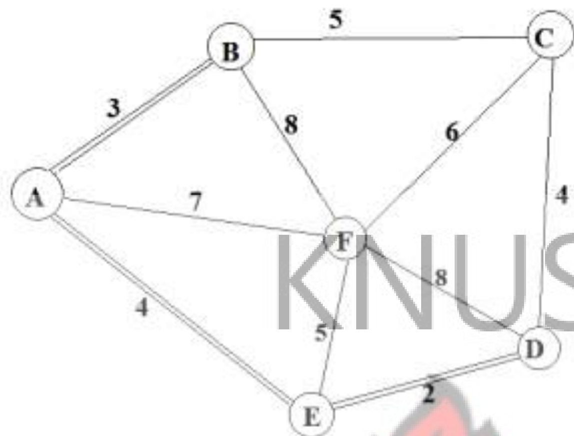
Select the shortest edge connected to any vertex already connected. $AE = 4$ is connected on the tree we are forming.

Fig 3.13



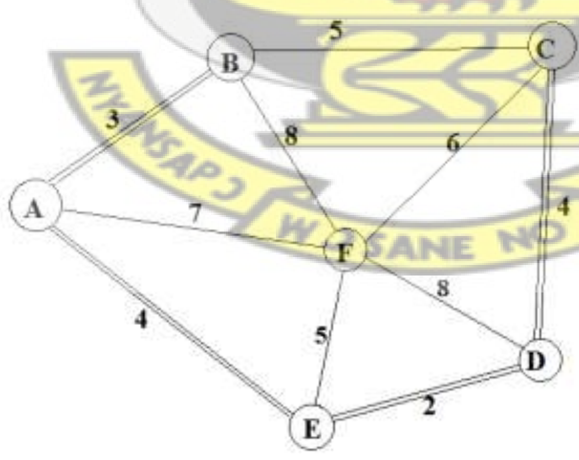
Select the shortest edge connected to any vertex already connected. $ED = 2$ is connected.

Fig 3.14



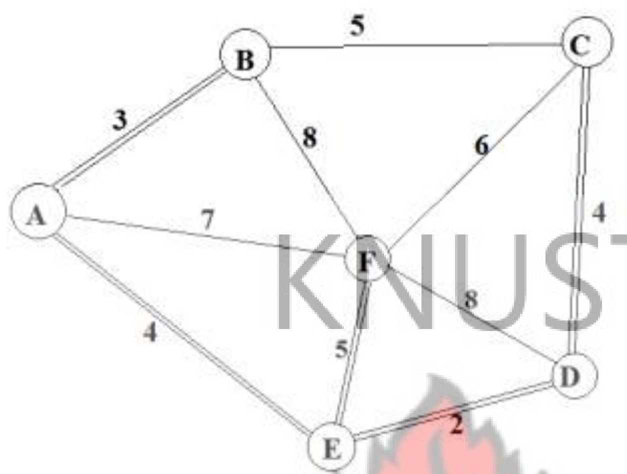
Select the shortest edge connected to any vertex already connected. $DC = 4$ is connected on the tree.

Fig 3.15



Select the shortest edge connected to any vertex already connected. $EF = 5$

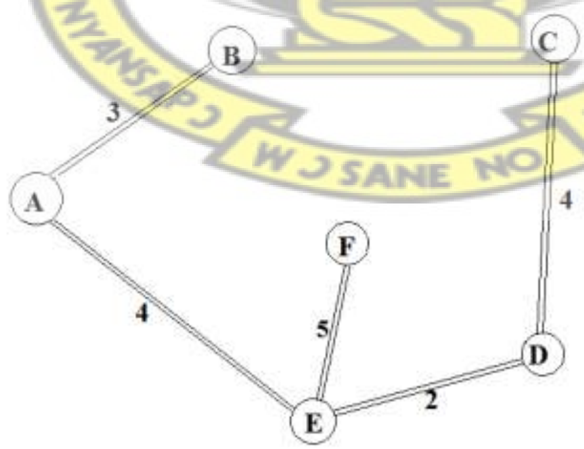
Fig 3.16



All vertices have been connected. Total weight of tree $= 3 + 4 + 2 + 4 + 5 = 18$

The minimum connector forms a tree indicated in Figure 3.17

Fig 3.17



b) METHOD TWO

The prim's algorithm can also be applied to find the minimum connector from the point A to all other points in the network using method two, as follows;

Starting from A let $S_1 = \{A, B\}$ where B is the shortest distance of all points connected to A

and $S_2 = \{C, D, E, F\}$ representing the set of all other remaining points on the network.

Then pair each member of S_1 to all other elements of S_2 indicating their respective distances and connect the shortest in distance among them.

$AC = \propto$ $BC = 5$

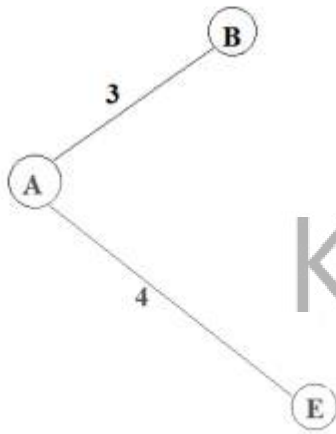
$AD = \propto$ $BD = \propto$

$AE = 4$ $BE = \propto$

$AF = 7$ $BF = 8$

Since AE has the shortest distance, we connect E to A and migrate E to join A, B in S_1 and continue the process.

Fig 3.18



$S_1 = \{A, B, E\}$

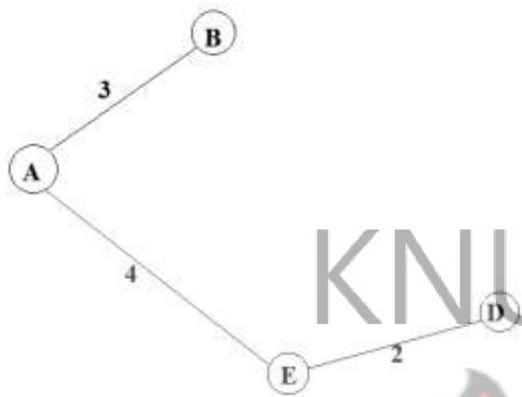
$S_2 = \{C, D, F\}$

We pair again using our new S_1 and S_2

$AC = \propto$	$BC = 5$	$EC = \propto$
$AD = \propto$	$BD = \propto$	$ED = 2$
$AF = 7$	$BF = 8$	$EF = 5$

$ED = 2$ and the shortest, so we connect D to E in Fig 3.19 and migrate D to S_1

Fig 3.19



$S_1 = \{A, B, E, D\}$

$S_2 = \{C, F\}$

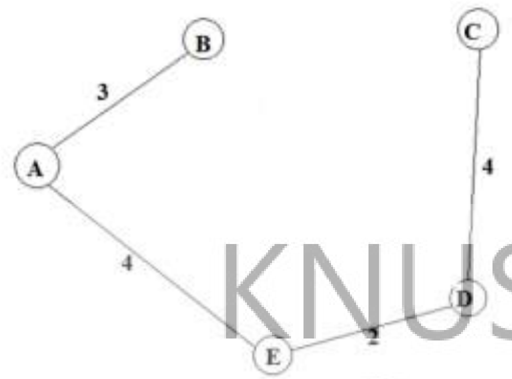
We pair again using our new S_1 and S_2

$AC = \propto$ $\quad BC = 5$ $\quad EC = \propto$ $\quad DC = 4$

$AF = 7$ $\quad BF = 8$ $\quad EF = 5$ $\quad DF = 8$

$DC = 4$ is the shortest therefore C is migrated to S_1 and then connected to D in Fig 3.20

Fig 3.20



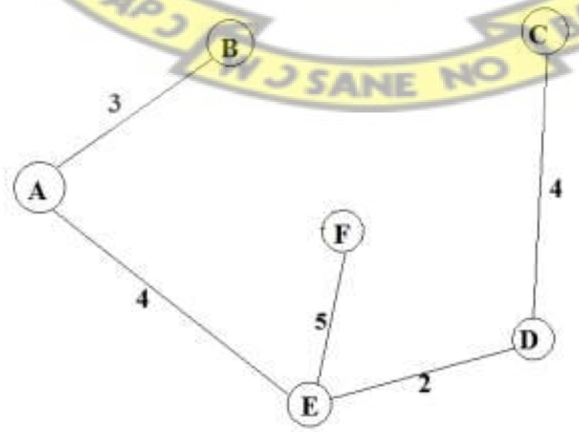
$S_1 = \{A, B, E, D, C\}$

$S_2 = \{F\}$

$AF = 7$ $BF = 8$ $EF = 5$ $DF = 8$ $CF = 6$

Since $EF = 5$ is the shortest among the remaining, F is finally connected to E in Fig 3.21

Fig 3.21



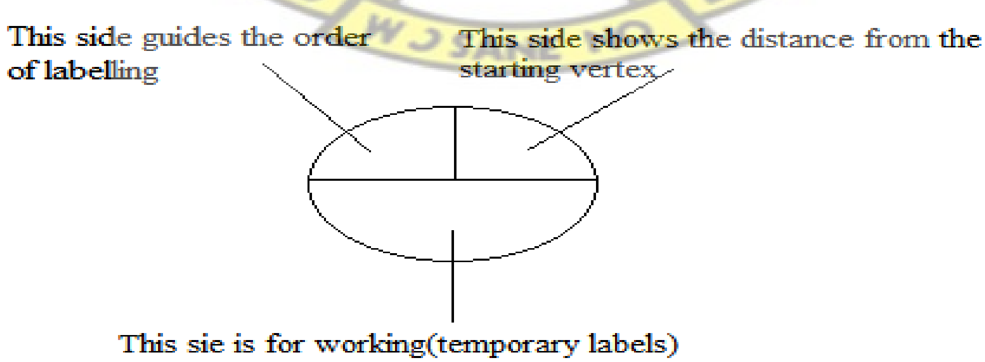
Total weight on tree $= 3 + 4 + 2 + 4 + 5 = 18$

The Kruskal and Prim algorithms will always give solutions with the same length. They will usually select edges in a different order. Occasionally they will use different edges – this may happen when you have to choose between edges with the same length. In this case there is more than one minimum connector for the network.

3.12 DIJKSTRA'S ALGORITHM

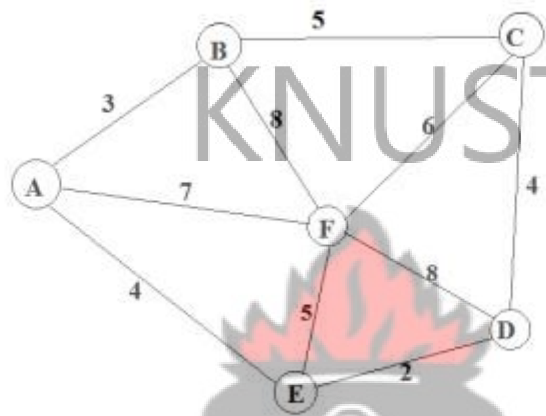
1. Assign the permanent label 0 to the starting vertex.
2. Assign temporary labels to all the vertices that are connected directly to the most recently permanent labeled vertex.
3. Choose the vertex with the smallest temporary label and assign a permanent label to that vertex.
4. Repeat steps 2 and 3 until all vertices have permanent labels
5. Find the shortest path by tracing back through the network

The algorithm gradually changes all temporary labels into permanent ones



We will apply Dijkstra's algorithm to the network in Fig 3.22 to find the shortest path from A to D

Fig 3.22

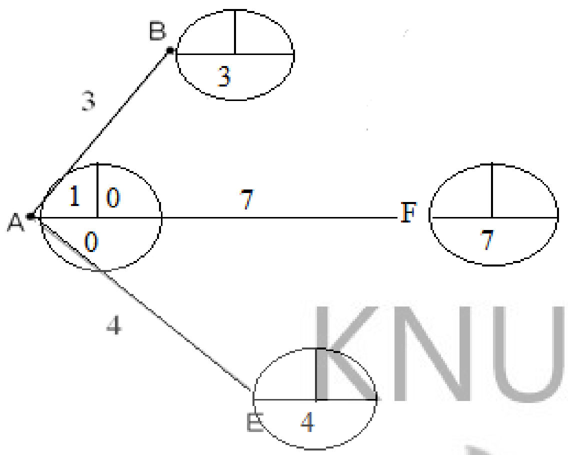


We can apply Dijkstra's algorithm to the network below to find the shortest path from A to D as follows;

1. Start by giving a permanent label 0 to our starting vertex A.
2. Next we consider the vertices which can be reached directly from A and give them temporary labels equal to their distances from A and give them temporary labels equal to their distances from A

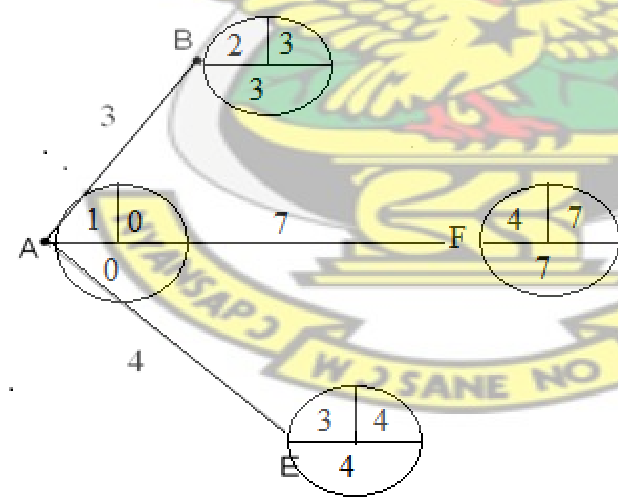
This means that B is labeled 3, E labeled 4 and F labeled 7

Fig 3.23



We now select the smallest among them and then assign a permanent label to it. In this case we assign a permanent label of 3 to B, 4 to E and then 7 to F.

Fig 3.24



Again we consider all vertices that are connected directly to our newly-labeled vertices B, E, F and they are C and D. We then calculate the shortest distance from A to C via B and F, and then A to D via E because has also been permanently labeled.

$A \rightarrow B \rightarrow C = 3 + 5 = 8$

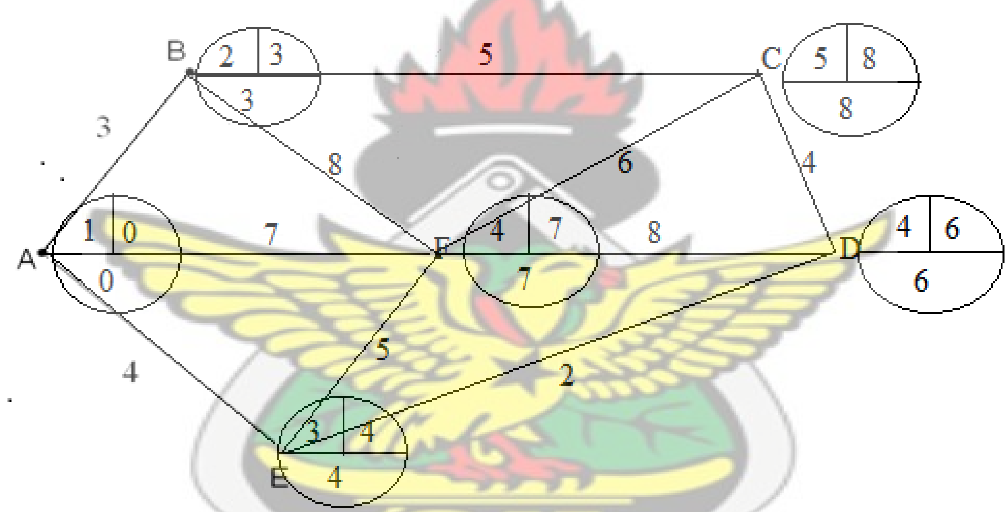
$A \rightarrow F \rightarrow C = 7 + 6 = 13$

$A \rightarrow F \rightarrow D = 7 + 8 = 15$

$A \rightarrow E \rightarrow D = 4 + 2 = 6$

Permanent labels are then given to D and then finally C.

Fig 3.25



Now all the vertices have permanent labels and we can see that the shortest distance from A to D is A→E→D with length 6.

Obviously, Djikstra's algorithm cannot be used for minimum connector problems because it does not connect all the vertices, it will however be perfect if we were finding the shortest path to D.

3.13 P-MEDIAN PROBLEM FORMULATION

The p-median problem involves the location of a fixed number of facilities in such a manner that the total weighted distance of all users assigning to their closest facility is minimized. By minimizing weighted distance, accessibility is maximized. As the weighted distance is minimized, so is the average distance that demand is away from the closest facility.

The appropriate objective then is to find the minimum of the calculated weighed (w_{i}) distances for all the potential sites.

The problem involves placing p facilities so that the total distance to one of those facilities is minimized. The model can represented mathematically as follows.

$$Minimize \quad \sum \sum_{i}^{n} h_i d_{ij} y_{ij} \qquad (1)$$

$$Subject\ to\colon \sum_i X_i = n \qquad (2)$$

$$\sum_i Y_{ij} = 1 \qquad (3)$$

$$X_i = \{0,1\} \qquad (4)$$

$$Y_{ij} = \{0,1\} \qquad (5)$$

Where $w_i =$ weighed distance for site i

i = index of selected site

j = index of site for potential facility placements

n= number of site(s) to locate facility

h_i =demand at node i

d_y= distance between node I and node j

x_i = 1,0 where 1 implies a potential facility is located at site j and 0 implies no facility is located at site j

Y_{ij} = 1,0 where 1 implies site i is served by a facility at site j and 0 implies site i is not served by a facility at site j

The objective function (1) seeks to minimize the total distance covered by other sites to access a facility at a selected site. Constraint (2) limits the number of facilities to be placed to some number n. Constraint (3) assures that each node i is served by one facility. Constraints (4) and (5) define the decision variables X and Y.

3.14 COMPUTATION OF THE WEIGHTED DISTANCE BY P-MEDIAN

Given the locations, the distances to cover to have access to a facility at a selected location and the population of people at the locations;

$$w_i = \sum_{i,j}^{n} h_i d_{ij} y_{ij}$$

can be used to find the weighted distances, and the minimum selected as the optimal site.

For instance; if four villages P, Q, R and S forming a district are connected with known distances and population size, then a facility can be located appropriately at one of the villages for use by the district using the p-medain method.

Fig 3.26 network of villages in the district

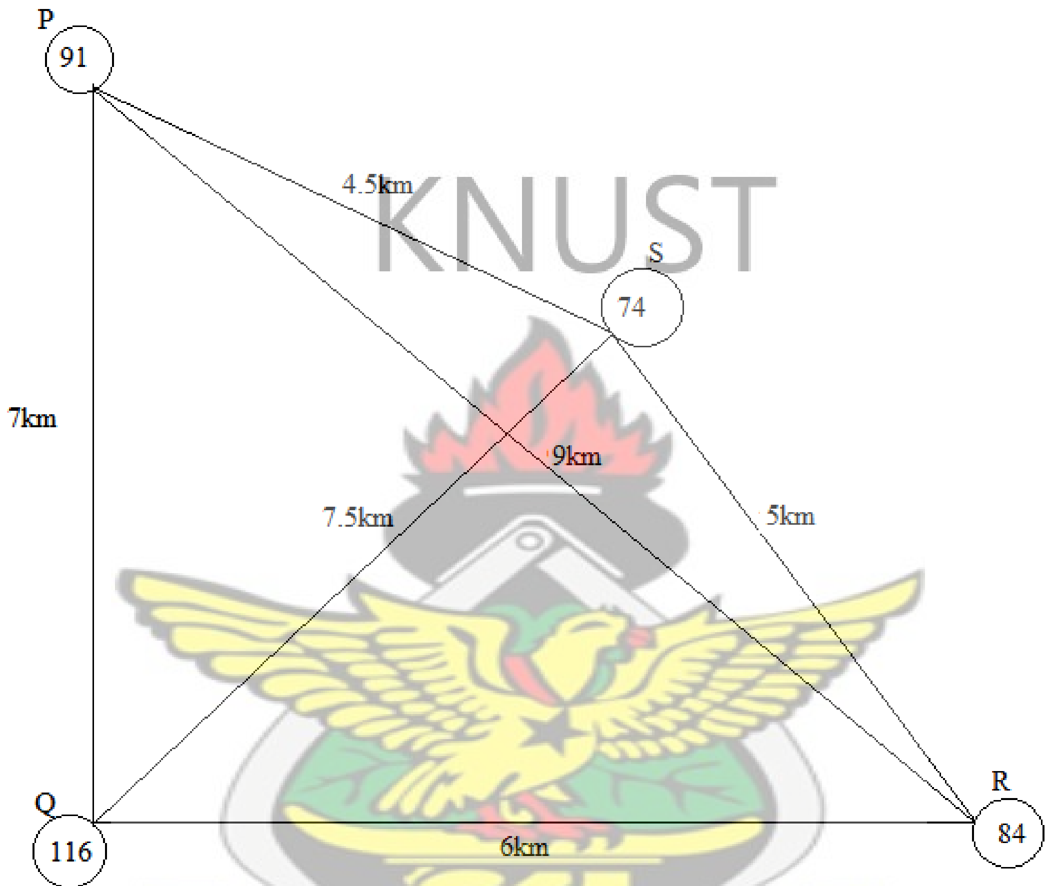


Table 3.1 Distance between villages

	j					
		SITE	1	2	3	4
	SITE	VILLAGE	P	Q	R	S
i	1	P	0	7	9	4.5
	2	Q	7	0	6	7.5
	3	R	9	6	0	5
	4	S	4.5	7.5	5	0

Table 3.2 Population size at villages

Village	1	2	3	4
	P	Q	R	S
Population(h)	91	116	84	74

$$w_i = \sum_{i,j}^{n} h_i d_{ij} y_{ij}$$

Where w_1, w_2, w_3 and w_4 are the total weighted distances for the four villages

d_{ij} = The distance between village i and village j

h_i = The population of village i

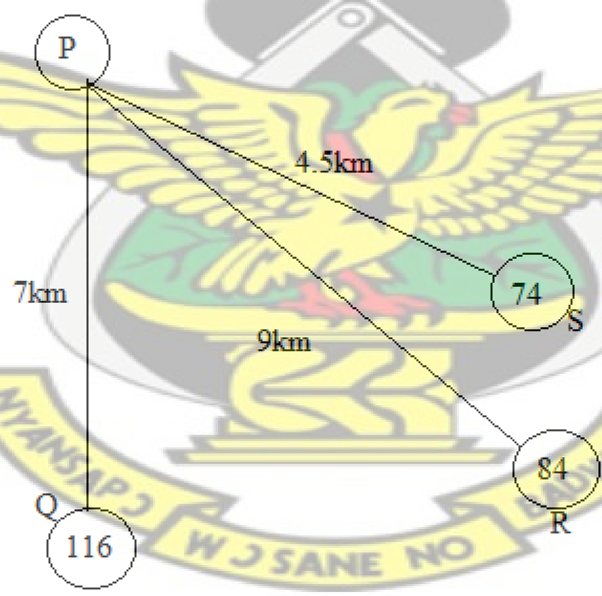
n = Number of sites(villages)

w_1, w_2, w_3 and w_4 are calculated and the minimum of them is selected to locate the facility.

The distance between a village and itself is zero, that is, the distance from P to P is equal to zero.

For location P the total weighted distance (w_1) of is calculated as follows:

Fig 3.27



$$w_i = \sum_{i,j}^{n} h_i d_{ij} y_{ij}$$

Where $h_2 = 116$, $h_3 = 84$, $h_4 = 74$, and $d_{12} = 7$ $d_{13} = 9$ $d_{14} = 4.5$

$\sum_i Y_{ij} = X_j = Y_{ij} = n = 1$ because each location will be served by all other locations

$w_1 = \sum_{ij}^{n} h_i d_{ij} y_{ij} \qquad i = 2,3,4 \quad j = 2,3,4$

$w_1 = d_{12}h_2 + d_{13}h_3 + d_{14}h_4$

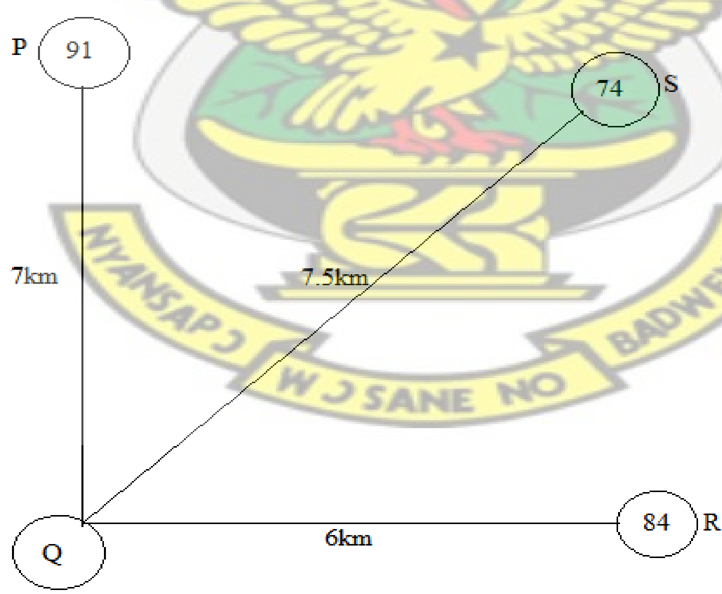
$w_1 = (7 \times 116) + (9 \times 84) + (4.5 \times 74)$

$w_1 = 1901km$

This implies that for all the people in other villages to have access to the facility at P, a total distance of $1901km$ must be covered.

For location Q the total weighted distance (w_2) is calculated as follows

Fig 3.28



$w_i = \sum_{ij}^{n} h_i d_{ij} y_{ij}$, where $h_1 = 91, h_3 = 84, h_4 = 74$, and $d_{21} = 7, d_{23} = 6, d_{24} = 7.5$

$w_2 = \sum_{ij}^{n} h_i d_{ij} y_{ij} \quad i = 2,3,4 \;\; j = 2,3,4$

$w_2 = d_{21}h_1 + d_{23}h_3 + d_{24}h_4$

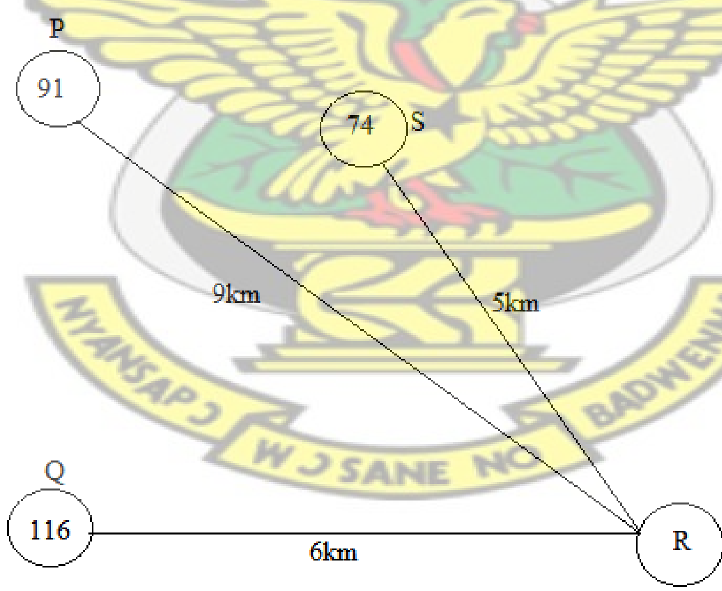
$w_2 = (7 \times 91) + (6 \times 84) + (7.5 \times 74)$

$w_2 = 1696km$

This implies that for all the people in other villages to have access to the facility at Q, a total distance of $1696km$ must be covered.

For location R the total weighted distance (w_3) is calculated as follows:

Fig 3.29



$w_i = \sum_{ij}^{n} h_i d_{ij} y_{ij}$, where $h_1 = 91, h_2 = 116, h_4 = 74$, and $d_{31} = 9, d_{32} = 6, d_{34} = 5$

$w_3 = \sum_{ij}^{n} h_i d_{ij} y_{ij} \quad i = 2,3,4 \;\; j = 2,3,4$

$w_3 = d_{31}h_1 + d_{32}h_2 + d_{34}h_4$

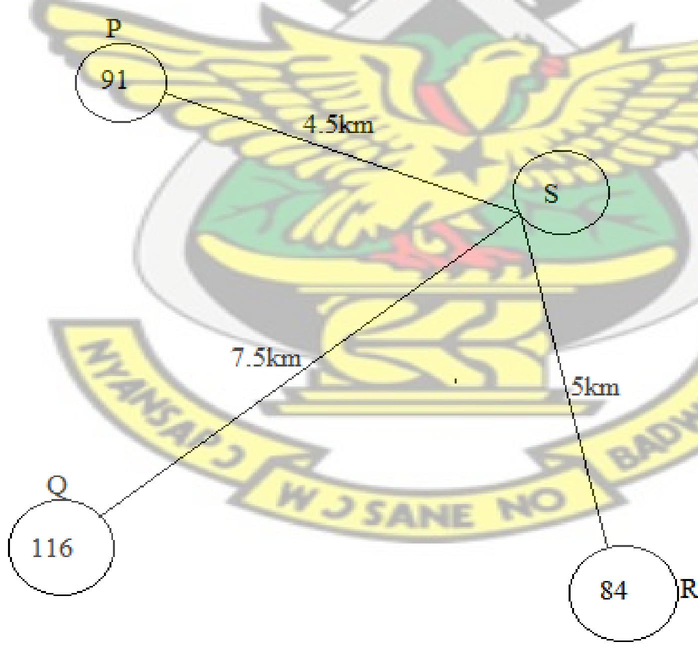
$w_3 = (9 \times 91) + (6 \times 116) + (5 \times 74)$

$w_3 = 1885km$

This implies that for all the people in other villages to have access to the facility at R, a total distance of $1885km$ must be covered.

For location S the total weighted distance (w_4) is calculated as follows:

Fig 3.30



$w_i = \sum_{ij}^{n} h_i d_{ij} y_{ij}$, where $h_1 = 91, h_2 = 116, h_3 = 74,$ and $d_{41} = 4.5, d_{42} = 7.5, d_{43} = 5$

$w_4 = \sum_{ij}^{n} h_i d_{ij} y_{ij} \quad i = 2,3,4 \quad j = 2,3,4$

$w_4 = d_{41}h_1 + d_{42}h_2 + d_{43}h_3$

$w_4 = (4.5 \times 91) + (7.5 \times 116) + (5 \times 84)$

$w_4 = 1699.5km$

This implies that for all the people in other villages to have access to the facility at S, a total distance of $1699.5km$ must be covered.

Village Q is the best location for the facility because the minimum among the calculated weighted distances is $1696km$ and it corresponds to location Q.

CHAPTER FOUR

4.0 DATA COLLECTION AND ANALYSIS

In this chapter, the Prim's algorithm will be used in a minimum connector problem to find a tree and then the P-median method will be used to locate a facility on a node on the tree.

The minimum connector problem and siting of electrical facilities such as substations is a real practical problem in the electrical cable layout construction process.

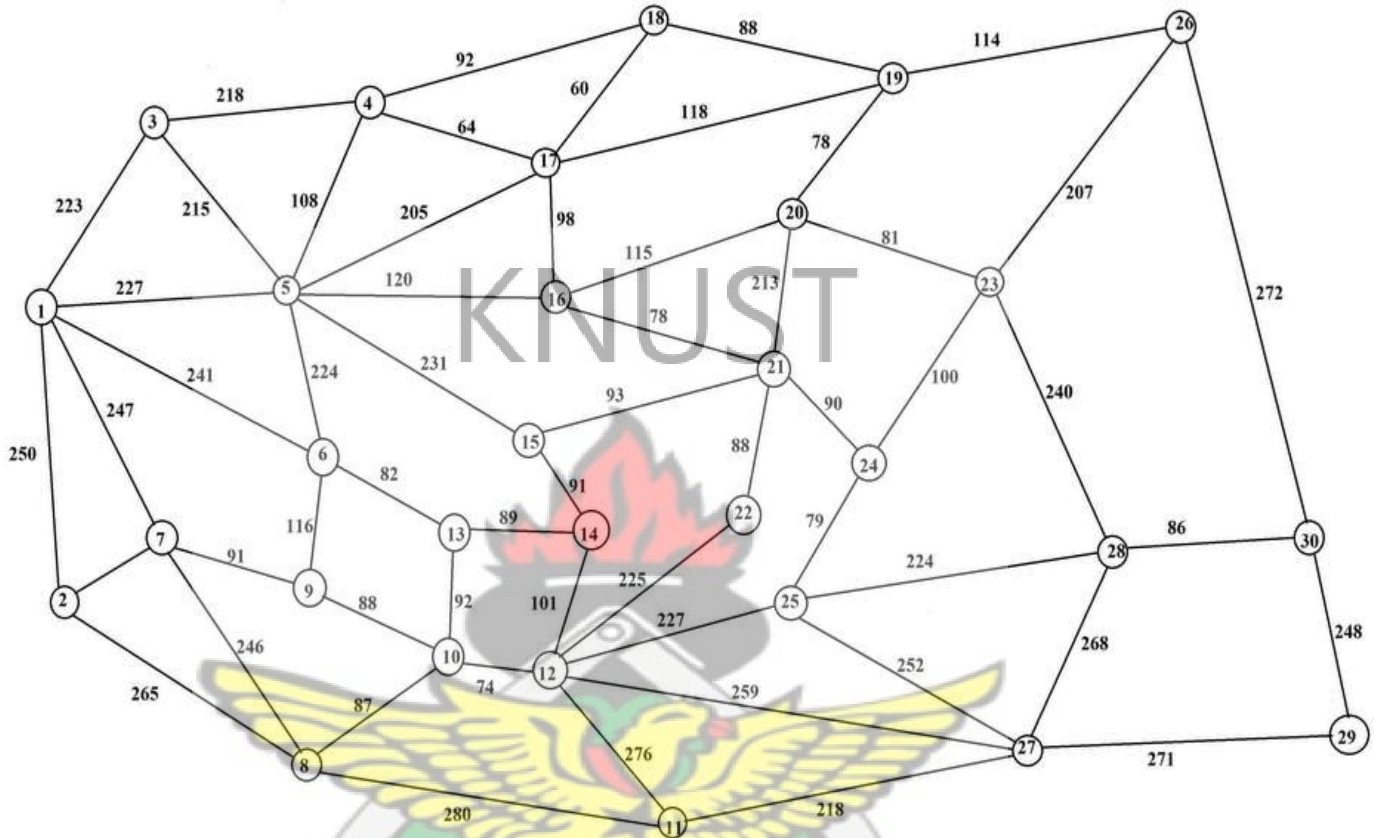
The aim is to locate an electrical substation on a tree by first, determining the minimum distance linking all nodes on a network to form a tree so as to reduce cost and delivery time to all these points.

Data for the research was obtained from two main sources; Town and Country Planning Department and The Statistical Services (Sunyani Metropolitan Assembly)

The data consist of network of roads and thirty junctions (nodes) with their corresponding interconnected distances in metres and the population of people currently residing around the various nodes at the Kootokrom township in the Sunyani municipality.

The data is illustrated as a network problem in figure 4.1

Fig 4.1 Network of roads and junctions(nodes) in Kootokrom



The network consisting of thirty (30) nodes indicating the distances from each node to every other node is put in a 30x30 matrix as shown below in Table 4.1. The full matrix is shown at appendix A

The matrix will be run in a Matlab code to find a tree which is also the minimum connector for the cable layout in the Kootokrom Township.

The populations of nodes are indicated in Table 4.2

Table 4.1Distances from all nodes to every other node

$$
\begin{pmatrix}
0 & 250 & 223 & \dots & \dots & \dots & \dots & \propto & \propto & \propto \\
250 & 0 & \propto & \dots & \dots & \dots & \dots & \propto & \propto & \propto \\
223 & \propto & 0 & \dots & \dots & \dots & \dots & \propto & \propto & \propto \\
\vdots & \vdots & \vdots & \ddots & \dots & \dots & \dots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \dots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \dots & \ddots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \dots & \dots & \dots & \ddots & \vdots & \vdots & \vdots \\
\propto & \propto & \propto & \dots & \dots & \dots & \dots & 0 & \propto & \propto \\
\propto & \propto & \propto & \dots & \dots & \dots & \dots & \propto & 0 & \propto \\
\propto & \propto & \propto & \dots & \dots & \dots & \dots & \propto & \propto & 0
\end{pmatrix}
$$

Table 4.2 Population size around nodes(junctions)

NODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
POPULATION	35	36	38	50	100	115	80	60	110	90	40	70	98	108	120

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
112	79	45	50	89	119	98	87	95	110	31	40	52	30	32

4.1 COMPUTATIONAL PROCEDURE

The distances between nodes with direct and indirect connections were written in the form of matrix. Matlab program software was used for coding the Prim’s algorithm.

Again, Matlab program software was used for coding the P-median method.

The matrices were inputted in the Matlab program code and ran on Satellite L655 Intel Pentium(R) Dual-Core processor T4500,4GB RAM, 64-bit operating system, 2.30GHz speed, Windows 7 Toshiba laptop computer. The code ran successfully on six trials.

4.1.1 PRIM'S ALGORITHM STEPS

- Select any vertex
- Select the shortest edge connected to that vertex
- Select the shortest edge connected to any vertex already connected
- Repeat step 3 until all vertices have been connected

4.1.2 RESULTS

After the 30x30 matrix was run using the Matlab code for Prims Algorithm, edges in minimum spanning tree and their cost were displayed after twenty nine (29) iterations as shown in table 4.2

Table 4.3 Results of Matlab code for Prim's algorithm

EDGES	1,3	3,5	5,4	4,17	17,18	18,19	19,20	20,23
COST	223	215	108	64	60	88	78	81

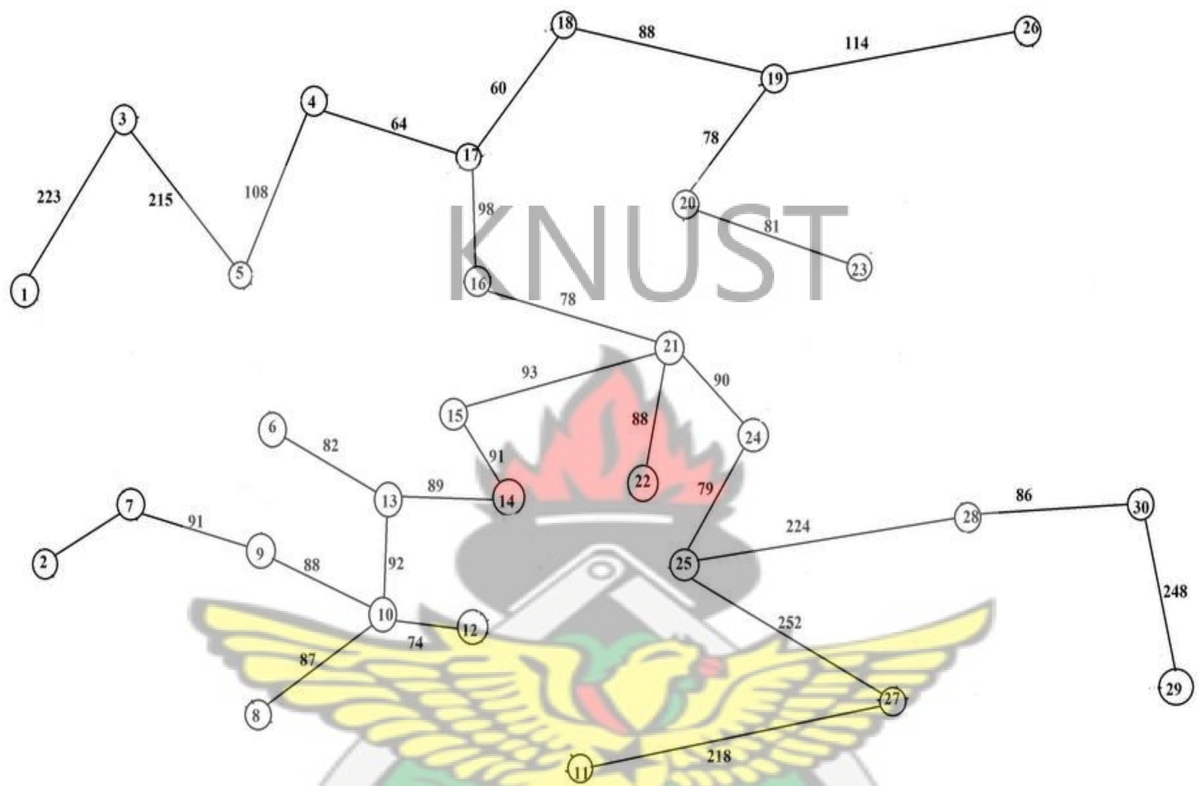
17,16	16,21	21,22	21,24	24,25	21,15	15,14	14,13	13,6	13,10	10,12
98	78	88	90	79	93	91	89	82	92	74

10,8	10,9	9,7	7,2	19,26	25,28	28,30	30,29	25,27	27,11
87	88	91	88	114	224	86	248	252	218

The above table shows nodes connected on the tree and their respective distances.

The minimum connector was indicated as 3367m.

Fig 4.2 Minimum connector of the network of roads and junctions in Kootokrom



4.1.3 FORMULATION FOR P-MEDIAN

The problem involves placing p facilities so that the total distance to one of those facilities is minimized. The model can represented mathematically as follows.

$$Minimize \quad \sum \sum_{i}^{n} h_i d_{ij} y_{ij} \qquad (1)$$

$$Subject\ to: \sum_i X_i = n \qquad (2)$$

$$\sum_i Y_{ij} = 1 \qquad (3)$$

$$X_i = \{0,1\} \qquad (4)$$

$$Y_{ij} = \{0,1\} \qquad (5)$$

Where w_i = weighed distance for site i

i = index of selected site

j = index of site for potential facility placements

n = number of site(s) to locate facility

h_i = demand at node i

d_y= distance between node I and node

x_i = 1,0 where 1 implies a potential facility is located at site j and 0 implies no facility is located at site j

Y_{ij} = 1,0 where 1 implies site i is served by a facility at site j and 0 implies site i is not served by a facility at site j

The objective function (1) seeks to minimize the total distance covered by other sites to access a facility at a selected site. Constraint (2) limits the number of facilities to be placed to some number *n*. Constraint (3) assures that each node i is served by one facility. Constraints (4) and (5) define the decision variables X and Y.

4.1.4 ALGORITHM STEPS FOR P-MEDIAN

- Choose a starting node
- Join this node to the next node, not already in the solution

- Multiply the distance of the next node by the population of the starting node
- Repeat with all other nodes until all nodes have been included
- Sum all products for each node
- Select the node with the minimum value as best site for location of facility

4.1.5 RESULTS

Table 4.4 indicates the results of thirty iterations, after the 30x30 matrix and the populations were run using Matlab code for p-median

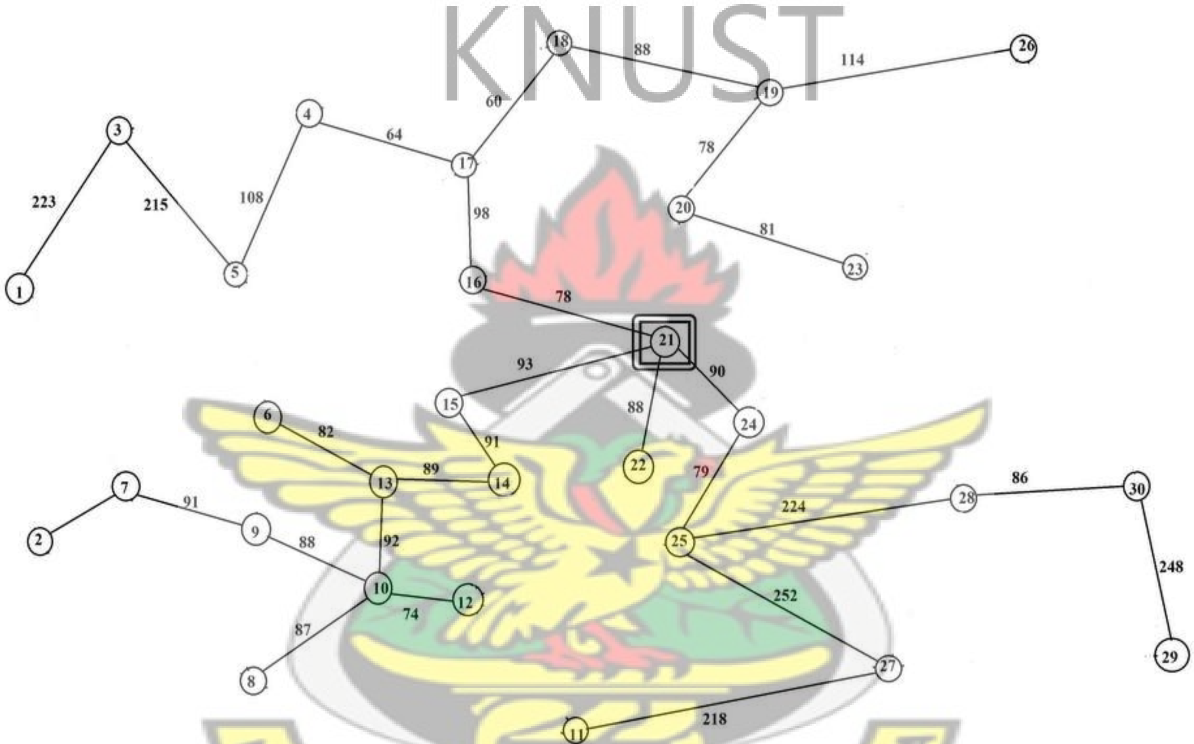
Table 4.4 P-median values of the thirty nodes (junctions) in Kootokrom

NODE	P-M VALUE		NODE	P-M VALUE
1	1,902,649		16	1,759,259
2	2,039,690		17	1,747,422
3	2,036,205		18	2,008,740
4	1,935,280		19	1,948,758
5	1,674,630		20	1,741,180
6	1,812,631		21	1,642,027
7	1,934,583		22	1,958,222
8	1,961,250		23	1,900,606
9	1,852,540		24	1,836,100
10	1,820,096		25	1,897,123
11	2,053,420		26	2,051,413
12	1,748,748		27	1,953,636
13	1,835,322		28	1,956,992
14	1,849,712		29	2,135,776

15	1,815,995		30	2,094,344

The above table shows the nodes and their various p-median values and the best location indicated as node 21 with a minimum value of 1,642,027metres.

Fig 4.3 The best site to locate an electric substation



4.2 DISCUSSION

The minimum connector is the sum of distances on all connected node, Hence

Minimum Connector for the cable network = 223 + 215 + 108 + 64 + 60 + 88 + 78 + 81 + 98 + 78 + 88+ 90+ 79 + 93 + 91 + 89 + 82 + 92 + 74 + 87 + 88 + 91 + 88 + 114 + 224 + 86 + 248 + 252 + 218 = 3367m

The connected nodes also forms a tree as indicated in figure 4.2

The study has clearly established that it will take a minimum length of 3367m of electrical cable to connect all thirty (30) nodes in the Kootokrom Township

It is also apparent at this point to say that the best location to site an electrical substation will be at node 21 since it has the minimum p-median value of 1,642,027m.

CHAPTER FIVE

5.0 CONCLUSION

The problem of distributing cables was modeled as a network problem and the Prim's Algorithm was used to find a tree and the minimum connector of electrical cables.

On the tree, the p-median method was used to locate an electrical substation on one of the thirty nodes.

Prim's Algorithm showed an optimal cable layout length of 3367m and P-median method indicated node 21 as the best location to situate an electrical substation in the Kootokrom Township with a minimum weighted distance of 1,642,027m

5.1 RECOMMENDATIONS

Based on the study, the following recommendations are made.

Prim's algorithm gives a good reduction in distance and I recommend it to stakeholders in the electrical cable layout construction industry as it would help reduce cost and time involved in laying cables.

Similarly, p-median method provides better location sites for facilities as important as electrical substations, ambulance stations, fire vehicle stations, ICT centers etc

It's my wish that this work serves as basis for further studies in the field of choosing a site to locate a facility, and in the quest of optimality in any cable layout plan as well.

Finally, since Kootokrom Township plan was used as case study, the researcher recommends that GridCo Ghana Ltd adopt the project code to help determine the minimum connector for cables in their distribution process and also site appropriately, facilities such as their substations in order to reduce the operational cost and hence the rather high cost of electricity in Ghana.

REFERENCES

1. Schijver, A.(1996) On the History of Combinatorial Optimization. pp 27 CWI and University of Amsterdam, Amsterdam.

2. Prim R.C., (1957) Shortest connection networks and some generalizations. In: Bell System technical Journal, pp.1389-1401.

3. Jayawant, P. and Glavin,(2009). Minimum spanning trees. Involve, 2(4): 437-448.

4. Agarwal, D., (2010). Kruskal Algorithm. Technical Collection. Retrieved from: http://sites.google.com/site/mytechnicalcollection/algorithm/graphs/minimum-spanning-trees/kruskal-s-algorithm. (Accessed on December 14,2011)

5. Zachariasen, M.(1998). Algorithm for plane Steiner Tree Problems. Ph.D. Thesis, Department of Computer Science, University of Copenhagen, Copenhagen.

6. Ivanov, A. O. and A. A. Tuzhilin, (1994). Minimal Networks: The Seiner Problem and its Generalizations. CRC Press, Florida.

7. Gilbert, E.N. and H.O. Pollack, (1968). Steiner Minimal Trees. J. Soc. Indus. Math., 16(1): 1-29.

8. Dott, D.R., (1997). Optimal Network Design for Natural Gas Pipelines. University of Calgary, Alberta, pp: 57-90.

9. Brimberg et al,(2003). Applying Tree Knapsack approaches to general network design: a case study. North-West University Institutional, Potchefstroom Campus.

10. Arogundade, O. T. and A.T. Akinwale,(2009) On the Application of Prim's Algorithm to a Profit-Oriented Transportation System in Rural Areas of Nigeria. University of Agriculture, Abeokuta, 37: 4-9. Retrieved from: http://www.google.com.gh

11. Xu,et al; Guanhui G; Min S; Suping L (2008) Pipe Network 3D Visualization Service Architecture, Beijing Univ. of Posts & Telecommunication, Beijing

12. Barab A. L (2002) Linked: The New Science of Networks. Perseus Publishing, Li,

13. Mouli, V.V.S.C; Kumari, S. M; Geethanjali, N (2010) Shortest Path Algorithms in Transportation Networks. Global Journal of Computer Science and Technology Press, Sri Krishna Devaraya University ,Anantapur,A.P,India

14. Gonina, E. and Kale, L.(2007) Parallel Prim's algorithm on Dense Graphs with Novel Extension.

15. Moore, E.F. (1959) The shortest path through a maze. Cambridge: Harvard University Press.

16. S.L. Hakimi, Optimum locations of switching centers and the absolute centers and medians of a graph. Opns. Res. 11, 450-459 (1964).

17. Daniel Seira and Vladimir Marianov,July 23 (1996), Page 1. The P-median problem in changing network; the case of Barcelona.

18. Cooper, L. (1963). Location–Allocation Problems, *Operations Research* 11, 331–343

19. Revelle C S, Swain R W (1970),Central Fascilities Location, Geographical Analysis 2, 30-42

20. Maranzana F,(1964) On the Location of Supply Points to Minimize Transport Costs,*Operations Research Quarterly*, 15, 261-270

21. Teitz M and Bart P, (1968) Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph, *Operations Research*, 16, 955-961.

22. Morris, J., (1978) "On the extent to which certain fixed-charge depot location problems can be solved by LP." *Journal of the Operational Research Society*, 29, 71-76.

23. Rosing, K., ReVelle C., and H. Rosing-Vogelaar, (1979) "The p-median model and its linear programming relaxation: an approach to large problems", *Journal of the Operational Research Society* 30(9), 815-823.

APPENDICES

APPENDIX A

30x30 matrix that was run in a Matlab code for Prims Algorithm

0	250	223	1000	227	241	247	1000	1000	1000	1000	1000	1000	1000	1000
250	0	1000	1000	1000	1000	88	265	1000	1000	1000	1000	1000	1000	1000
223	1000	0	218	215	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
1000	1000	218	0	108	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
227	1000	215	108	0	224	1000	1000	1000	1000	1000	1000	1000	1000	231
241	1000	1000	1000	224	0	1000	1000	116	1000	1000	1000	82	1000	1000
247	88	1000	1000	1000	1000	0	246	91	1000	1000	1000	1000	1000	1000
1000	265	1000	1000	1000	1000	246	0	1000	87	280	1000	1000	1000	1000
1000	1000	1000	1000	1000	116	91	1000	0	88	1000	1000	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	87	88	0	1000	74	92	1000	1000
1000	1000	1000	1000	1000	1000	1000	280	1000	1000	0	270	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	74	270	0	1000	101	1000
1000	1000	1000	1000	1000	82	1000	1000	1000	92	1000	1000	0	89	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	101	89	0	91
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	91	0
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	93
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	225	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	227	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	259	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000

$$
\begin{pmatrix}
1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
1000 & 64 & 92 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
120 & 205 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 218 & 1000 & 1000 & 1000 \\
1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 225 & 1000 & 1000 & 227 & 1000 & 259 & 1000 & 1000 & 1000 \\
1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
1000 & 1000 & 1000 & 1000 & 1000 & 93 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
0 & 98 & 1000 & 1000 & 115 & 78 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
98 & 0 & 60 & 118 & 114 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
1000 & 60 & 0 & 88 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
1000 & 118 & 88 & 0 & 78 & 1000 & 1000 & 1000 & 1000 & 1000 & 114 & 1000 & 1000 & 1000 & 1000 \\
115 & 114 & 1000 & 78 & 0 & 213 & 1000 & 81 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
78 & 1000 & 1000 & 1000 & 213 & 0 & 88 & 1000 & 90 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
1000 & 1000 & 1000 & 1000 & 1000 & 88 & 0 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 \\
1000 & 1000 & 1000 & 1000 & 81 & 1000 & 1000 & 0 & 100 & 1000 & 207 & 1000 & 240 & 1000 & 1000 \\
1000 & 1000 & 1000 & 1000 & 1000 & 90 & 1000 & 100 & 0 & 79 & 1000 & 1000 & 1000 & 1000 & 1000 \\
1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 79 & 0 & 1000 & 252 & 224 & 1000 & 1000 \\
1000 & 1000 & 1000 & 114 & 1000 & 1000 & 1000 & 207 & 1000 & 1000 & 0 & 1000 & 1000 & 1000 & 272 \\
1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 252 & 1000 & 0 & 268 & 271 & 1000 \\
1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 240 & 1000 & 224 & 1000 & 268 & 0 & 1000 & 86 \\
1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 271 & 1000 & 0 & 248 \\
1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 1000 & 272 & 1000 & 86 & 248 & 0
\end{pmatrix}
$$

The distances valued 1000m in the matrix above are assumed to be very large compared to all other distances having direct connections to nodes and hence have no direct connection with the node in question.

APPENDIX B

Matlab code for Prim's algorithm

```
function [mst, cost] = prim(A)

% User supplies adjacency matrix A.  This program uses Prim's algorithm
% to find a minimum spanning tree.  The edges of the minimum spanning
% tree are returned in array mst (of size n-1 by 2), and the total cost
% is returned in variable cost.  The program prints out intermediate
% results and pauses so that user can see what is happening.  To continue
% after a pause, hit any key.
A=input('Enter A:')
[n,n] = size(A);                       % The matrix is n by n, where n = # nodes.
A, n, pause,
if norm(A-A','fro') ~= 0 ,             % If adjacency matrix is not symmetric,
  disp(' Error:  Adjacency matrix must be symmetric ') % print error message and quit.
  return,
end;
% Start with node 1 and keep track of which nodes are in tree and which are not.
intree = [1];  number_in_tree = 1;  number_of_edges = 0;
notintree = [2:n]';  number_notin_tree = n-1;
in = intree(1:number_in_tree),         % Print which nodes are in tree and which
out = notintree(1:number_notin_tree), pause, % are not.
% Iterate until all n nodes are in tree.
while number_in_tree < n,
```

```
% Find the cheapest edge from a node that is in tree to one that is not.
mincost = Inf;                % You can actually enter infinity into Matlab.
for i=1:number_in_tree,
  for j=1:number_notin_tree,
    ii = intree(i);  jj = notintree(j);
    if A(ii,jj) < mincost,
      mincost = A(ii,jj); jsave = j; iisave = ii; jjsave = jj;  % Save coords of node.
    end;
  end;
end;
% Add this edge and associated node jjsave to tree.  Delete node jsave from list
% of those not in tree.
number_of_edges = number_of_edges + 1;     % Increment number of edges in tree.
mst(number_of_edges,1) = iisave;          % Add this edge to tree.
mst(number_of_edges,2) = jjsave;
costs(number_of_edges,1) = mincost;
number_in_tree = number_in_tree + 1;       % Increment number of nodes that tree connects.
intree = [intree; jjsave];                % Add this node to tree.
for j=jsave+1:number_notin_tree,           % Delete this node from list of those not in tree.
  notintree(j-1) = notintree(j);
end;
number_notin_tree = number_notin_tree - 1;  % Decrement number of nodes not in tree.
in = intree(1:number_in_tree),             % Print which nodes are now in tree and
```

```
  out = notintree(1:number_notin_tree), pause,% which are not.
end;
disp(' Edges in minimum spanning tree and their costs: ')
[mst  costs]                          % Print out edges in minimum spanning tree.
cost = sum(costs)
```

APPENDIX C

Matlab code results for Prim's algorithm

Edges in minimum spanning tree and their costs:

Ans	=	1	3	223
		3	5	215
		5	4	108
		4	17	64
		17	18	60
		18	19	88
		19	20	78
		20	23	81
		17	16	98
		16	21	78
		21	22	88
		21	24	90
		24	25	79
		21	15	93
		15	14	91

14	13	89
13	6	82
13	10	92
10	12	74
10	8	87
10	9	88
9	7	91
7	2	88
19	26	114
25	28	224
28	30	86
30	29	248
25	27	252
27	11	218

Cost = 3367

APPENDIX D

Matlab code for p-median method

%h is 1 x n matrix indicating the population of the suburbs

%d is n x n matrix indicating the distances between suburbs

```
d =input('Enter d;')
h=input('Enter h;')
mn=size(d)
m=mn(1)
n=mn(2)
i=1:n
j=1:n
w=(h*d)'
[p_median_value,location]=min(w)
```

APPENDIX E

Matlab code results for the p-median method

```
mn =

    30    30

m =

    30

n =

    30

i =

  Columns 1 through 13

     1     2     3     4     5     6     7     8     9    10    11    12    13

  Columns 14 through 26

    14    15    16    17    18    19    20    21    22    23    24    25    26

  Columns 27 through 30

    27    28    29    30

j =

  Columns 1 through 13

     1     2     3     4     5     6     7     8     9    10    11    12    13

  Columns 14 through 26

    14    15    16    17    18    19    20    21    22    23    24    25    26

  Columns 27 through 30

    27    28    29    30
```

w =

1902649

2039690

2036205

1935280

1674630

1812631

1934583

1961250

1852540

1820096

2053420

1748748

1835322

1849712

1815995

1759259

1747422

2008740

1948758

1741180

1642027

1958222

1900606

1836100

1897123

2051413

1953636

1956992

2135776

2094344

p_median_value = 1642027

location = 21