KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

INSTITUTE OF DISTANCE LEARNING

PROGRAMME: MSC. INDUSTRIAL MATHEMATICS

TOPIC

OPTIMIZING THE LAYOUT OF PIPE NETWORK SYSTEM OF ASUAKWA - SUNYANI FROM THE MAIN SOURCE, ABESIM IN THE BRONG AHAFO REGION

BY

OWUSU-KYEREMEH MARTIN

(BED MATHEMATICS)

SUPERVISOR: MR. F. K. DARKWAH

A THESIS SUBMITTED TO INSTITUTE OF DISTANCE LEARNING TO DEPARTMENT OF MATHEMATICS OF KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY IN PARTIAL FULFILLMENT OF THE AWARD OF MASTER OF SCIENCE INDUSTRIAL MATHEMATICS

NOVEMBER, 2013

DECLARATION

I hereby declare that this submission is my own work towards the MSc 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 other degree of the University, except where due acknowledgment has been made in the text.

Owusu-Kyeremeh Martin	……………………..	…………………..
PG4070110	Signature	Date

Certified by:

F. K. Darkwah	……………………..	…………………..
Supervisor	Signature	Date

Certified by:

Prof. S.K. Amponsah	…………………..	…………………..
Head of Department	Signature	Date

Prof. I.K. Dontwi	…………………..	…………………..
Dean, IDL	Signature	Date

ABSTRACT

Network design phase is done before usually laying the network infrastructure.

Hence it is very important that the design achieves optimality and reliability.

As the water supply industry matures, there is vital interest in good pipe network design. The study is to access some of the causes of water supply problems faced by the Sunyani Community.

To find the optimal solution to the pipe network system for the supply path for efficient supply, a solution-oriented modeling approach can help to devise efficient automatic planning and optimization schemes for Sunyani municipal. This thesis discuses how good mathematical models can be obtain for real-world problems using prim's algorithm, which model structures to minimize pipe network systems and which can help in easy and less expensive water supply especially in the Asuakwaa in Sunyani from Abesim thus Tano river.

The prim's algorithm employed to find the optimal solution was able to reduce the total pipe network system of about 30% of the original network. After having briefly explained the main aspect of network planning such as minimum connection, reliability, network configuration and so on, it characterizes the difference situation where basic algorithms of graph theory could be reused.

ACKNOWLEDGMENT

I wish first and foremost give praises and thanks to God Almighty for his divine protection, direction and help throughout the course of my education.

I wish to express my profound thanks to Mr. F. K. Darkwah, my supervisor, and head of Department of Mathematics, K.N.U.S.T., for the excellent guidance, for making valuable suggestions for the improvement of the work, for his constant interest, constructive criticism and support he provided during the planning and carrying out of this project.

My sincere thanks also go to the entire staff of the Department of Mathematics, K.N.U.S.T., for their kind advice and encouragement.

I am grateful to Mr. Godwin Osei Awuah, Town and Country Planning, Sunyani Municipal Assembly for kindly making available to me all necessary data needed for my project. I am also indebted to Mr. Martin Martey Kpabitey, Mr. Anthony Donkor and Florence Sefa Dankwah.

DEDICATION

I dedicate this thesis with all my love and respect to my dear wife, Florence Sefa Dankwah and my dear sons, Nana Owusu-Dukuh kyeremeh and Nhyera Sefa Kyeremeh not forgetting my mother Madam Georgina Yeboah.

TABLE OF CONTENT

Content **Page**

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

Water, undoubtedly is a universal Commodity and very important for use for the survival of mankind. Its use spans from domestic, industrial, agriculture to power generation. The use and important of water is felt most when our taps cease to flow and our rivers run dry. We suddenly realize that water indeed is life. Sometimes, this tells us why people struggle to their last breath to get even the worse water –from canals and ponds. There is therefore the need to use every good method and means available to solve issues relating to how water gets to where ever it is needed for usage by mankind.

1.0 Background to Study

The world water day is celebrated annually on the 22nd of March. It's main purpose is to address issues relating to water resources, their management and the supply of portable water. In Ghana, portable water coverage is very low –about 45% for rural areas and 70% for urban centers.

However, the Millennium Development Goals (MDG) aim at reducing the number of people without portable drinking water by 2015. It is difficult for one to live without water. The uses of water are many, from drinking and cleaning to irrigation crops and landscapes. Water is use for cooking, for recreation and dust control. Water resources are sources of water that is useful or potentially useful. Uses of water include agricultural, industrial, household recreational and environmental activities. Virtually all of these human uses require fresh water. 97% of the water on the earth is salt water and only 3% is fresh water.. Fresh water is renewable resources, yet the world's supply of clean, fresh water in steadily decreasing .water demand already exceeds supply in many parts of the world and as the world population continues to rise, so too does the water demand.

The MST solution although optimal, may be highly vulnerable to failure due to our reliance on a few vertices. Hence it is sometimes necessary to employ shortest path (SP) on sewage Network Company to reduce cost. The SP is important in the design of sewage network, energy network Transportation, Telecommunication network and plumbing.

This study intends to find the optimal network system to deal with water stress in Ghana especially some part of Sunyani municipality. The study provides policy framework to the stakeholders of Ghana water Company Limited on how best pipe network can be laid to reduce water supply problems and cost.

1.1 Mathematical Theories on Shortest Path Algorithms

Theory graph in mathematics network in the digraph with weighted edges. These networks have become useful concepts for analyzing the interaction between science and mathematics; electricity and mathematics, communication and mathematics and others. Using networks of all types various application based on creativity of the mathematician along with their environment can be evaluated in all sort of manners. Some may visualize networks in various contexts to feel the networks which the nodes belong. Creating an environment for the nodes to belong is an essential to the mathematical evaluation and furthermore the mathematician belonging to the environment just as the networks nodes (Prim, 1957)

The history of the minimum cost spanning tree problem is rather interesting and complex until recently, most textbooks discussions of this problem made reference to the work of two American mathematician then the employees at Bell laboratories who each developed an algorithm for solving the minimum cost spanning tree problem. They are Joseph B Kruskal and Robert K Prim.

The researchers considered few algorithms that can enable the researcher to find a possible way of minimizing pipe network system. Some of these algorithms are as follows;

Prim's algorithm is an algorithm in graph theory that finds a minimum spanning tree for a connected weighted graph. The algorithm 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 will only find a minimum spanning tree for one of the connected components. The algorithm was discovered in 1930 by mathematician Vojtech Jarnik and later independently by computer scientist Robert Prim and rediscovered by Dijkstra. Therefore it is sometimes called DJP algorithm or Jarnik algorithm.

It works as follows;

- Create a tree containing a single vertex, chosen arbitrarily from the graph
- Create a set containing all the edges in the graph
- Loop until every edge the set connects two vertices in the tree
    i. Remove from the set and edge with minimum weight that connects a vertex in the tree with the vertex not in the tree

ii. Add that edge to the tree

Like Kruskal's algorithm, Prim's algorithm is based on a generic Minimum Spanning Tree (MST) algorithm. The idea of Prim's algorithm is similar to that of Djkstra's algorithm for finding shortest path in a given graph. Prim's algorithm has a property that the edges of set A always form a single tree. We begin with some vertex V in a given graph G= (V, E), defining the initial set of vertices A. Then, in each iteration we choose a minimum weight edge (U, V) connecting a vertex V in a set A to the vertex U outside the set A. the vertex U is brought into A. the process is repeated until the spanning tree is formed.

Dijkstra's algorithm is one of the most popular algorithms in computer science. It is also popular in operations research. It has many attractions for both lectures and students. Unfortunately however, a number of important features of this fascinating algorithm are not as transparent to the average user as they should be. In particular of the consequences of the history of the algorithm is that is generally regarded as a computer science method rather than as an operation reset method.

In graph theory the shortest path problem is the problem of finding the path between two vertices 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 location and the edges represent segments of road and are weighted by the time needed to travel that segment

1.2 History and recent development of Ghana Water Company Limited

The first public water supply system in Ghana then Gold Coast was established in Accra just before World War 1. Extensions were made exclusively to other urban areas among them the colonial capital of Cape Coast, Winneba and Kumasi.

During this period the water supply system were managed by the hydraulic division of public works Department. With time the responsibilities of the hydraulic division were widened to include the planning and development of water supply system in other parts of the country.

The Department of rural water development was established to engage in the development and management of rural water supply through the drilling of bore hole and construction of wells for rural communities

After Ghana's independence, a water supply division with headquarters in Kumasi was set up under the ministry of work and housing with responsibilities for both urban and rural water supplies.

During the dry season in, there was severe water shortage in the country. Following this crisis an agreement was signed between the Government of Ghana and the World Health Organization (WHO) for a study to conduct into the water sector development of the country.

The study focused not only on technical engineering but also on the organization of a national water and sewerage authority and method of financing. Furthermore the study recommended the preparation of master plan for water supply and sewerage service in Accra-Tema covering the twenty –year period. (Ghana Water Company Limited website).

In line with the recommendations of the **WHO**, the Ghana water and sewerage corporation (GWSC) was to be responsible for:

- Water supply and sanitation in rural as well as urban areas.
- The conduct of research on water and sewerage as well as making of engineering surveys and plans.
- The construction and operation of water sewerage works.
- The setting of standards and prices and collection of revenues

Various reforms have been introduced to address the problems of the sector. The key objectives of the reforms were to separate rural and urban service, to introduce independent regulatory agencies and to promote private sector participation.

In order to pay more attention to water supply and sanitation in rural areas, the Community Water and Sanitation Division was founded as semi-autonomous division of (GWSC). Four years later it name changed to Community water and Sanitation Agency (CWSA) and became fully independent. The GWSC was replaced by the publicly owned GWCL. At the same time the responsibility for rural water supply and sanitation was decentralized to the District Assemblies. In addition, sanitation was separated from water supply and became responsibility of the District Assemblies in urban and rural areas.

As a result, the GWCL remained responsible only for urban water supply, whereas more than 110 small towns' water systems were transferred to District Assemblies, which received support from the CWSA. In terms of sanitation, District Assemblies are responsible in urban and rural areas. In the later case, a demand-driven and community-managed approach was introduced.

The regulation of water supply has been shifted from the government to independent agencies. Two commissions were created to regulate the sector: the Public Utilities Regulatory Commission (PURC) has been developed to formulate and approve appropriate pricing mechanisms aimed at full cost recovery, since the government began to phase out the subsidization of water services. The PURC has no authority over community-managed water system and only regulates GWCL services. Besides the provision of tariffs guidelines and the examination and approval of tariffs, it protects the interests of the consumers and providers, promotes fair competition, and initiates conducts and monitors standard s concerning the provided services.

Whereas the PURC takes responsibility for economic regulation of urban water supply and sanitation, the water Resources Commission (WRC) regulates water resources, it is in charge of licensing water abstraction and waste discharges. To carry out the private sector participation of GWCL, originally a 10-year lease contract was envisaged. In 2000, a lease contract between GWCL and the US Company Azurix failed due to public opposition and accusation of corruption which led to the formation of the Coalition against water privatization.

Under the framework of the Urban Water Project, a five –year management contract was signed between the GWCL and AVRL. The main objectives of this private sector participation are:

- Extending reliable water supply especially to low –income areas
- Making portable water affordable for low- income consumers
- Increasing cost recovery
- Ensuring investment based on low- cost and concession financing
- Supporting further involvement of the private sector
- Reducing non- revenue water
- Increasing water treatment

The project is financed by the World Bank, the Nordic Development Fund and the Republic of Ghana (Wonder, 2007). To overcome the lack of coordination between the numerous institutions which are created, Boniface Abubakar Saddique, the minister of Water Resources, Works and Housing lunched a National Water Policy (NWP) which focuses on the three strategic areas:

I. Water resources management

II. Urban water supply

III. Community water and sanitation (Ghanaian chronicle, Feb. 2008)

The Ghana Water Company Ltd (GWCL) is responsible for providing, distributing and conserving water for domestic , public and industrial purposes in eighty-two(82) urban systems in localities with more five thousand (5000) inhabitants. Moreover the company is mandated to establish, operate, and control sewage systems in Ghana (Wonder, 2007)

1.3 Profile of Abesim water supply

Abesim is a suburb of Sunyani in the Brong Ahafo Region. The Abesim treatment plant was established for the treatment and production of water by the Ghana Water Company for distribution to the Sunyani municipality where Asuakwa is part of it.

The source of the water is from Tano River in Abesim. Before surface water can be selected for the public water supplies, the following criteria are considered;

- Reliability of source – must have sufficient water throughout the whole year.
- Quality of raw water- less costly to treat.
- Distance from town or settlement –The father the distance the more expensive (pipes and pumping costs)
- Other uses of sources –irrigation etc.
- No waste discharge near source.
- Area with less turbulence and suspended solids.

These were all considered by Abesim water supply before using Tano River.

Surface water sources are open to contamination from human and animal waste and other pollutions. Consequently they are susceptible to contamination by organisms such as bacteria, viruses, parasites and also toxic material that can cause serious illness and disease. So properly designed and operated treatment system s which includes filtration and disinfection are effective in preventing waterborne illness.

1.4 Statement of the Problems

A lot of resources go into the running of pipe works due to ineffective pipe network system.

There is also lack of financial resources to maintain and extend the infrastructure.

People in Sunyani pay so much for water due to high cost of materials for distribution of water by GWCL. Those who are unable to pay their bills get their taps disconnected. This leaves such people with no choice but to go in for untreated water from other sources. As such there is prevalence of water borne diseases.

1.5 Objectives

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

Main

i. To find the minimum connection and location of the substation of Asuakwaa Township in Sunyani.

Specific

i. Model a map into a network.

ii. Use Prim's Algorithm to minimize the pipe network of Asuakwa Township in Sunyani.

iii. Use P-Median to locate the substation for the pipe network of Asuakwa Township in Sunyani.

1.6 Significance of the Study

It is envisage that the findings of this research will enable management to have in depth knowledge on how to minimize pipe network and finding the location point in general to enable better distribution of water from source to various destination point which will provide a good drinking water.

Irregular supply of water year round due to high cost of materials for distribution, therefore affects the livelihood of people depending on it in Sunyani Municipality.

Application of findings from this research will help reduce cost of water distribution. This means that there will be regular supply of water and so people will not resort to other sources of water. Quantity of water produced would increase and so could be extended to areas that do not have access to tap water like Asuakwa.

An administrative cost (i.e. a cost) is an indication or a measure of the performance associated with transmitting data via a particular network segment or network path. For example, an operator may determine a cost associated with a plurality of network paths between a source node and a destination node and select the network path associated with the least administrative cost to connect the data. Clearly this work proves to address these issues in a more systematic manner and can be referenced.

When people get good drinking water for all kinds of work, it prevents people from getting various kinds of waterborne diseases, because this will help provide good drinking water.

Finally, the study serves as partial fulfillment of the requirement for the Master of Science degree in industrial mathematics.

1.7 Scope of the Study

This research work is limited to Asuakwa in Sunyani Township. The map provided in Chapter 4 represents the pipe layout from source Abesim to various sub stations within the Sunyani Municipality. However, other areas in the country with similar characteristics could adopt the findings to the solutions of their own water supply.

1.8 Methodology

Since the study is on minimizing the pipe network system of Ghana Water Company Limited, Asuakwa Township, the researcher deployed the use of Prim's algorithm under shortest path algorithms etc.

The researcher used a map provided by the Ghana Water Company Limited, Brong Ahafo Regional Office, Sunyani. The map is the pipe network system of Sunyani Township and the distances they cover within the township.

Getting funds for treatment, production and distribution of water in Sunyani township specifically Asuakwa township, has been difficult task for long. Also laying of pipes to the new site like Asuakwa area is also a proplem, so the use of prim's algorithm which will reduce cost and provide the shortest path will help GWCL to use the little revenue they have to produce water to the people.

The lay out plan for the pipe network system for Asuakwa area was provided by the GWCL Sunyani –B/A, Sunyani Municipal Assembly. Most of the information provide in this thesis are from GWCL, Abesim water supply ,Town and country planning etc.

Manual program will be develop using the prim`s algorithm.

1.9 Organization of the Study

Chapter 1 presents the background to the study, problem statement, objectives, significance and justification of the study, scope of the study, limitations and organization of the study.

Chapter 2 focuses on review of relevant literature. This chapter discusses both the theoretical and empirical background literature.

Chapter 3 discusses the methodology.

Chapter 4 looks at data collection and analysis.

In chapter 5 we shall put forward the summary, conclusion and recommendations of the study.

CHAPTER 2

LITERATURE REVIEW

2.0 Introduction

We shall review some pertinent literature in the field of shortest path algorithms in this chapter. The aim of this study is to find the optimal network system to solve the problem of water supply to the people of Asuakwa in Sunyani Township, focusing on the use of shortest path algorithm approaches. This Chapter constitutes a literature review of the water supply, network, shortest path algorithms and Location point.

2.1 The Amount of Water Your Body Needs

Another important factor is the amount of water necessary for our body to function at its peak performance. Bearing in mind again that your body is about 75 percent water it is easy to understand that water must be your body's most essential daily ingredient. Your body looses each day about 2-3 litters of water through elimination, urination, perspiration and respiration. However, this may increase during illness, high performance, exercise, pregnancy and nursing. The beverages most people choose to consume are often counter-productive in promoting hydration. Coffee, tea, alcohol, soft and sugary drinks are all diuretics and will cause not only the loss of water that are dissolved in, but they will also draw water the bodies reserves. In normal conditions your body needs to replace the fluids it has lost throughout the day. Most of fluids should be replaced by drinking pure water. The rest you should get from fruit, vegetables and their juices. Attention must be given that the elderly and children are meeting their daily requirements. Dry mouth is not the only indication of dehydration; in fact it is the last sign. You need to acquire the habit to drink water even when you think you don't need it and eventually your true thirst mechanisms will be reawakened. Signs to look for that identify with dehydration are constipation, headaches, indigestion, weight gain, fluid retention, dark and pungent urine, and their associated pathologies colitis, kidney stones, bladder and urinary tract infections to name only a few (Batmanghelidj, Fereydoon, Dr. (1992)).

2.2 Network systems

According to Xu, (2008) the visualization of pipe network has become one of the essential characteristics of urban pipe network management system. But for the distribution of related resources (e.g. data storage, modeling platform etc.) and complex structure, the design and implementation of a pipe network of a pipe network visualization application is always a challenge. This contribution presents a service view in design and deployment phase to cope with the questions mentioned above, which takes separate resources as individual, services and then make an integration to present pipe network 3D visualization service architecture with those services. With this architecture, services providers will be able to assemble a pipe network 3D visualization service that serves the need of a company (or authority) responsible for some particular underground pipe network systems. This paper presents the pipe network visualization service architecture and a developed pipe network visualization service prototype.

Water distribution and supply network in Vyskov with approximately twenty thousand (20,000) inhabitants was built gradually with development of residential districts since the year 1936 and its length represents nearly one hundred (100) km of pipelines with diameter from DN 50 to DN 500. Dominant materials of pipelines acre cast iron and PVC. There are four main water sources and they supply the whole area together. With a respect to the great terrain elevation differences hydrostatic pressure reaches up to the value of 7.5 bar on a relatively extensive area. A high pressure evokes redundant loss of water in both the main pipe network and at single attached consumers. The objectives of the hydraulic analysis were focused on development, calibration and verification of a Mathematical model of water distribution and supply network. Simulations covered different alternatives of the given network under steady state and extended period conditions. Based on the simulation result, the existing water supply network was divided into several pressure zones. This lead to optimum hydraulic grade line distribution, decrease of water losses and savings of electricity consumptions required in pumping stations. Modeling real time control scenarios of the water distribution and supply network will help the network operator to model its breakdown, reconstruction and/or planning new pipes, cover new consumer's demands (Baranek et al., 1998).

Many social and biological networks consist of communities-groups of nodes within which connections are dense, but between which connections are sparser. Recently, there has been considerable interest in designing algorithms for detecting community structures in real-world complex networks. (Maini, 2005) proposed an evolving network which exhibits community structure. The network model is based on the inter-community preferential attachment and the inter-community preferential attachment mechanisms.

The degree of distributions of these network simulations indicates that this network model has community structure and scale-free properties.

Barab writes "The diversity of networks in business and the economy is mind-boggling. 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."

1. A water supply system or water supply network is a system of engineered hydrologic and hydraulic components which provide water supply. A water supply system typically includes: the watershed (area of land where surface water from rain and melting snow or ice converges to a single point , usually the exit of the basin , where the water join another water body , such as river , lake , reservoir , estuary , wetland, sea , or ocean)
2. A raw (untreated) water collection point(above all below ground) where the water accumulates, such as lake, a river or ground water from underground aquifer. Untreated drinking water (usually water being transferred to the water purification facilities) may be transferred using uncovered ground-level aqueducts, covered tunnels or underground water pipes.
3. Water purification facilities. Treated water is transferred using water pipes(usually underground)
4. Water storage facilities such as reservoirs, water tanks, or water towers. Smaller water systems may store the water in cisterns or pressure vessels. (All buildings

may also need to store water locally in pressure vessels in order for the water to reach the upper floors.)

5. Additional water pressurizing components such as pumping stations may need to be situated at the outlet of underground or above ground reservoirs or cisterns(if gravity flow is unfeasible)
6. A pipe network for distribution of water to the consumers (which may be private houses or industrial, commercial or institution establishment) and other usage points(such as fire hydrants)
7. Connections to the sewers(underground pipes or above ground ditches in some developing countries) are generally found downstream of the water consumers, but the sewer system is considered to a separate system, rather than part of the water supply system.

A new method for the direct least cost solution of pipe sizes of a branched pipe network of known geometry consisting of pumping station and network of pipes for known water consumptions incorporating various cost functions is being developed. The method developed gives optimum pumping head and total head loss in the network system and its optimum distribution among the various pipes of the overall least cot solution. The variation of the total cost with the total head loss in the system is studied. The method outline herein is simple and can be efficiently used for the least cost design of branched water main systems only, with the help of a desk calculator. A systemic procedure of calculation has been described to illustrate the applicability of the technique using two sample problems.

According to Shau, (2005), in the last three decades, a significant number of methods for optimal design of pipe network systems have been developed using the linear programming, non-linear programming, dynamic programming, enumeration techniques, and genetic algorithm (G.A). This paper represents genetic algorithm (G.A) approach to the design of pipe network systems. The objectives considered are minimization of the network cost in the practical requirements. Of all the preceding methods, G.A based methods appear to be robust, as they can handle discrete pipe sizes with ease and produce a set of alternatives. By judgmentally selecting the Ruey-Fang district water supply system for the case study and by comparing the data gathered from the case study, this study also aims to verify the efficiency of the proposed method. The finding indicates that the proposed method is superior to enumeration techniques with respect to solution speed and cost.

2.3 Shortest Path Algorithms

The shortest path problem is concerned with finding the shortest path from a specified origin to a specified destination in a given network while minimizing the total cost associated with the path. The shortest path problem is an archetypal combinatorial optimization problem having widespread applications in a variety of settings. The applications of the shortest path problem include vehicle routing in transportation system, traffic routing in communication networks, pipe network system and path planning in robotic systems. Furthermore, the shortest path problem also has numerous variations such as the minimum weight problem, the quickest path problem, the most reliable path problem, and so on.

In graph theory, the shortest path 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 from one location to another on a road map; in this case, the vertices represent location and 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 $f : E \to R$) and one element *v* of ***V***, find the 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'***

The problem is also sometimes called the single-pair shortest path problem, to distinguish it from the following generalizations:

- The single-source shortest path problem, in which we have to find shortest paths from a source vertex v to all other vertices in the graph.

- The single-destination shortest path problem in which we have to find shortest paths from all vertices in the graph to a single destination vertex ***v***. This can be reduced to single-source shortest path problem by reversing the edges in the graph.

- The all-pair shortest path problem in which we have to find shortest paths between every pair of vertices ***v***, v^1 in the graph.

These generalizations have significantly more efficient algorithms than the simplistic approach of running a single-pair shortest path algorithm on all relevant pairs of vertices. According to Mouli, (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 storage. In their projects they tried to reduce the runtime of conventional techniques by exploiting the physical structure of the road network and using network pre-processing technique. 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 travelled 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 (SP) problem has been well addressed using intelligent optimization techniques e.g., artificial neural networks, genetic algorithms (GAs), particles swarm optimization, and others. However , with the advancement in wireless communications, more and more mobile wireless networks is the topology dynamics, i.e., the network topology changes over time due to energy conservation or node mobility.

2.4 Some forms of Shortest Path Algorithms

There are several well-known algorithms for solving the shortest path problem, some of which are listed below. However, for the purpose of this work we will only concentrate on a few of them.

- Dijkstra's algorithm solves the singular-pair, single-source, and single destination shortest path problems.
- Bellman –Ford algorithm solves the single source problem if edge weights may be negative.
- A search algorithm solves for single pair shortest path using heuristics to try to speed up the search.
- Floyd – Warshall algorithm solves all pairs shortest paths.
- Johnson's algorithm solves all pairs shortest paths, and may be faster than Floyd-Warshall on sparse
- graphs.
- Perturbation theory finds (at worst) the locally shortest path.

2.5 Prim's Algorithm

In computer science, prim's algorithm is an algorithm that finds a minimum spanning tree for a connected weighted undirected 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. Prism's algorithm is an example of a greedy algorithm. The algorithm was developed 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, (2007) described parallel implementation of Prim's algorithm for finding a minimum spanning tree of a dense graph. Their algorithm uses 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 processor.

Gloor, (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 directed weighted graph. They believe that their system is particularly appropriate for greedy algorithm, though much of what 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 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, (2002), studied the expected performance of Prim`s Minimum spanning trees MST algorithm implemented using the 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.

2.6 Dijkstra's algorithm

Chang described the reasons about why it is beneficial to combine with graph theory and board game. For bye, it also descants three graph theories Dijsktra's, Prim's, and Kruskal's minimum spanning tree. Then it would describe the information about the board game they choose and how to and the game specifically.

Dijkstra's algorithm, conceived by Dutch computer scientist Edsger Dijkstra, is a graph search algorithm that solves the single-source shortest path problem for a graph with non-negative edge path costs, producing a shortest path tree. This algorithm is often used in routing. An equivalent algorithm was developed by Moore. For a given source vertex (node) in the graph, the algorithm finds the path with the lowest cost (i.e. the shortest path) between that vertex and every other vertex. It can also be used for finding cost 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 represents cities and edge path and costs represents 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 it 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 circle 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.

Buriol, (2008), described dynamic shortest –path algorithms as an update of the shortest paths taking into account a change in an arc weight. This paper describes a new generic technique that allows the reduction of heap sizes used by several dynamic single-destination shortest-path algorithms.

The proposed scheme finds the shortest paths using a simultaneous multi-path search method. In contrast with Dijkstra's algorithm, several nodes can be determined at one time. Moreover, we partition the network into different groups (networks groups) and find the all-node pair's shortest path in each group using a pipeline operation. Networks can be abstracted, and the shortest paths in very large networks can be found easily. The proposed scheme can decrease calculation time from O (N2) to O (N) using a pipeline operation on DAPDNA-2. Our simultaneous show that the proposed algorithm can be applied to the very large Internet network designs of the future.

Pangilinan, (2007), presented an overview of the multi-objective shortest path problem (MSPP) and a review of essential and recent issues regarding the methods to its solution. The paper further explored a multi-objective evolutionary algorithm as applied to the MSPP and describes its behavior in terms of diversity of solutions, computational complexity, and optimality of solutions to the MSPP in polynomial time (based on several network instances) and can be an alternative when other methods are trapped by the tractability problem.

Chang, described the shortest distance between two points as a straight line. But in the real world, if those two points are located at opposite ends of the country, or even in different neighborhoods, it is unlikely you will find a route that enables you to travel from origin to destination via one straight road. You might pull to determine the fastest way to drive somewhere, but these days, you are just as likely to use a Web-based service or a handheld device to help with driving directions. The popularity of mapping applications for mainstream consumer use once again has brought new challenges to the research problem knows as the "shortest-path problem." The shortest-path problem, one of the fundamental quandaries in computing and graph Theory, is intuitive to understand and simple to describe, In mapping terms, it is the problem of find the quickest way to get from one location to another. Expressed more formally, in a graph in which vertices are joined by edges and in which each edge has a value, or cost, it is the problem of finding the lowest-cost path between two vertices. There are already several graph-search algorithms that solve this basic challenge and its variations, so why is shortest path perennially fascinating to computer Scientists?

Goldberg, (2001), Principal researcher at Microsoft Research Silicon Valley said there are many reasons why researchers keep studying the shortest-path problem. "Shortest

path is an optimization problem that's relevant to a wide range of applications, such as network routing, gaming, and circuit design and mapping," Goldberg says. "The industry comes up with new applications all the time, creating different parameters for the problem. Technology with more speed and capacity allows us to solve bigger problems, so the scope of the shortest-path problem itself has become more ambitious. And now there are Web-based services, where computing time must be minimized so that we can respond to queries in real time.

2.7 Floyd-Warshall's Algorithm

The shortest path between the 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 the network. The Floyd-Warshall's algorithm obtains a matrix of shortest path distances with O{ n^3 } computations.

The Floyd- Warshall algorithm, also variously know as Floyd's algorithm, the Roy-Floyd algorithm, the Roy- Warshall algorithm, or the WFI algorithm, is an algorithm for efficiently and simultaneously finding the shortest paths (i.e., graph geodesies) between every pair of vertices in a weighted and potentially directed graph (Floyd, 1962).

Hougardy, the Floyd-Waeshall algorithm is a simple and widely used algorithm to compute shortest paths between all pairs of vertices in an edge weighted directed graph.

It can also be used to detect the presence of negative cycles. Hougardy, show that

For this task many existing implementations of the Floyd-Warshall algorithm will fail because exponentially large numbers can appear during its execution.

Misra et al., presented a new solution to the Dynamic All-Pairs Shortest Path

Routing Problems, suing a linear reinforcement learning scheme. The particular instance of the problem that we have investigate concerns finding the all-pairs shortest paths in a stochastic graph, where there are continuous probabilistically-based update 111 edge-weights. They presented the detail of the algorithm with an illustrative example. The

Algorithm can be used to find the all-pairs shortest paths for the "statistical" average graph, and the solution converges irrespective of whether there are new changes in edge-weights or not. On the other hand, the existing algorithms will fail to exhibit such a behavior and would recalculate the affected shortest paths after each edge-weight update. There are two important contributions of the proposed algorithm. The first contribution is that not all the edges in a stochastic graph are probed and even if they are, they are not all probed equally often .indeed .the algorithm attempts to almost always probe only those edges that will be included in the final list involving all pairs of nodes in the graph., while probing the other edges minimally .This increase performance of the proposed algorithm.

Hsieh, designed shortest path routing algorithm is in general more difficult than designing simple routing algorithms. In this paper, we derive a shortest path routing algorithm for pyramid network. The proposed algorithm takes a (1) time to determine a shortest path between any two nodes in a pyramid network. We also design a distributed routing algorithm so that an intermediate node takes a (1) time to confirm the next node along the shortest path without any centralized controller.

Traffic information systems are among the most prominent real-world applications of Dijkstra's algorithm for shortest paths. Schulz et al considered the scenario of a central information server in the realm of public railroad transport in wide-area network s. such a system has to process a large number of on-line queries for optional travel connections in real time in practice, this problem is usually solved by heuristic variations of Dijkstra's algorithm, which do not guarantee an optional result. They reported results from a pilot study, in which we focused on the travel time as the only organization criterion. In their study, various speed-up techniques for Dijkstra's algorithm were "snapshot" or half a million customer queries.

2.8 Kruskal Algorithm

Kruskal's algorithm is an algorithm in graph theory that finds a minimum spanning tree for connected weight graph. This means it find a subset of the edges that form 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.

An algorithm for computing a minimum spanning tree. It maintains a set of partial minimum spanning tree, 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 in1956, 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
  1. remove an edge with minimum weight from S
  2. If that edge connects two different trees, then add to the forest, combining two trees into a single tree.
  3. otherwise discard the edge

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

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

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. In fact, 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. (1979) 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.

Overall Strategy

Kruskal’s algorithm, as described in CLRS, is directly based on the generic MST algorithm. It builds the NST in forest. Initially, each vertex is in its own tree in forest. Then, algorithm considers each edge in turn, order 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, weight graph with a function w:E ➞R.

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

2.9 Summary

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

CHAPTER 3

METHODOLOGY

3.0 Introduction

The problem of supplying water from the Abesim Dam through tunnels and pipes to various homes, school, churches, organizations and industries in the Asuakwa township by the Ghana Water Company Limited (GWCL) can be set up as a shortest path problem.

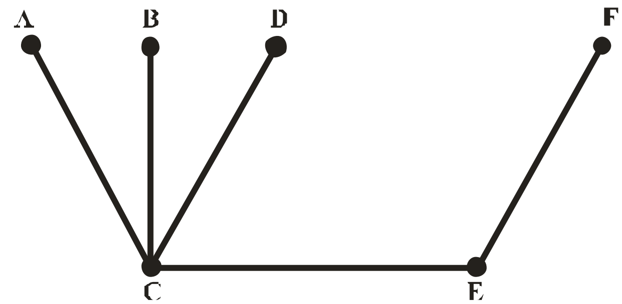
The supply problem involves the movement of treated water from the dam site to the various destinations passing through numerous tunnels and pipes. Shortest path problems are the most fundamental and most commonly encountered problem in the study of transportation and water companies. For the purpose of this research work the shortest, economical and fastest path from the supply plant to the Asuakwa Township.

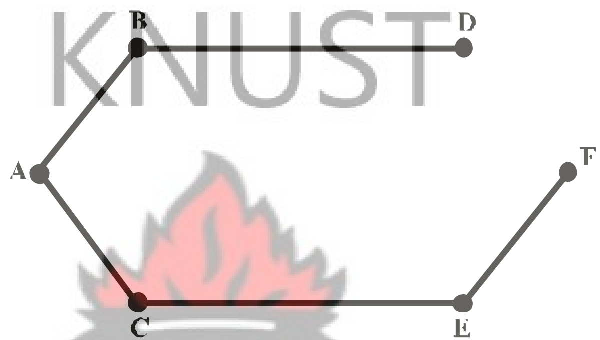
3.1 Minimum – Connector

These points are to be noted;

i. A connected graph which contains no cycles is called a tree.

ii. A spanning tree is a sub graph that includes all the vertices in the original graph and is also a tree. A graph will have several trees.





These are all spanning trees.

3.2 Data Collection

Information and Data needed for the analyses is been gathered from Ghana Water Company Limited,Town and country planning.

1. Sketch of the town Map showing the relative position of the areas with the distance (in kilometers) between them.
2. Statistical service.

3.3 Single-Source Path , Non-Negative Weight (Dijkstra's Algorithm)

Dijkstra 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 non-negative edge path costs, producing a shortest path tree. This algorithm is often used in routing. An equivalent algorithm was developed by Edward F. Moore in 1957.

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 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 IS – IS and OSPF (Open Shortest Path First).

3.4 Algorithm

Let the node from which we are starting be called an initial node. Let a distance of a node Y be the distance from the initial node to it. Dijkstra's algorithm will assign some initial distance values and will try to improve them step- by – step.

(i) Assign to every node a distance value. Set it to zero for our initial node and to infinity for all other nodes.

(ii) Mark all nodes as unvisited. Set initial node as current

(iii) For current node, consider all its unvisited neighbors and calculates their distance (from the initial node). For examples, if current node (A) has distance of 6, and an edge connecting it with another node (B) is 2, the distance to B through A will be 6+2 =8. If this distance is less than the previously recorded distance (infinity in the beginning, zero for the initial node), overwrite the distance.

(iv) When we are done considering all neighbors of the current node, mark it as visited. A visited node will not be checked ever again; its distance recorded now is final and minimal.

(v) Set the unvisited node with the smallest distance (from the initial node) as the next "current node" and continue from step iii.

Dijkstra's algorithm finds the shortest paths from a source node S to all other nodes in a network with non-negative are lengths. Dijkstra's algorithm maintains a distance label sources node to each node i. 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 basic idea of the algorithm is to find out from the source node S and permanently labeled nodes in the order of their distance from the node S. Initially, node S is assigned a permanent label of zero, 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 permanently labeled. The algorithm selects a node I with minimum temporary label (breaking ties arbitrarily), makes it permanent, and reaches out from node – that is, scans all the edges / arcs emanating from node I to update the distance labels of adjacent nodes.

The algorithm terminates when it has designated all nodes permanent.

We can now express Dijkstra's algorithm as a set of steps.

i. Assign the permanent label O to the starting vertex.

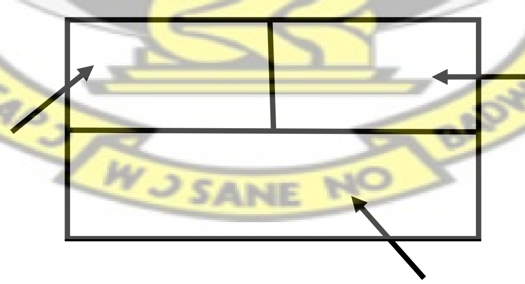
ii. Assign temporary labels to all the vertices that are connected directly to the most recently permanent labeled vertex.

iii. Choose the vertex with the smallest temporary label and assign a permanent label to that vertex.

iv. Repeat steps 2 and 3 until all vertices have permanent labels.

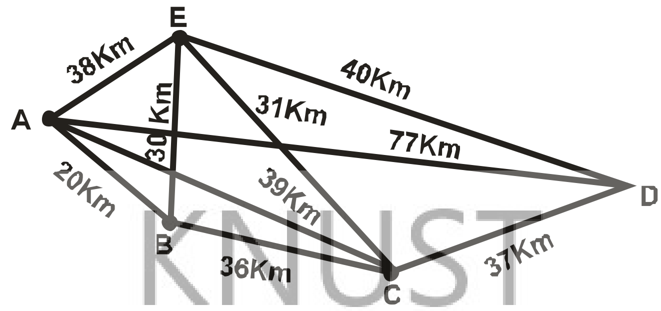
v. Find the shortest path by tracing back through the network.

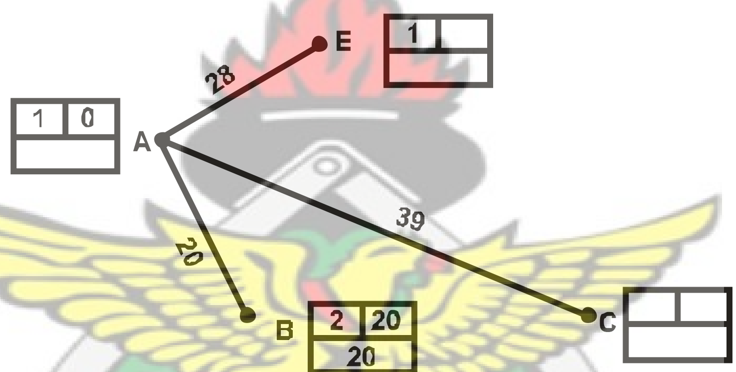
Note: Recording the order in which we assign permanent labels to the vertices is an essential part of the algorithm.

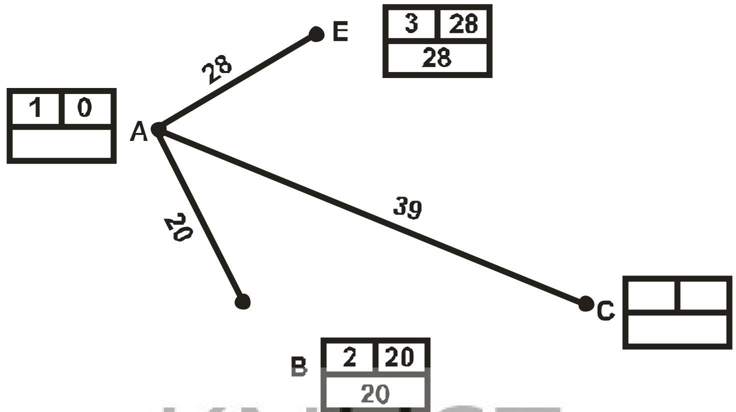


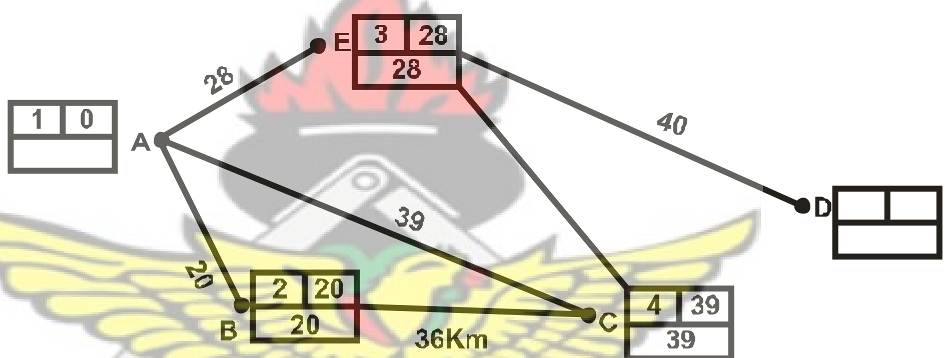
The algorithm gradually changes all temporary labels into permanent ones.

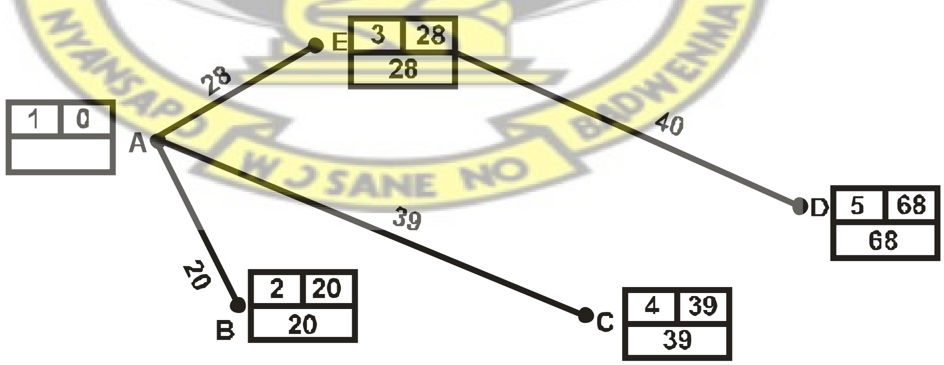
(Dijkstra’s Algorithm).

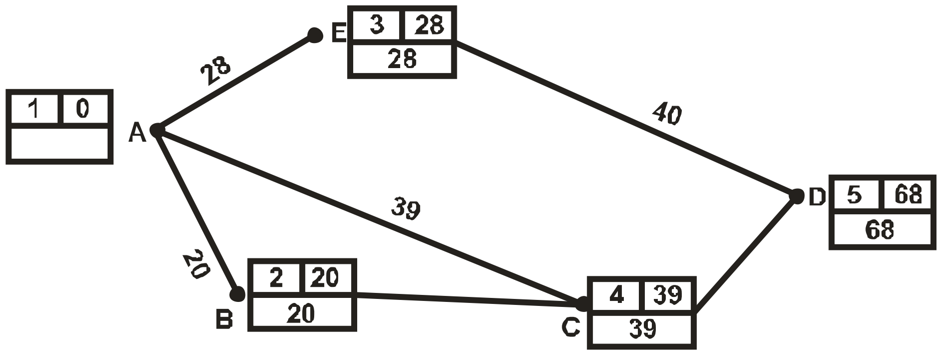





E
3
28
28
1
0
A
28
20
39
C
B
2
20
20


E
3
28
28
28
1
0
A
40
D
39
20
2
20
20
B
36Km
C
4
39
39


E
3
28
28
28
1
0
A
40
D
5
68
68
39
20
2
20
20
B
C
4
39
39



Now all the vertices have permanent labels and we can see that the shortest distance from A to D is 68.

We find the shortest path by working backward from D, ie. D → E → A.

Hence the shortest path from A to D is A → E → D with length 68.

3.5 All – Pairs Shortest Path Problem

The shortest path between two nodes might not be a direct edge between them. But instead involves a detour through other nodes. The all–pairs shortest path problem requires that we determine shortest path distances between every pair of nodes it nth network.

3.6 Floyd-Warshall Algorithm

The Floyd – Warshall algorithm obtains a matrix of shortest path distances 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 1,2,3,..,k – 1 as internal nodes clearly, d^{n+1} (i,j) represents the actual shortest path distance from i and 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 pairs of node i and j. It repeats the process until it obtains d^{n+1} (i,j) for all node pairs i and j then it terminates. Given $d^k(i, j)$, the algorithm computes $d^{k+1}(i, j)$= min { $d^k(i, k)$, $d^k(i, k$}. The Floyd – Warshall algorithm remains of interest because it handles negative weight edges correctly.

3.7 FLOYD – WARSHALL ALGORITHM

The Floyd – Warshall algorithm is also use to find the shortest distances between all pairs of nodes. The symbol as means that there is no direct connection between these vertices.

The matrix below represents the distance (in Km) of direct direction between five towns.

The algorithm can be stated as follows:

Step 1: Choose a starting vertex.

Step 2: Join this vertex to the nearest vertex directly connected to it.

Step 3: Join the nearest vertex, not already in the solution to any vertex in the solution, provided it does not form a cycle.

Step 5: Find the shortest path by tracing through the network.

Step 4: Repeat until all the vertices have been included

Table 3.1 Distance between all pairs of nodes

	A	B	C	D	E
A	___	20	∞	77	28
B	20	___	36	∞	30
C	∞	36	___	37	31
D	77	∞	37	___	40
E	28	30	31	40	___

i) From A to B (the direct distance from A to B is 20).

Since no other direct connection between A and any other node is less than 20. So we retain the value 20 as the minimum distance between A and B.

ii) A to C : (The direct distance is ∞)

A $\xrightarrow{20}$ B $\xrightarrow{36}$ C = 56 Now Min (56, 114, 69) = 56

A $\xrightarrow{77}$ D $\xrightarrow{37}$ C = 114

A $\xrightarrow{38}$ E 31 C = 69 So we replace ∞ from A to C and C to A by 56.

iii) A to D (The direct distance is 77).

A $\xrightarrow{20}$ B $\xrightarrow{\infty}$ D = ∞ Now Min (∞, 78)

A $\xrightarrow{38}$ E $\xrightarrow{40}$ D = 78 These values are not less than 77.

So we retain 77

iv) A to E (The direct distance 38)

Only A to B is less than 38.

A to B = 20. So we evaluate the distance.

A $\xrightarrow{20}$ B $\xrightarrow{30}$ E = 50

So we retain 38 in the cell AE and EA.

NOW FROM THE NODE B

i. From B to A (The direct distance is 20) BA with value 20, not other value in the cells beginning with B is less than 20, So we retain the value 20 in the cells BA and AB.

ii. B to C (The direct distance is 36)

B $\xrightarrow{20}$ A $\xrightarrow{\infty}$ C = ∞ } Now Min (∞, 61) which are not less than 36.

So we retain B $\xrightarrow{30}$ E $\xrightarrow{\infty}$ C = 61 the value 36 in the cell BC and CB.

iii. B to D (The direct distance is ∞)

B 20 A 77 D = 97 Now Min (97, 73, 70)

B 36 A 37 D = 73 } = 70

B 30 E 40 D = 70 So we replace ∞ from B to D and D to B by 70

iv. B to E (The direct distance is 30). Only B to A is less than 30. So we evaluate B 20 A 38 E = 58

Now Min (58) So we retain the value 30 in the cell BE. Since 58 is not less than 30.

<u>NOW FROM THE NODE C</u>

i. C to A (The direct distance is ∞)

C $\xrightarrow{36}$ B $\xrightarrow{20}$ A = 56   Now Min (56, ∞ , 69)

C $\xrightarrow{37}$ D $\xrightarrow{\infty}$ A = ∞   } = 56

C $\xrightarrow{31}$ E $\xrightarrow{38}$ A = 69   So we replace the value in the cell CA by 56.

ii. C to B (The direct distance is 36). Only C to E has value less than 36 so we evaluate .

C $\xrightarrow{31}$ E $\xrightarrow{30}$ B = 61

Since 61of the cell CB is not less than 36. We retain the cell CB which is 36.

iii. C to D (The direct distance is 37)

C $\xrightarrow{36}$ B $\xrightarrow{\infty}$ D = ∞   } Now Min (∞, 71)

C $\xrightarrow{31}$ E $\xrightarrow{40}$ D 71   = 71

Since 71 is not less than 37, we retain 37 in the cells CD and DC.

iv. C to E (The direct distance is 31). No other value beginning with C is less than 31, so we retain 31 in the cells CE and EC.

NOW FROM THE NODE D

i. D to A (The direct distance is 77)

D $\xrightarrow{37}$ C $\xrightarrow{\infty}$ A = ∞
D $\xrightarrow{40}$ E $\xrightarrow{38}$ A = 78

} We retain 77

ii. D to B (The direct distance ∞)

D $\xrightarrow{77}$ A $\xrightarrow{20}$ B = 97
D $\xrightarrow{37}$ A $\xrightarrow{36}$ B = 73
D $\xrightarrow{40}$ A $\xrightarrow{30}$ B =70

} Now Min (97, 73, 70) = 70. So we replace ∞ in the cell DB by 70

iii. D to C (The direct distance is 37). No value in the cell beginning with D is less than 37. So we retain the value 37.

iv. D to E (The direct distance 40)

D $\xrightarrow{37}$ C $\xrightarrow{31}$ E = 68

Since 68 is not less than 40, we retain 40.

Hence, the shortest distance between all pairs of nodes is as shown in the table below.

Table 3.2 The shortest distance between all pairs of nodes

	A	B	C	D	E
A	___	20	56	77	28
B	20	___	36	70	30
C	56	36	___	37	31
D	77	70	37	___	40
E	28	30	31	40	___

3.8 Prim's Algorithm

Prim's algorithm is an algorithm in graph theory that finds a minimum spanning tree for connected weighted graph. This means it finds a subset of the edges that forms a tree it includes every vertex, where the total weight of all the edges in the tree is minimized. If the graph is not connected, then it will only find a minimum spanning tree for one of the connected components. The algorithm was discovered in 1930 by mathematician Vojtech Jamik and later independently by computer scientist Robert Prim in 1957 and rediscovered by Dijkstra in 1959. Therefore it is sometimes called DJP algorithm or Jamik algorithm.

It works as follows:

- Create a tree containing a single vertex, chosen arbitrarily from the graph
- Create a set containing all the edges in the graph

- Loop until every edge in the set connects two vertices in the tree
    - remove from the set an edge with minimum weight that connects a vertex in the tree with a vertex not in the tree
    - add that edge to the tree

Prim's algorithm works from a starting point and build up the spanning tree step by step, connecting edges into the existing solution. It can be applied directly to the distance matrix, as well as to the network itself. So it is more suitable for using a computer if the network is large. The algorithm can be stated as follows:

Step 1: Choose a started vertex.

Step 2: Join this vertex to the nearest directly connected to it.

Step 3: Join the next nearest vertex, not already in the solution to any vertex in the solution provided it does not form a cycle.

Step 4: Repeat until all vertices have been included

PRIM'S ALGORITHM

	D	E	C	A	B
D	∞	40	37	77	∞
E	40	∞	31	28	30
C	37	31	∞	39	36
A	77	28	39	∞	20
B	∞	30	36	20	∞

Chose a starting vertex say D. Delete the row D. Look for smallest entry in column D.

1

	D	E	C	A	B
D	~~∞~~	~~40~~	~~37~~	~~77~~	~~∞~~
E	40	∞	31	28	30
C	37	31	∞	39	36
A	77	28	39	∞	20
B	∞	30	36	20	∞

DC is the smallest edge joining C to the other vertices put edge DC into the solution. Delete row C. Look from the smallest entry in columns D and C.

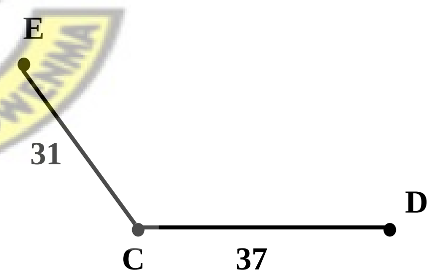
1 2

	D	E	C	A	B
E	40	∞	31	28	30
~~C~~	~~37~~	~~31~~	~~∞~~	~~39~~	~~36~~
A	77	28	39	∞	20
B	∞	30	36	20	∞

CE is the smallest edge D and C to the other vertex. Put edge EC into the solution. Delete row E. Look for the smallest entry in columns D, C and E.

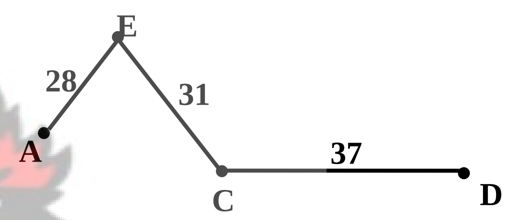
1 3 2

	D	E	C	A	B
~~E~~	~~40~~	~~∞~~	~~31~~	~~28~~	~~30~~
A	77	28	39	∞	20
B	∞	30	36	20	∞



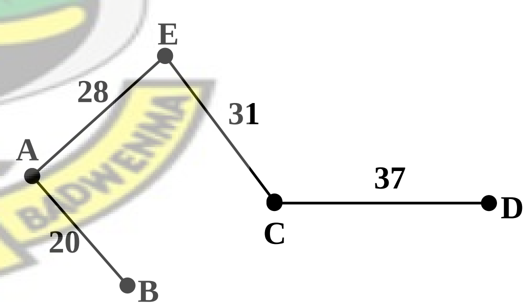
AE is the smallest edge joining D, C and E to the other vertices. Put edge EA into the solution. Delete row A. Look for the smallest entry in the columns D, C, E and A.

	1	3	2	4	
	D	E	C	A	B
A	77	28	39	∞	20
B	∞	30	36	20	∞



AB is the smallest edge joining D, C, E and A to the other vertices. Put edge BA into solution. Delete row B.

	1	3	2	4	5
	D	E	C	A	B
B	∞	30	36	20	∞



We have now connected all the vertices into the spanning tree.

Length = 37 + 31 + 28 + 20 = 116 Km.

This is the algorithm suitable to the minimum connector of this project.

3.9 P-Median

The solution to the model forms the basis of the results and analysis that will b presented in Chapter four (4). The purpose of this is to develop models for decision makers to minimize the total distance that will be covered by the people in all the nine (9) areas in the Town to have access to a substation at one of the areas.

The hypothesis in this research is that an optimal location can be chosen from the nine (9) areas in the municipality to serve the others with facility location methodology to ensure high patronage. Locations are selected from all areas with the help of the municipal capital map, a set of points were selected to represent the nine (9) areas.

P – median Problem Formation

The *p*-median is employed if the objective is to minimizing the weighted distance is the primary goal. This methodology can also be employed, as is the case in this research, to find the minimum weighted distance to locate the substation. So the appropriate objective then is to find the minimum of the calculated weighted (w,) distance for all the potential sites.

The *p*-median problem involves placing p facilities so that the total user cost or distance to travel to one of those facilities is minimized. The model can be represented mathematically as follows.

$$\text{Minimize } w_i = \sum_{i,j}^{n} h_i d_{ij} y_{ij} \qquad (1)$$

Subject to: $$\sum_{i} X_1 = n \qquad (2)$$

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

$$X_1 = \{ 0,1 \} \qquad (4)$$

$$Y_{11} = \{ 0,1 \} \qquad (5)$$

Where w_i = weighted 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_{ij} = distance between node *i* and node *j*

X_j = {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 o implies site *i* is not served by a facility at site *j*.

Computation of the Weighted Distance by p-Median

Given the sites (locations) the distance an individual should cover to have access to the facility at a selected site as well as the number of people commuting from the

various sites to the selected site, $w_i = \sum_{i,j}^{n} h_i d_{ij} y_i$ can be used to find the weighted distance and select the minimum of them as the optimal site. The distance between the various towns is put in a table form to form a matrix. The populations (h) of the various locations are noted.

For example, if five (5) suburbs A, B, C, D and E forming a town are linked with the distance between them and their population as shown:

The algorithm can be started as follows:

Step 1: Choose a starting vertex

Step 2: Join this vertex to the next vertex, not already in the solution

Step 3: multiply the distance of the next vertex by the population of it.

Step 4: Repeat it with all the other vertices until all vertices have been included.

Step 5: Sum all the products up.

Step 6: Find the minimum value.

P-Median

Figure: 3.1 The Tree of the Network of towns

P-median can be used to select the most appropriate suburb where desirable facility such as substation should be sited. This can be done as follows.

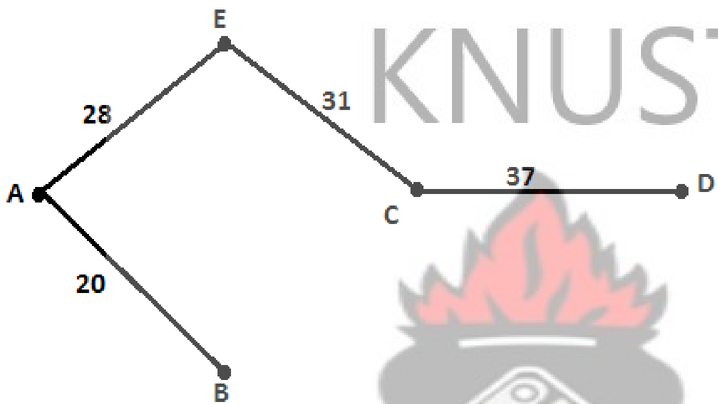


Table 3.3 Distance between the towns in table form.

		J						
			SITE	1	2	3	4	5
		SITE	TOWN	A	B	C	D	E
90	*I*	1	A	---	20	59	96	28
85		2	B	20	---	79	116	48
100		3	C	59	79	---	37	31
112		4	D	96	116	37	---	68
120		5	E	28	48	31	68	---

POPULATION OF THE TOWNS

	1	2	3	4	5
Towns	A	B	C	D	E
Population	90	85	100	112	120

$$wi = \sum_{i.j}^{n} h_i d_{ij} yi$$

where w_1, w_2, w_3, w_4 and w_5 are the total weighted distances for the five suburbs

d_{ij} = the distances between the other suburbs and the selected suburbs

h_i = the population of the four suburbs.

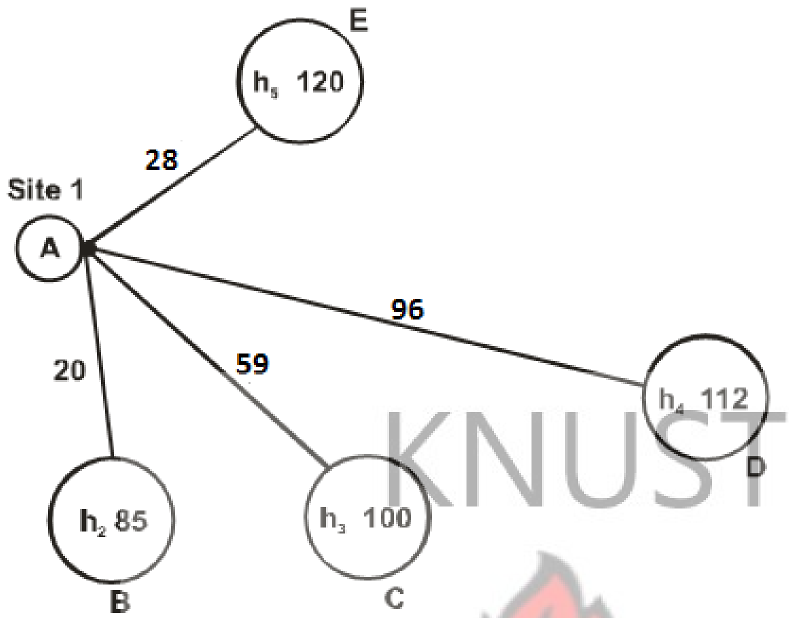
$n = 5$

w_1, w_2, w_3, w_4 and w_5 are calculated and the minimum of them is selected to locate the facility.

h_1, h_2, h_3, h_4 and h_5 are the populations of the suburbs A, B, C,D and E respectively.

The distance between a suburb and itself = 0, eg. From A to A is equal to zero (0).

The total weighted distance (W_1) of site 1 (A) is calculated as follows.



$$W_i = \sum_{i,j}^{n} h_i d_{ij} y_i$$

Subject to $\sum_{i} X_j = n$

$$\sum_{j} Y_{ij} = n$$

$$X_j = \{0, 1\}$$

$$Y_{ij} = \{0, 1\}$$

Where h_2, h_3, h_4 and h_5 = 85,100,112 and 120 respectively d_{12}, d_{13}, d_{14} and d_{15} =20, 59, 96 and 28 respectively.

$$\sum_{i} Y_{ij} = X_j = Y_{ij} = N = 1$$

Because each site will be served by all other sites

$$wi = \sum_{i.j}^{n} h_i d_{ij} y_{ij} \quad i = 2, 3, 4, 5 \quad j = 2, 3, 4, 5$$

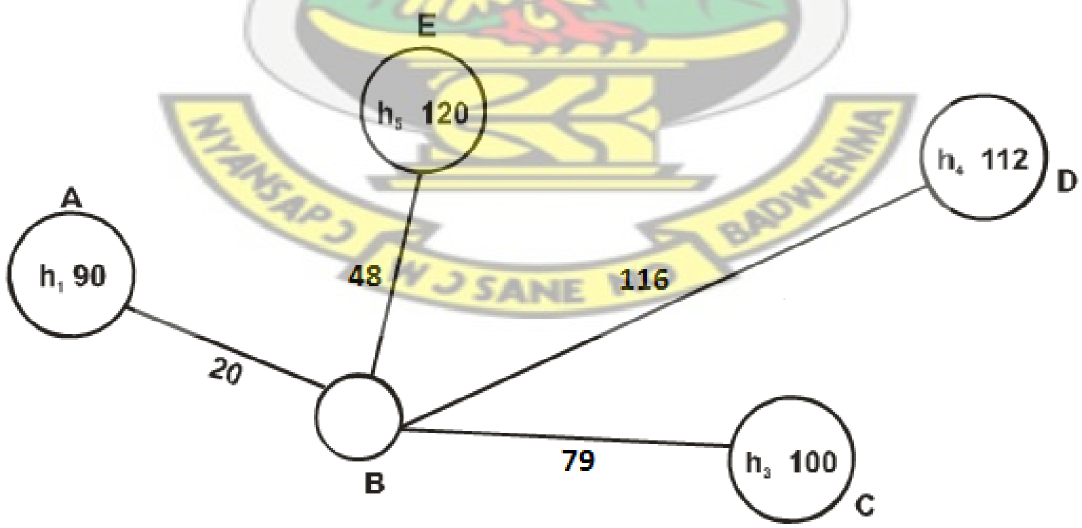
$W_i = d_{12}\, h_2 + d_{13}, h_3 + d_{14} h_4$

i. $W_i \sum_{i,j}^{n} h_i d_{ij} Y_{ij} = (20 \times 85) + (59 \times 100) + (96 \times 112) + (28 \times 120)$

$W_i = 21712$ Km

Therefore if w_i which is site A is selected, it means that for all the peoples from other suburbs to access the facility at town A, a total distance of 21712km must be covered.

The total weighted distance (w_2) of site 2 (B) is calculated as follows:



h_1, h_3, h_4 and h_5 = 90, 100, 112 and 120 respectively d_{21}, d_{23}, d_{24} and d_{25}

= 20, 48, 79, and 116 respectively.

$$W_2 = \sum_{i,j}^{n} h_i d_{ij} Y_{ij} \quad i = 2, 3, 4, 5 \qquad j = 2, 3, 4, 5$$

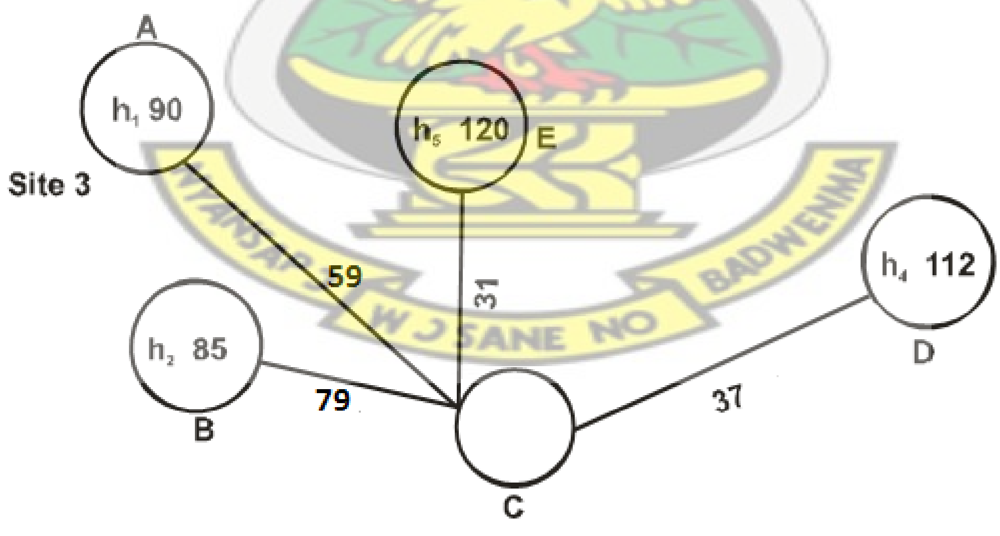
$$W_2 = d_{21}h_1 + d_{23}h_3 + d_{25}h_4 + d_{25}h_5$$

$$W_2 = \sum_{i,j}^{n} h_i d_{ij} Y_{ij} = (20 \times 90) + (48 \times 120) + (116 \times 112) + (79 \times 100)$$

$$W_2 = 28452km.$$

Therefore, if w_2 which is site B is selected, it means that for all the peoples from other suburbs to access the facility at town B, a total distance of 28452km must be covered.

The total weighted distance (w_3) of site 3(C) is calculated as follows:



h_1, h_2, h_4 and h_5 = 85, 90, 120, and 112

d_{31}, d_{32}, d_{34} and d_{35} = 79, 59, 31 and 37

$$W_3 = \sum_{i,j}^{n} h_i d_{ij} Y_{ij} \quad i = 2, 3, 4, 5 \qquad j = 2, 3, 4, 5$$

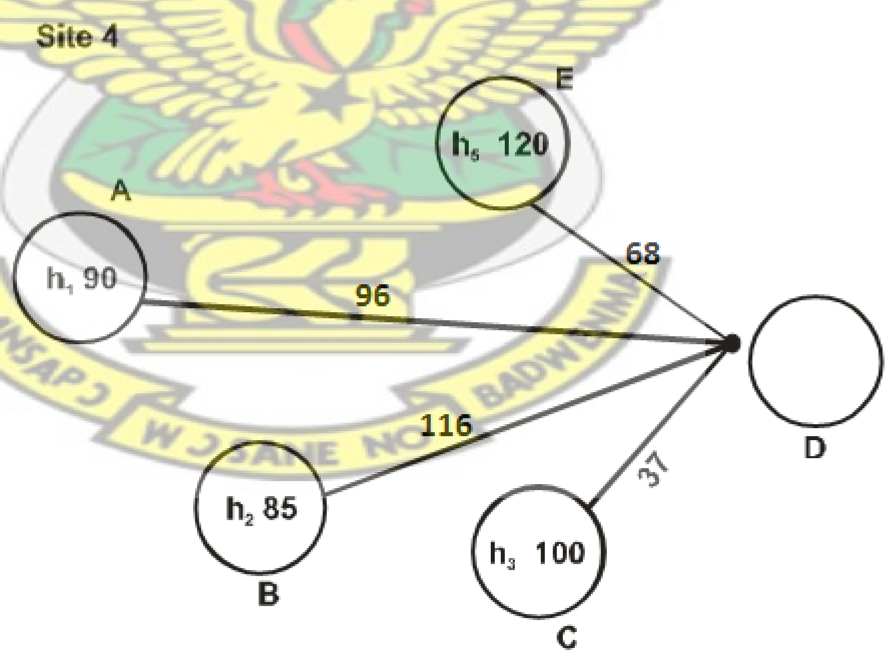
$W_3 = d_{31}h_1 + d_{32}h_2 + d_{34}h_4 + d_{35}h_5$

W_3 = (85 x 79) + (90 x 59) + (120 x 31) + (112 x 37)

W_3 = 19889km

Therefore if w_3 which is site C is selected, it means that all the peoples from other suburbs to access the facility at town C, a total distance of 19889km must be covered.

The total weighted distance (w_4) of site 4(D) is calculated as follows:



h_1, h_2, h_4 and h_5 = 90, 85, 100 and 120

d_{41}, d_{42}, d_{43} and d_{45} = 96 , 116, 37 and 68

$$W_4 = \sum_{i,j}^{n} h_i d_{ij} Y_{ij} \quad i = 2, 3, 4, 5 \qquad j = 2, 3, 4, 5$$

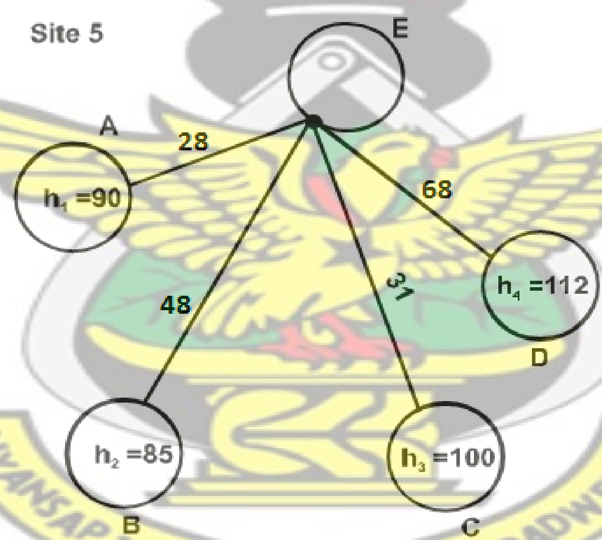
$W_4 = d_{41}h_1 + d_{42}h_2 + d_{43}h_3 + d_{45}h_5$

$W_4 = (90 \times 96) + (85 \times 116) + (100 \times 37) + (120 \times 68)$

$W_4 = 30360$km

Therefore if w_4 which is site D is selected, it means that for all the peoples from other suburb to access the facility at town D, a total distance of 30360km must be covered.

The total weighted distance (w_5) of site 5(E) is calculated as follows:



h_1, h_2, h_3 and h_4 = 90 , 85, 100 and 112 respectively

d_{51}, d_{52}, d_{53} and d_{54} = 28, 48, 31, 68 respectively

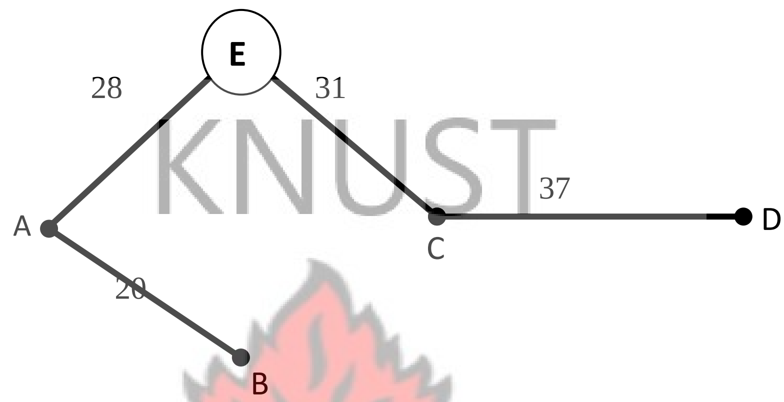
$$W_5 = \sum_{i,j}^{n} h_i d_{ij} Y_{ij} \quad i = 2, 3, 4, 5 \qquad j = 2, 3, 4, 5$$

$W_5 = d_{51}h_1 + d_{52}h_2 + d_{53}h_3 + d_{54}h_4$

$W_5 = (90 \times 28) + (85 \times 48) + (100 \times 31) + (112 \times 68)$

$W_5 = 17316km$

The minimum of the weighted distance above is 17316km which corresponds to site 5. Therefore town E is the best location for the substation.



CHAPTER 4

DATA COLLECTION AND ANALYSIS

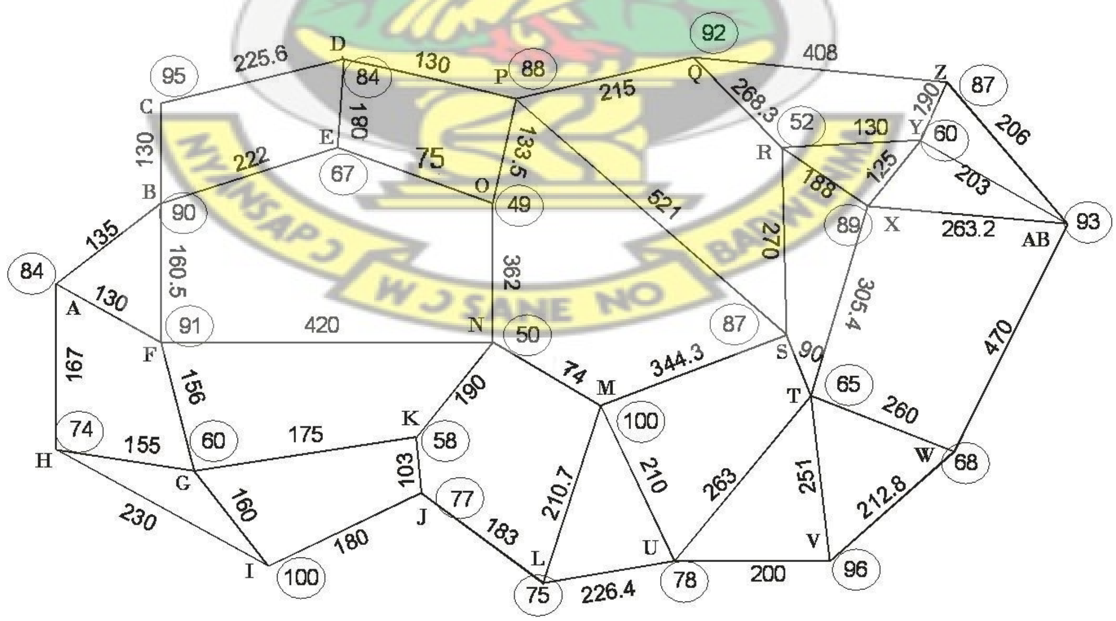
4.0 INTRODUCTION

In this chapter the data and the network taken from the Town and Country Planning Unit in Sunyani will be analyzed. P-median and Prim's algorithm are used to locate the substation and the least amount of pipe layout needed respectively.

4.1 DATA

Table 4.1 shows the suburbs and their respective population and Fig 4.2 also provides the network of the area.

Fig.4.1: Network of Asuakwa, Sunyani



Source: Town and Country Planning, Sunyani

Table 4.1: Population of the Suburbs.

S/N	SUBURB	POPULATION
1	A	84
2	B	90
3	C	95
4	D	84
5	E	67
6	F	91
7	G	60
8	H	74
S/N	SUBURBS	POPULATION
9	I	100
10	J	77
11	K	58
12	L	75
13	M	100
14	N	50
15	O	49
16	P	88
17	Q	92
18	R	52

19	S	87
20	T	65
21	U	78
22	V	96
23	W	68
24	X	89
25	Y	60
26	Z	87
27	AB	93

Source: Statistical Service Department, Sunyani

4.2 COMPUTATIONAL PROCEDURE FOR PRIM'S

The computer brand used in running the programming code was TOSHIBA with 150GB capacity hard disk drive, processing speed of 2.00GHz and a random access memory (RAM) of 1.00GB. The programming code was written in matlab to run data ***A*** shown in Appendix B. The programming codes can be found in the appendix A. The result is as shown below:

0	135	0	0	0	130	0	167	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
135	0	130	0	222	160 .5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	130	0	225 .6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	225 .6	0	180	0	0	0	0	0	0	0	0	0	0	130	0	0	0	0	0	0	0	0	0	0	0
0	222	0	180	0	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0	0	0	0	0	0	0	0
130	160 .5	0	0	0	0	156	0	0	0	0	0	0	420	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	156	0	155	160	0	175	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
167	0	0	0	0	0	155	0	230	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	160	230	0	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	180	0	103	183	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	175	0	0	103	0	0	0	190	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	183	0	0	210 .7	0	0	0	0	0	0	0	226 .4	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	210 .7	0	74	0	0	0	0	344 .3	0	210	0	0	0	0	0	0
0	0	0	0	0	420	0	0	0	0	190	0	74	0	362	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	125	0	0	0	0	0	0	0	0	362	0	133 .5	0	0	0	0	0	0	0	0	0	0	0
0	0	0	130	0	0	0	0	0	0	0	0	0	0	133 .5	0	215	0	521	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	215	0	268 .3	0	0	0	0	0	0	0	408	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	268 .3	0	270	0	0	0	0	188	130	0	0
0	0	0	0	0	0	0	0	0	0	0	0	344 .3	0	0	521	0	270	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	0	263	251	260	305 .4	0	0	0
0	0	0	0	0	0	0	0	0	0	0	226 .4	210	0	0	0	0	0	0	263	0	200	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	251	200	0	212 .8	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	260	0	212 .8	0	0	0	0	470
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	188	0	305 .4	0	0	0	0	125	0	263 .2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	130	0	0	0	0	0	125	0	90	203
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	408	0	0	0	0	0	0	0	90	0	206
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	470	263 .2	203	206	0

4.3 Prim's Algorithm

Prim's algorithm is an algorithm in graph theory that finds a minimum spanning tree for connected weighted graph. This means if finds a subset of the edges that forms a tree it includes every vertex, where the total weight of all the edges in the tree is minimized. If the graph is not connected, then it will only find a minimum spanning tree for one of the connected components. The algorithm was discovered in 1930 by mathematician Vojtech Jamik and later independently by computer scientist Robert Prim in 1957 and rediscovered by Dijkstra in 1959. Therefore it is sometimes called DJP algorithm or Jamik algorithm.

It works as follows:

- Create a tree containing a single vertex, chosen arbitrarily from the graph
- Create a set containing all the edges in the graph
- Loop until every edge in the set connects two vertices in the tree
    - remove from the set an edge with minimum weight that connects a vertex in the tree with a vertex not in the tree
    - add that edge to the tree

Prim's algorithm works from a starting point and build up t he spanning tree step by step, connecting edges into the existing solution. It can be applied directly to the distance matrix, as well as to the network itself. So it is more suitable for using a computer if the network is large. The algorithm can be stated as follows:

Step 1: Choose a started vertex.

Step 2: Join this vertex to the nearest directly connected to it.

Step 3: Join the next nearest vertex, not already in the solution to any vertex in the solution provided it does not form a cycle.

Step 4: Repeat until all vertices have been included

4.4 RESULTS FOR PRIM'S ALGORITHM

- Using the Prim's Algorithm, the table below shows the edges in minimum spanning tree and their distances:

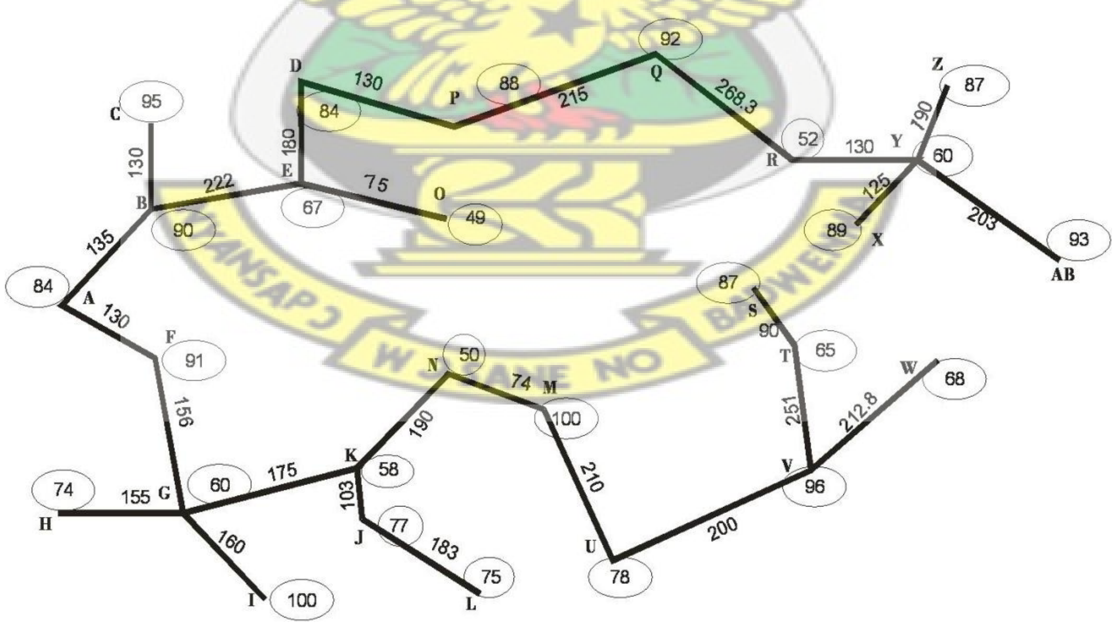
EDGES		DISTANCE (m)
FROM	TO	
A	F	130
A	B	135
B	C	130
F	G	156
G	H	155
G	I	160
G	K	175
K	J	103
J	L	183
K	N	190
N	M	74
M	U	210
U	V	200
V	W	212.8
B	E	222
E	O	75
E	D	180
D	P	130
P	Q	215
V	T	251

T	S	90
Q	R	268.3
R	Y	130
Y	Z	190
Y	X	125
Y	AB	203
TOTAL DISTANCE		**4,293.1**

- ❖ Hence the optimal pipe layout distance is **4,293.1m**

- ❖ The edges of the minimum spanning tree are indicated on the network below:

Fig.4.2: Spanning Tree From Prim's Algorithm



The shortest path of pipeline

4.5 COMPUTATIONAL DATA FOR P-MEDIAN

The following data from the spanning tree obtained from the Prim's Algorithm were used in the Matlab Codes for the P-Median at Appendix C.

d =

[0,135,0,0,0,130,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;135,0,130,0,222,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0;0,130,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;0,0,0,0,180
,0,0,0,0,0,0,0,0,0,0,130,0,0,0,0,0,0,0,0,0,0,0;0,222,0,180,0,0,0,0,0,0,0,0,0,0,75,0,0,0,0,0,0
,0,0,0,0,0,0;130,0,0,0,0,0,156,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;0,0,0,0,0,156,0,155,1
60,0,175,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;0,0,0,0,0,0,155,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0;0,0,0,0,0,0,160,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;0,0,0,0,0,0,0,0,0,0,103,183,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0;0,0,0,0,0,0,175,0,0,103,0,0,0,190,0,0,0,0,0,0,0,0,0,0,0,0,0;0,0,0,0,
0,0,0,0,0,183,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;0,0,0,0,0,0,0,0,0,0,0,0,0,74,0,0,0,0,0,0,210,
0,0,0,0,0,0;0,0,0,0,0,0,0,0,0,0,190,0,74,0,0,0,0,0,0,0,0,0,0,0,0,0,0;0,0,0,0,125,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;0,0,0,130,0,0,0,0,0,0,0,0,0,0,0,0,215,0,0,0,0,0,0,0,0,0,0;0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,215,0,268.3,0,0,0,0,0,0,0,0,0;0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,268.3
,0,0,0,0,0,0,0,130,0,0;0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,90,0,0,0,0,0,0,0;0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,90,0,0,251,0,0,0,0,0;0,0,0,0,0,0,0,0,0,0,0,0,210,0,0,0,0,0,0,0,0,200,0,0,
0,0,0;0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,251,200,0,212.8,0,0,0,0;0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,212.8,0,0,0,0,0;0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,125,0,0;0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,130,0,0,0,0,0,125,0,190,203;0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,190,0,0;0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,203,0,0]

h = [84,90,95,84,67,91,60,74,100,77,58,75,100,50,49,88,92,52,87,65,78,96,68,89,60,87,93]

4.6 COMPUTATIONAL PROCEDURE FOR P-MEDIAN

The computer brand used in running the programming code was TOSHIBA with 150GB capacity hard disk drive, processing speed of 2.00GHz and a random access memory (RAM) of 1.00GB. The programming code was written in matlab to run data ***d*** and ***h***. The programming codes can be found in the appendix C. The result is as shown below:

P – median Problem Formation

The *p*-median is employed if the objective is to minimizing the weighted distance is the primary goal. This methodology can also be employed, as is the case in this research, to find the minimum weighted distance to locate the substation. So the appropriate objective then is to find the minimum of the calculated weighted (w,) distance for all the potential sites.

The *p*-median problem involves placing p facilities so that the total user cost or distance to travel to one of those facilities is minimized. The model can be represented mathematically as follows.

$$\text{Minimize } w_i = \sum_{i,j}^{n} h_i d_{ij} y_{ij} \qquad (1)$$

$$\text{Subject to: } \sum_{i} X_1 = n \qquad (2)$$

$$\sum$$

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

j

$$X_1 = \{ 0,1 \} \qquad (4)$$

$$Y_{11} = \{ 0,1 \} \qquad (5)$$

Where w_i = weighted 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_{ij} = distance between node i and node j

X_j = {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 o implies site i is not served by a facility at site j.

Computation of the Weighted Distance by p-Median

Given the sites (locations) the distance an individual should cover to have access to the facility at a selected site as well as the number of people commuting from the

$$\sum^{n}$$

various sites to the selected site, $w_i = \sum_{i,j} h_i d_{ij} y_i$ can be used to find the weighted distance and select the minimum of them as the optimal site. The distance between the various towns is put in a table form to form a matrix. The populations (h) of the various locations are noted.

The algorithm can be started as follows:

Step 1: Choose a starting vertex

Step 2: Join this vertex to the next vertex, not already in the solution

Step 3: multiply the distance of the next vertex by the population of it.

Step 4: Repeat it with all the other vertices until all vertices have been included.

Step 5: Sum all the products up.

Step 6: Find the minimum value.

4.7 RESULTS FOR P-MEDIAN

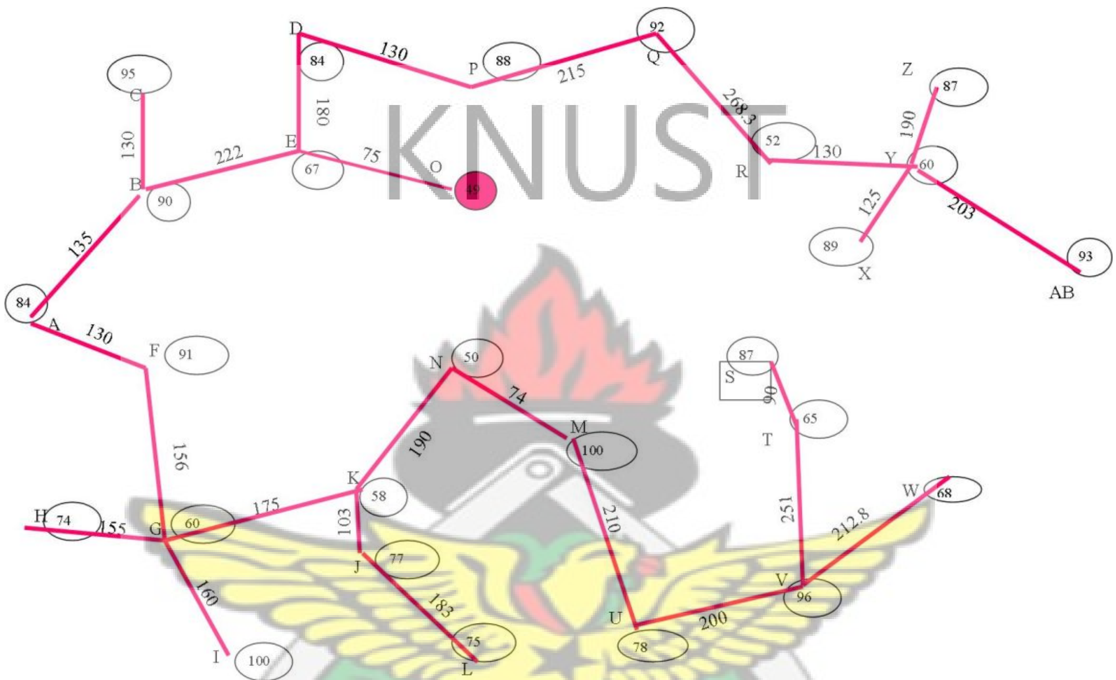
- Using the P-median, the table below shows the weighted distance for each suburb:

S/N	SUBURB	WEIGHTED DISTANCE(m)
1	A	23,980
2	B	38,564
3	C	11,700
4	D	23,500
5	E	38,775

6	F	20,280
7	G	51,816
8	H	9,300
9	I	9,600
10	J	19,699
11	K	27,931
12	L	14,091
13	M	20,080
14	N	18,420
15	O	5,025
16	P	30,700
17	Q	32,872
18	R	32,484
19	S	12,350
20	T	40,626
21	U	40,200
22	V	46,385
23	W	20,429
24	X	7,500
25	Y	53,294
26	Z	11,400
27	AB	12,180

❖ Hence Suburb **O** with the minimum weighted distance of **5,025m** is chosen to be the substation.

Fig.4.3: Spanning tree and location point



CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.0 CONCLUSIONS

The network and the data provided by The Town and Country Planning-Sunyani were analyzed and modeled to suit P-median and Prim's Algorithm in determining the substation and the optimal layout of the pipe network system of Asuakwaa in Sunyani.

The result shows that Prim's algorithm employed to find the optimal solution was able to reduce the total pipe network system of about 30% of the original network.

5.1 RECOMMENDATIONS

It was realized from the conclusion that finding the optimal solution to the pipe network system for the supply path for efficient supply, a solution-oriented modeling approach can help to devise efficient automatic planning and optimization schemes for Sunyani municipal.

It is also recommended that Ghana Water Company Limited (GWCL) be educated to use scientific methods to help them to reduce cost during layout of pipe network system.

REFERENCES

Algorithms, Second Edition. MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7. Section 23.2: The algorithms of Kruskal and Prim, pp. 567-574

Alp O, Erkut E, and Drezner Z, (2003) An efficient genetic algorithm for the p-median problem. Annals of Operations Research.

Arun K. D. (1974) Least Cost Design of Branched Pipe Network System, Roy F. Weston, Inc., West Chester, PA

Avellaand P, Sforza A, (1999) Logical reduction tests for the p-median problem. Annals of Operations Research.

Baranek, P; Kostek , J; Havlik, V (1998) Hydraulic Analysis of Water Supply Network. Czech Technical University., Thakurova 7, 16629, Praha 6, Czech Republic

Batmanghelidj, Fereydoon, Dr. (1992) *Your Body's Many Cries for Water*, Global Health Solutions, ISBN 0-9629942-3-5

Beasley J. E. (1985) A note on solving large p-median problems. European Journal of Operational Research.

Cao, H; Jundong Z; Aiping Z; Baozhu J and Guang R (2010) Simulation and control of marine Complex Pipe Network System based on distribution processing technology, Dalian Maritime Univ., Dalian China

Cherita L. C, Raheem A. B, John A. C, (2006) Using Active Scanning to Identify Wireless network interface card U.S. Military Academy, West Point, New York

Cormen T. H., Leiserson C. E. and Rivest R. L. (2001), and Clifford Stein. Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7. Section 23.2: The algorithm of Kruskal and Prim, pp. 567-574

Dijkstra, E. W. and Scholten G. S. (1980) Termination detection for diffusing computations. Info. Proc. Lett.,

Floyd, R. W. (1962) Algorithm 97: Shortest path. Communications of the ACM Illinois Institute of Technology, Illinois

Gloor, P. A., Laubacher, R., Dynes, S.B.C. & Zhao, Y. (1993) Visualization of communication patterns in collaborative innovation networks. Preceedings of the 12th int. Conference on Information and Knowledge Management (pp. 56-60). New Orleans, USA.

Goldberg, A. (2001) Shortest path algorithms; Engineering aspects. Springer-Verlag, London, UK

Gonina, E and Kale, L. V. (2007) Parallel Prim's algorithm on dense graphs with a novel extension. Technical Report. Department of Computer Science, University of Illinois at Urbana-Champaign.

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

Kruskal J. B (1956): on the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. In: Proceedings of the American Mathematics Society, Vol 7, No.1 (Feb, 1956), pp. 48-50

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

Pangilinan. J. M. A. and Janssens, G. K (2007) Evolutionary Algorithms for the Multi-objective Shortest Path Problem. Saint Louis University Press, Baguio, 2600 Philippines.

Prim R. C. (1957) shortest connection networks and some generalizations. In: Bell System Technical Journal.

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

Revelle C S and Swain R W 1970,Central Fascilities Location, Geographical Analysis 2, 30-42

Rosing, K., ReVelle C., and Rosing-Vogelaar H., 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.

Sam-Okyere E. (2010) The state of Ghana's Water Resources. Department of Geological Engineering KNUST, Kumasi

Shau H, Bi-Liang L, and Wen-Chih H. (2005) Genetic Algorithms For Design Of Pipe Network Systems, Journal of Marine Science and Technology, Vol. 13, No. 2, pp.116-124

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

APPENDIX A

MATLAB CODES FOR P-MEDIAN

```
% 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)'

[min_weighted_distance,location]=min(w)
```

APPENDIX B

MATLAB CODE FOR PRIMS 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

COMPUTATIONAL DATA FOR PRIM'S

Using the Matlab Codes for the Prim's Algorithm at Appendix B, the following data was used:

A:[0,135,1000,1000,1000,130,1000,167,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,
1000,1000,1000,1000,1000,1000,1000,1000,1000;135,0,130,1000,222,160.5,1000,1000,1000,10
00,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000
;1000,130,0,225.6,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,
1000,1000,1000,1000,1000,1000,1000,1000,1000;1000,1000,225.6,0,130,1000,1000,1000,1000,
1000,1000,1000,1000,1000,1000,130,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,100
0;1000,222,1000,130,0,1000,1000,1000,1000,1000,1000,1000,1000,1000,125,1000,1000,1000,1
000,1000,1000,1000,1000,1000,1000,1000,1000;130,160.5,1000,1000,1000,0,156,1000,1000,10
00,1000,1000,1000,420,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000;
1000,1000,1000,1000,1000,156,0,155,160,1000,175,1000,1000,1000,1000,1000,1000,1000,1000,
1000,1000,1000,1000,1000,1000,1000,1000;167,1000,1000,1000,1000,1000,155,0,230,1000,100
0,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000;1000,
1000,1000,1000,1000,1000,160,230,0,180,1000,1000,1000,1000,1000,1000,1000,1000,1000,100
0,1000,1000,1000,1000,1000,1000,1000;1000,1000,1000,1000,1000,1000,1000,1000,180,0,103,
183,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000;1000,10
00,1000,1000,1000,1000,175,1000,1000,103,0,1000,1000,190,1000,1000,1000,1000,1000,1000,1
000,1000,1000,1000,1000,1000,1000;1000,1000,1000,1000,1000,1000,1000,1000,1000,183,100
0,0,210.7,1000,1000,1000,1000,1000,1000,1000,226.4,1000,1000,1000,1000,1000,1000;1000,10
00,1000,1000,1000,1000,1000,1000,1000,1000,1000,210.7,0,174,1000,1000,1000,1000,344.3,10
00,210,1000,1000,1000,1000,1000,1000;1000,1000,1000,1000,1000,420,1000,1000,1000,1000,1
90,1000,174,0,362,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000;1000,1000

,1000,1000,125,1000,1000,1000,1000,1000,1000,1000,1000,362,0,133.5,1000,1000,1000,1000,1
000,1000,1000,1000,1000,1000,1000;1000,1000,1000,130,1000,1000,1000,1000,1000,1000,100
0,1000,1000,1000,133.5,0,215,1000,521,1000,1000,1000,1000,1000,1000,1000,1000;1000,1000,
1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,215,0,268.3,1000,1000,1
000,1000,1000,1000,1000,408,1000;1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,100
0,1000,1000,1000,1000,1000,268.3,0,270,1000,1000,1000,1000,188,130,1000,1000;1000,1000,1
000,1000,1000,1000,1000,1000,1000,1000,1000,1000,344.3,1000,1000,521,1000,270,0,90,1000,
1000,1000,1000,1000,1000,1000;1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1
000,1000,1000,1000,1000,1000,1000,90,0,263,251,260,305.4,1000,1000,1000;1000,1000,1000,1
000,1000,1000,1000,1000,1000,1000,1000,226.4,210,1000,1000,1000,1000,1000,1000,263,0,200
,1000,1000,1000,1000,1000;1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1
000,1000,1000,1000,1000,1000,1000,251,200,0,212.8,1000,1000,1000,1000;1000,1000,1000,10
00,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,260,1000,
212.8,0,1000,1000,1000,470;1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,
1000,1000,1000,1000,1000,188,1000,305.4,1000,1000,1000,0,125,1000,263.2;1000,1000,1000,1
000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,130,1000,1000,1000
,1000,1000,125,0,90,203;1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,100
0,1000,1000,1000,408,1000,1000,1000,1000,1000,1000,1000,90,0,206;1000,1000,1000,1000,10
00,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000,1000
,470,263.2,203,206,0]