LOCATING STANDPIPE ON AN OPTIMAL TREE NETWORK FOR WATER DISTRIBUTION IN ST. AUGUSTINE'S SENIOR HIGH SCHOOL – BOGOSO

BY

VINCENT FORSON (B.ED MATHEMATICS)

A THESIS SUBMITTED TO THE DEPARTMENT OF MATHEMATICS

KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCES

INDUSTRIAL MATHEMATICS

INSTITUTE OF DISTANCE LEARNING

JUNE 2014

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 or materials, which have been accepted for the award of any other degree of the University, except where the acknowledgement has been made in the text.

Vincent Forson	…………. …………	………………………
PG6319711	Signature	Date

Certified by

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

Certified by

……………………….	………………………	……………………….
Dean of IDL	Signature	Date

Certified by

Mr. K. F. Darkwah	……………………	………………………
Head of Department	Signature	Date

Abstract

Water is life, water is essential that you cannot live without it. The staff of Augusco who stay on campus find it difficult to get water as a result of pipelines not being connected to their houses. The installation of these pipelines to the bungalows is capitals intensive. Therefore the focus of this thesis seeks to find the minimum connection of pipe network using Prim's algorithm, to model location of standpipe on the minimum connection network as 1 median problem and to find the optimal location of the standpipe using ReVelle and Swain algorithm. Problem of pipe connection in Augusco was model as a minimum connector problem and the problem of locating standpipe as 1 – median problem. Data collection was mainly from secondary source. Prim's algorithm and ReVelle and Swain algorithm were used to minimize the pipeline network system of the school and location of standpipes respectively and were done by coding on a matlab platform. The total distance for the minimum spanning tree is **259406.1m**. The minimum weighted distance is **1856m** which occurs at the node 5 or locality E.

TABLE OF CONTENT

CHAPTER TWO: LITERATURE REVIEW **14**

CHAPTER THREE: METHODOLOGY

CHAPTER FOUR: DATA COLLECTION AND ANALYSIS

LIST OF TABLES

Content **Page**

LIST OF FIGURES

Content **Page**

DEDICATION

I dedicate this thesis to my parents: Armstrong Forson and Grace Nyame and all my siblings for their supports.

ACKNOWLEDGEMENT

Honestly, without my creator, God Almighty, I would not have reached this far. May his name be praised.

I ask for God's bountiful blessings for my supervisor Mr. Kwaku Dakwah, for his excellent direction, invaluable feedback, his constructive suggestions, detailed corrections, support and encouragement played enormous role resulted in this successful project.

I would also like to extend my gratitude to all the academic and administrative staff of the department of mathematics of KNUST.

I shall forever remain indebted to Elder Emmanuel Awuah, Anthony Donkor, Wisdom Naninja, Ebenezer Adjei for the love and immense contribution to the success of this project and to the staff of St. Augustine's Senior High School – Bogoso for providing me with all the information needed for the project.

CHAPTER ONE

INTRODUCTION

1.1 Background

Next to air (oxygen), water is the most essential element to human life; the body usually cannot survive longer than several days without water (a maximum of 1 week).

Water is essential to the functioning of every single cell and organ system in the human body.

Water makes up greater than 2/3 of the weight of the human body; the brain is 75% water, blood is 83% water, bones are 22% water, muscles are 75% water, and the lungs are 90% water (Snowflake's contributor profile).

Water is essential for the efficient elimination of waste products through the kidneys.

Water plays a role in the prevention of disease. Drinking adequate amounts of water can reduce the risk of colon and bladder cancer significantly, and some studies have suggested that water may also decrease the risk of breast cancer.

A decrease of as little as 2% in our body's water supply can have harmful effects and cause symptoms of dehydration, such as daytime fatigue, excess thirst, fuzzy memory, difficulty focusing on tasks and simple math, lightheadedness, and nausea.

A week without water will surely result in death. With the above functions, it clearly means water is really a life.

Apart from its importance to the body, it can also use for other purposes which includes domestic, industrial, agriculture, power generation, household recreational and environmental activities etc.

The share of non-functional supply systems in Ghana is estimated at almost one third, with many others operating substantially below designed capacity. Moreover, domestic water supply competes with a rising demand for water by the expanding industry and agriculture sectors. Ghana aims at achieving 85% coverage for water supply and sanitation by 2015, which would exceed the Millennium Development Goals' target of 78%. According to one estimate, only one quarter of the residents in Accra receive a continuous water supply, whereas approximately 30% are provided for 12 hours each day, five days a week. Another 35% is supplied for two days each week. The remaining 10% who mainly live on the outskirts of the capital are completely without access to piped water. According to another source, the situation is even worse: In February 2008 some communities within the Accra-Tema metropolis were served either once in a week, once in a fortnight or once in a month. The lack of clean drinking water and sanitation systems are severe public health concern in Ghana, contributing to 70% of diseases in the country. Consequently, households without access to clean water are forced to use less reliable and hygienic sources, and often pay more.

It is estimated that in 2000 the urban areas of Ghana generated about 763,698 m^3 of wastewater each day, resulting in approximately 280 million m^3 over the entire year. Regional capitals count for another 180 million m^3 (Agodzo 2008). Only a small share of the generated urban wastewater is collected, and an even smaller share is being treated. In Accra, the share of wastewater collected is approximately 10%. Moreover, less than 25% of the 46 industrial and municipal treatment plants in Ghana were functional according to an inventory undertaken by the Ghana

Environmental Protection Agency in 2001. Treatment plants for municipal wastewater are operated by local governments, and most of them are stabilization ponds. A biological treatment plant has been built in the late 1990s at Accra's Korle Lagoon. However, it only handles about 8% of Accra's wastewater.

Ghana is well endowed with water resources. The Volta river basin, consisting of the Oti, Daka, Pru, Sene and Afram rivers as well as the white and black volta rivers, cover 70% of the country area. Another 22% of Ghana is covered by the southwestern river system watershed comprising the Bia, Tano, Ankobra and Pra rivers. The coastal river system watershed, comprising the Ochi-Nawuka, Ochi Amissah, Ayensu, Densu and Tordzie rivers, covers the remaining 8% of the country.

Furthermore, groundwater is available in mesozoic and cenozoic sedimentary rocks and in sedimentary formations underlying the Volta basin. The Volta Lake, with a surface of 8,500 km^2, is one of the world's largest artificial lakes. In all, the total actual renewable water resources are estimated to be 53.2 billion m^3 per year.

In 2000, total water withdrawal was 982 million m^3, of which two thirds was used for agricultural purposes. Another 10% was withdrawn for industry, leaving 24% or 235 million m^3 for domestic use. Furthermore, 37,843 km^3 are used for hydroelectricity generation at the Akosombo Dam each year.

However, the drinking water supply and sanitation sector in Ghana face a number of challenges, including very limited access to sanitation, intermittent supply, high water losses and low water pressure. Since 1994, the sector has been gradually reformed through the creation of an autonomous regulatory agency, introduction of private sector participation, decentralization of the rural supply to 138 districts and increased community participation in the management of rural water systems. An international company managed all urban water systems since 2006 under a 5-year management contract which expired after achieving only some of its objectives. The reforms also aim at increasing cost recovery and a modernization of the urban utility Ghana Water Company Ltd. (GWCL). Another problem which partly arose from the recent reforms is the existence of a multitude of institutions with overlapping responsibilities. The National Water Policy (NWP), launched at the beginning of 2008, seeks to introduce a comprehensive sector policy.

The water supply problem is not different in Augusco, as the workers have to travel long distances before getting water for use. This problem is therefore compelling the school administration and the PTA to connect pipelines in the school.

1.2 History and recent development of Ghana Water Company Limited

Ghana Water Company Limited was established on 1st July 1999, following the conversion of Ghana Water and Sewerage Corporation into a state-owned limited liability company under the Statutory Corporations (Conversion to Companies) Act 461 of 1993 as amended by LI 1648.

The Ghana Water Company Ltd. (GWCL) is responsible for providing, distributing, and conserving water for domestic, public, and industrial purposes in 82 urban systems in localities with more than 5,000 inhabitants. Moreover, the company is mandated to establish, operate, and control sewerage systems in Ghana. Local private companies are in charge of meter installation, customer billing, and revenue collection. In the framework of the urban water project, since October 2006 the private operator AVRL supports GWCL under a five-year management contract to improve its performance and rehabilitate and extend the infrastructure. Urban sanitation is a responsibility of local governments

The first public water supply system in Ghana, then Gold Coast, was established in Accra just before World War I. Other systems were built exclusively for other urban areas among them the colonial capital of Cape Coast, Winneba and Kumasi in the1920s.

During this period, the water supply systems 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 systems in other parts of the country.

In 1948, the Department of Rural Water Development was established to engage in the development and management of rural water supply through the drilling of bore holes and construction of wells for rural communities.

After Ghana's independence in 1957, a Water Supply Division, with headquarters in Kumasi, was set up under the Ministry of Works and Housing with responsibilities for both urban and rural water supplies.

During the dry season of 1959, there was severe water shortage in the country. Following this crisis, an agreement was signed between the Government of Ghana and the World Health Organisation for a study to be conducted into water sector development of the country.

The study focused on technical engineering, establishment of a national water and sewerage authority and financing methods. Furthermore the study recommended the preparation of a Master Plan for water supply and sewerage services in Accra-Tema covering the twenty-year period 1960 to 1980.

In line with the recommendations of the WHO, the Ghana Water and Sewerage Corporation (GWSC), was established in 1965 under an Act of Parliament (Act 310) as a legal public utility entity. 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 the making of engineering surveys and plans.
- The construction and operation of water and sewerage works,
- The setting of standards and prices and collection of revenues.

By the late 1970s and early 1980s, the operational efficiency of GWSC had declined to very low levels mainly as a result of deteriorating pipe connections and pumping systems. A World Bank report in 1998 states that: "The water supply systems in Ghana deteriorated rapidly during the economic crises of the 1970´s and early 1980´s when Government's ability to adequately operate and maintain essential services was severely constrained."

GWSC largely experienced operational difficulties because of inadequate funding. From its inception, GWSC depended solely on government subvention to meet both operational and development costs. However, the annual government subvention was not adequate to meet the operational and development needs of the Corporation over the period. In addition, the annual subvention was either often not released on time or in most cases not released at all before the end of the budgetary year.

GWSC therefore met its operating costs at a level constrained by unavailability or inadequacy of funds. The lack of funds to meet operational costs resulted in the poor state of existing infrastructure especially, the distribution systems.

Before 1957, there were 35 pipe-borne water supply systems in the country. In a bid to promote rapid national development after Ghana's Independence, the government launched a crash programme for urban water expansion and accelerated rural development. As a result, by 1979 there were 194 pipe-borne and 2,500 hand pumped borehole systems in the country. By 1984, additional 3000 boreholes had been drilled and fitted with hand pumps. However by the late 1980's and early 1990, 33% of the water supply systems had deteriorated greatly or completely broken down due to inadequate funding to carry out maintenance and rehabilitation.

To reverse the decline in water supply services, various sector reforms and improvement projects were undertaken in 1970, 1981 and 1988 by the World Bank, IDA, donor countries and other external support agencies such as Austrian Government, Italian Government, Nordic Development Fund, the African Development Bank, CIDA, DFID, KfW, GTZ, OECF, ECGD and CFD/ADF.

Also in the mid-1990s, after the passing of a new democratic Constitution in 1992, the government of Ghana enacted five key laws that affected the responsibility for water supply and sanitation:

The Local Government Act No. 462 of 1993 defined the powers of District Assemblies as the highest political authority in each District, headed by a District Chief Executive (similar to a mayor). Neither the constitution nor the Local Government Act clearly defines the responsibilities of District Assemblies, and initially they had no role in water supply and sanitation until they were partially transferred to them in 1999.

The Water Resources Commission Act No. 552 of 1996 created the Water Resources Commission (WRC) that was made responsible for integrated water resources management including permits for water abstraction;

The Public Utilities and Regulatory Commission Act No. 538 of 1997 created the Public Utilities and Regulatory Commission (PURC) that was made responsible for the regulation of the electricity sector and urban water supply, including the review of requests for tariff increases, the monitoring of service quality and the protection of consumers;

The Community Water and Sanitation Agency Act No. 564 of 1998 created the Community Water and Sanitation Agency (CWSA) that became responsible for the support of local communities and District Assemblies in terms of rural water supply and sanitation. The CWSA emerged from the Community Water and Sanitation Division that had been created as a semi-autonomous division of GWSC in 1994.

Under the provisions of the Statutory Corporations (Conversion to Companies) Act 461 of 1993 and in line with a government policy to transform state corporations into companies operated on a commercial basis, GWSC was transformed into a company called GWCL, which remained responsible for urban water supply only, while responsibility for sewer systems was transferred to the District Assemblies.

Though some gains were derived from these interventions, their general impact on service delivery was very disappointing. Due to the failure of these interventions to achieve the needed results, several efforts were made to improve efficiency within the water supply sector in Ghana especially during the era of the Economic Recovery Programme from 1983 to 1993.

During this period, loans and grants were sought from the World Bank and other donors for rehabilitation and expansion programmes, training of personnel and procurement of transport and maintenance equipment.

In 1986, subvention for operations and maintenance was withdrawn although funding for development programmes continued. User fees for water supply were increased and subsidies on water tariffs were gradually removed for GWSC to achieve self-financing. The government at that time approved a formula for annual tariff adjustments to enable the Corporation generate sufficient funds to cover all annual recurrent costs as well as attain some capacity to undertake development projects.

For political reasons, this tariff formula was not applied and, over the years, irregular tariff increases were always below cost recovery levels resulting in heavy corporate deficit financing and ineffective service delivery.

In 1987, a "Five-Year Rehabilitation and Development Plan" for the sector was prepared which resulted in the launching of the Water Sector Restructuring Project (WSRP). Multilateral and bilateral donors contributed \$140 million to support the implementation of the WSRP.

The WSRP was aimed at reducing unaccounted for water, rationalisation of the workforce, hiring of professionals and training of staff. A strong focus of the WSRP was also on improved management and increased efficiency through organisational change of the water sector. Accordingly, a number of reforms within the Ghanaian water sector were initiated in the early 1990s.

As a first step, responsibilities for sanitation and small town water supply were decentralized and moved from Ghana Water and Sewerage Corporation to the District Assemblies in 1993.

The Environmental Protection Agency (EPA) was established in 1994 to ensure that water operations would not cause any harm to the environment.

The Water Resources Commission (WRC) was founded in 1996 to be in charge of overall regulation and management of water resources utilization.

In 1997, the Public Utilities Regulatory Commission (PURC) came into being with the purpose of setting tariffs and quality standards for the operation of public utilities. 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 standards concerning the provided services.

Community Water and Sanitation Agency (CWSA) was established in 1998 to be responsible for management of rural water supply systems, hygiene education and provision of sanitary facilities. After the establishment of CWSA, 120 water supply systems serving small towns and rural communities were transferred to the District Assemblies and Communities to manage under the community-ownership and management scheme.

Finally, pursuant to the Statutory Corporations (Conversion to Companies) Act 461 of 1993 as amended by LI 1648, on 1st July 1999, GWSC was converted into a 100% state owned limited liability, Ghana Water Company Limited, with the responsibility for urban water supply only.

1.3 Brief history about the school

St. Augustine's, Bogoso was an integral part of St. Augustine's college, Cape Coast, which offered simultaneously two different courses namely the secondary school course and the teacher Training course. In 1957, the Catholic Church thought it wise to separate the two schools. The Teacher Training section was thus uprooted from Cape Coast and transplanted at Bogoso. The college was sited at Marlu camp, a property bought by the Catholic Church from Marlu mining company.

In 1973, there was the need to phase out some training colleges in the country and converted them into secondary schools, St. Augustine's college happened to be one of them. With the introduction of the new educational reforms in January, 1991 which place emphasis on a three – year secondary education, the school became known as St. Augustine's senior secondary school (AUGUSCO).

1.4 Statement of the problem

Water is life, water is essential that you cannot live without it. The staff of St. Augustine's Senior High School- Bogoso who stay on campus find it difficult to get water as a result of pipelines not being connected to their houses. As the school was developing, the number of teachers and the students increased. This resulted in building of more infrastructures. This has necessitated the laying of new pipes in the school since the old ones have rusted and also pipes have not been laid to the new building and the teachers have been complaining for going a long distances before getting water for use . The installation of these pipelines to the bungalows is capital intensive. It has caused sleepless nights for the school administrators to connect pipelines to the various houses. The challenge is how to connect the pipes with a least cost. Therefore the focus of this thesis seeks to find the shortest-path these pipelines should be installed so as to reduce the cost of installation, minimize the number of pipelines needed.

1.5 Objectives

The study intends

i. To find the minimum connection of pipe network using Prim's algorithm.

ii. To model location of standpipe on the minimum connection network as 1 median problem.

iii. To find the optimal location of the standpipe using ReVelle and Swain algorithm

1.6 Methodology

Problem of pipe connection in St. Augustine's Senior High School- Bogoso will be modeled as a minimum connector problem and the problem of locating standpipe as 1 – median problem. The map of the network system of St. Augustine's Senior High School- Bogoso and the distances they cover was provided by school. Data collection was mainly from secondary source. The researcher used prim's algorithm and ReVelle and Swain algorithm to minimize the pipeline network system of the school and location of standpipes respectively and this will be done by coding on a matlab platform . The consulted information included journals, internet, Mathematical books from KNUST library Mathematics department, magazines articles and also previous research reports.

1.7 Justification

The school's population is increasing at a faster rate hence the demand for water is high. By coming out with a model, minimum connection of pipes lead to saving of cost. The school administrators can therefore use the amount left for other developmental project in the school.

The workers and students will be free from water borne diseases as they will get good water to drink.

The finding of this research will give a clear idea to the management on how to minimize pipe network and finding location point in general to enable better distribution of water from source to various destination points which will provide a good drinking water.

The study will also serve as a guide for further research guide in other area such as network design in transportation (rail lines, water pipelines etc) and rural and urban electrification in a town or country.

Finally, the study serves as a partial fulfillment of the requirement for the master degree in Industrial Mathematics

1.8 Organization of the study

The background study, statement of problem, objective, Justification of the study, scope of the study and organization of the study were covered in chapter one. Chapter two provides an overview of existing literature on networking. Chapter three discusses detailed methodology. This includes the general algorithms of the method of solutions of the problem. Chapter four looks at the analysis of the data using prim's algorithm and the discussion of the data. Chapter five includes the summary, conclusions, limitations and an outline of recommendations.

CHAPTER TWO

LITERATURE REVIEW

This chapter comprises a review of relevant literature on networking and shortest path algorithms.

Arogundade and Akinwale (2009) successfully used Prim's algorithm at a profit-oriented transportation systems in rural areas of Nigeria. The authors used Odeda Local Government map which has 88 villages connected by 96 roads. The project requires 388,270 meters to be covered. Their findings shows that given the cost of fuel for travelling over 8000 meters to be 70.00 Naira a transport company will make more profit with the implementation and use of Prim's algorithm; taking into an account the cost of travelling over 8000 meters by an averagely good Toyota 14 sector bus. They affirm that the company will make more profit with the implementation of the algorithm taking into account the cost of travelling over 8000 meters is estimated. In the course of their work, they discovered that the algorithm is very effective in providing shortest distances between two set of villages. In addition, Arogundade and Akinwale found that in one way it reduces the cost of fuel and time for transportation of passengers from one town to another which normally determines prices in Nigerian transportation.

Brimberg et al. (2003) studies the optimal design of an oil pipeline network for the South Gabon oil field. This oil field has 33 nodes representing onshore platforms, onshore wells, several connection points and one port (Gamba) with 129 possible arcs having total distance of 188.2 miles. Their finding in concert with Prim’s algorithm reduces the connection to 156.2 miles which is a reduction of 17% of the total distance to be covered. They reported that the government saved a substantial amount of money from the number of pipelines used, labour quantity, time and even minimize environmental destruction of the earth’s topography.

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

.

Gloor et al. (1993) described a system for visualizing correctness proofs of graph algorithms. The system has been demonstrated for a greedy algorithm, Prim's algorithm for finding a minimum spanning tree of an undirected, weighted graph. We believe that our system is particularly appropriate for greedy algorithms, though much of what we discuss can guide visualization of proofs in other contexts. While an example is not a proof, our system provides concrete examples to illustrate the operation of the algorithm. These examples can be referred to by the user interactively and alternatively with the visualization of the proof where the general case is portrayed abstractly.

Goldberg et al. (2001) proposed shortest path algorithms that use a search in combination with a new graph-theoretic lower-bounding technique based on landmarks and the triangle inequality. Our algorithms compute optimal shortest paths and work on any directed graph. We give experimental results showing that the most efficient of our new algorithms outperforms previous algorithms, in particular a search with Euclidean bounds, by a wide margin on road networks and on some synthetic problem families.

Dijkstra (1959) submitted his short paper for publication in Numerische Mathematik June 1959. In November 1959, Pollack and Wiebenson [1960] submitted a paper entitled Solutions of the shortest-route problem - a review to the journal Operations Research. This review briefly discusses and compares seven methods for solving the shortest path problem.
However, the review presents a 'highly efficient' method, attributed - as a dateless private communication - to Minty. The procedure is defined as follows (Pollack and Wieberson [1960, p. 225]: The objective is to find the shortest path from city A to city B).

Pangilinan et al. (2007) presented an overview of the multi-objective shortest path problem (MSPP) and a view essential and resent issue 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 optimally of solutions. Results show that the evolutionary algorithm can find diverse solutions to the MSPP in polynomial time (based on several networks instances) and can be an alternative when other methods are trapped by the tractability problem.

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

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 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.

Buriol et al. (2008) described dynamic shortest-path algorithms update 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. For unit weight changes, the updates can be done without heaps. These reductions almost, always reduce the computational times for these algorithms. In computational testing, several dynamic shortest-path algorithms with and without the heap-reduction technique are compared. Speedups of up to a factor of1.8 were observed using the heap-reduction technique on random weight changes and of over a factor of five on unit weight changes. We compare as well with

Dijkstra's algorithm, which recomputed the paths from scratch. With respect to Dijkstra's algorithm, speedups of up to five orders of magnitude are observed.

Chen et al. (2007) focused on the optimization problems about complicated network, this paper presents an algorithm KSPA to solve the K-shortest paths problem in complicated network based on algorithm, in which the time cost is taken as target function and the establishment of the target function model is given. Experimental results show the proposed KSPA maintains an excellent efficiency on certain public traffic data. It can be used to solve the K-shortest paths problems in multi-graph.

Venkataraman et al. (2003) proposed a block version of Floyd’s all-pairs shortest-paths algorithm. The blocked algorithm makes utilization of cache than does Floyd’s original algorithm. Experiments indicate that the blocked algorithm delivers a speedup (relative to the unlocked Floyds algorithm) between 1.6 and 1.9 on a Sun Ultra Enterprise 4000/5000 for graphs that have between 240 and 1200 vertices is between 1.6 and 2.0.

McCarthy et al. (2009) presented the application of two well known graph algorithms, Edmonds‟ algorithm and Prim’s algorithm, to the problem of optimizing distributed SPARQL queries. In the context of this paper, resolved by contacting any number of remote SPARQL endpoints. Two optimization approaches are described. In the first approach, a static query plan is computed in advance of query executions, using one of two standard graph algorithms for finding minimum spanning trees (Edmond’s and Prim’s algorithm). In the second approach, the planning and the execution of the query are interleaved, so that as each potential solution is expanded it is permitted to an independent query plan. Our optimization approach requires basic statistic regarding RDF predicates which must be obtained prior to the user's query, through automated querying of the remote SPARQL endpoints.

Divoky (1990) presented a framework for solving the shortest-path, cost-flow problem with positive edge weights can be implemented by itself or as a subordinate process in a solution procedure for bigger problems. The efficiency of such shortest-path frameworks depend on the technique employed to use the structure of the network from the solution of the bigger problem, as well as on the efficiency of the frameworks. The topology of the networks is characterized by arcs with positive weights and of sub-networks called zero-weight components. The techniques for using topology include; identifying the zero-weight edges and adding them to the graph; identifying and adding the basic sub-trees to the shortest-path tree; and interrupting the shortest-path frameworks scanning or fixing the labels of all the nodes in the sub-tree.

Maranzana (1964) describes aheuristic that randomly locates the *p* facilities and then solves the allocation problem (which has apolynomial 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.

ReVelle and Swain (1970) addressed the problem they call "central facility location", consisting of designating *m* of *n* communities in a geographical region as centers, so that the average time or distance travelled by people to go to these centers is minimum. They also suggest that the formulation they use is applicable to the case in which the facilities are supply points from where good emanate to the communities.

Hua Lo-Keng and others (1962) proposed an algorithm for locating the 1-median on trees (and networks with cycles), and proved that locating medians points on vertices was better than locating them somewhere along the edges. Their paper was intended for practitioners who need to set up threshing floors for dispersed wheat fields, so it did not include much mathematical insight.

Goldman (1969) generalized the *p*-median, defining a problem in which commodities are transported over a path between origin and destination nodes, and the total transportation cost is minimized. The path goes through one or two medians.

Holmes et al. (1972) introduced two interesting generalizations of the *p*-median. The first generalization is elasticity of the demand, i.e. a situation in which customers lose interest in the service or goods if these are located beyond a threshold distance. This generalization is useful in the case of non-essential goods or services. The second generalization considers a constraint on the capacity of the facilities. This constraint makes the problem much harder, since it leads to the appearance of many fractional-valued location and allocation variables in the solution, if the integer programming problem is solved in a linearly relaxed version.

Berman and Larson (1982) formulate the Stochastic Queue Median, which locates a single facility operating as a M/G/1 queue, on any point of a network.

Wesolowsky and Truscott (1975) introduce the multiperiod *p*-median problem, in which the facilities are relocated in response to predicted changes in demand, considering that relocating facilities has a cost

Hillsman and Rhoda (1978) identified three classes of aggregation errors: source A, B and C errors. Source A errors arise due to the approximation of the actual values of distance; source B errors are a particular case, that occurs when a demand point coincides with a candidate location and the distance between the demand and the potential facility is considered equal to zero; and source C errors correspond to an incorrect assignment of the demands to facilities.

Kariv and Hakimi (1979) proved that the general *p*-median, where p is a variable, is NP-hard even in the case of a planar network of maximum vertex degree 3, with vertices of weight 1 and edges of length 1. Also, they rediscovered Hua-Lo Keng and others (1962) algorithm for locating one median on a network, and proposed an $O(n^2p^2)$ algorithm to find more than one median on trees.

Pettie et al. (2002) evaluated the practical efficiency of a new shortest path algorithm for undirected graphs which was developed by the two authors. This algorithm works on the fundamental comparison-addition model. Theoretically, this new algorithm out-performs Dijkstra"s algorithm on sparse graphs for the all-pairs shortest path-problem, and more generally, for the problem of computing single –source shortest path from different sources. Our extensive experimental analysis demonstrated that this is also the case in practice. The authors presented results which showed the new algorithm to run faster than Dijkstra's on a variety of sparse graphs when the number of vertices ranges from a few thousand to a few million, and when computing single-source shortest paths from a few as three different sources.

CHAPTER THREE

METHODOLOGY

This chapter focuses on the method and algorithms for solving the problem of laying pipelines network.

3.1 DEFINITION OF TERMS:

A graph is an ordered pair G = (V, E) comprising a set V of vertices or nodes together with a set E of edges or lines, which are 2-element subsets of V (i.e., an edge is related with two vertices, and the relation is represented as unordered pair of the vertices with respect to the particular edge). To avoid ambiguity, this type of graph may be described precisely as undirected and simple.

A path in a graph is a sequence of vertices connected by edges. A simple path is one with no repeated vertices. The order of a graph is (the number of vertices). A graph's size is the number of edges. The degree of a vertex is the number of edges that connect to it, where an edge that connects to the vertex at both ends (a loop) is counted twice.

A graph can also be defined as a representation of a set of objects where some pairs of objects are connected by links. The interconnected objects are represented by mathematical abstractions called vertices, and the links that connect some pairs of vertices are called edges. Typically, a graph is depicted in diagrammatic form as a set of dots for the vertices, joined by lines or curves for the edges.

The edges may be directed or undirected. For example, if the vertices represent people at a party, and there is an edge between two people if they shake hands, then this is an undirected graph, because if person A shook hands with person B, then person B also shook hands with person A. In contrast, if there is an edge from person A to person B when person A knows of person B, then this graph is directed,

Directed graph (or digraph) is a graph, or set of nodes connected by edges, where the edges have a direction associated with them. In formal terms, a digraph is a pair G = (V, A)

- a set V, whose elements are called vertices or nodes,
- a set A of ordered pairs of vertices, called arcs, directed edges, or arrows (and sometimes simply edges with the corresponding set named E instead of A).

  A directed graph is called weakly connected if replacing all of its directed edges with undirected edges produces a connected (undirected) graph. It is strongly connected or strong if it contains a directed path from u to v and a directed path from v to u for every pair of vertices u, v.

A directed graph G is called symmetric if, for every arc that belongs to G, the corresponding reversed arc also belongs to G. Figure 3.1 shows the example of a directed graph

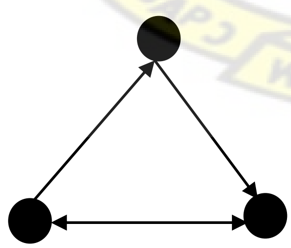


Figure 3.1: example of a directed Graph

An undirected graph is one in which edges have no orientation. The edge (a, b) is identical to the edge (b, a), i.e., they are not ordered pairs, but sets {u, v} (or 2-multisets) of vertices. An undirected graph G consists of a set V of vertices and a multi set of edges E unordered pairs of vertices. We usually write G = (V, E) is called undirected graph. Maximum number of edges in an undirected graph without self loop is n(n-1)/2

In an undirected graph G, two vertices u and v are called connected if G contains a path from u to v. Otherwise, they are called disconnected. A graph is called connected if every pair of distinct vertices in the graph is connected; otherwise, it is called disconnected.

A weighted digraph is a digraph with weights assigned to its arcs. In the context of graph theory a digraph with weighted edges is called a **network**.

An Eulerian graph is a connected graph which has a closed trail containing every arc precisely once. This can occur if and only if every node is even.

A **tree** is a cycle- free network which comprises of a subset of all the nodes.
A **spanning tree** of an undirected graph of n nodes is a set of n-1 edges that connects all nodes.
In the mathematical field of graph theory, a spanning tree T of a connected, undirected graph G is a tree composed of all the vertices and some (or perhaps all) of the edges of G. Informally, a spanning tree of G is a selection of edges of G that form a tree spanning every vertex. That is, every vertex lies in the tree, but no cycles (or loops) are formed. On the other hand, every bridge of G must belong to T.

A spanning tree of a connected graph G can also be defined as a maximal set of edges of G that contains no cycle, or as a minimal set of edges that connect all vertices.

A minimum spanning tree of weighted graph is a collection of edges connecting all of the vertices such that the sum of the weights of the edges is at least as small as sum of the weights of any other collection of edges connecting all of the vertices

One example would be a telecommunications company laying cable to a new neighborhood. If it is constrained to bury the cable only along certain paths, then there would be a graph representing which points are connected by those paths. Some of those paths might be more expensive, because they are longer, or require the cable to be buried deeper; these paths would be represented by edges with larger weights. A spanning tree for that graph would be a subset of those paths that has no cycles but still connects to every house. There might be several spanning trees possible. A minimum spanning tree would be one with the lowest total cost.

3.2 Minimum Connector Formulation

Given a connected undirected graph G = (V, E), with the node set V, the edge set E, weight ω_e for all edge in E and x_e for all edge in E_T and $x_e = \begin{cases} 1 & if\ edge\ e \in E_T \\ 0 & otherwise \end{cases}$.

Find a spanning tree $G_T = (V_T, E_T)$ of minimum total weight with the node set V_T and the edge set E_T. The problem can be model as

$$\min \sum_{e \in E} \omega_e x_e$$

$$s.t \quad \sum_{e \in E} x_e = n - 1$$

$$\sum_{e \in (S,S)} x_e \leq |S| - 1 \quad \forall S \subseteq V \qquad (1c)$$

$$x_e \in \{1, 0\} \quad \forall e \subseteq E$$

$$x_e = \begin{cases} 1 & if\ edge\ e \in E_T \\ 0 & otherwise \end{cases}$$

Where

(S, S) = all edges that go from a locality in the set S to another locality in the set S,

V = the set of nodes

E = the set of edges

x_e = edges of the minimum spanning tree

$$x_e = \begin{cases} 1 & if\ edge\ e \in E_T \\ 0 & otherwise \end{cases}$$

$\omega_e x_e$ = weight of the edges of the minimum spanning tree o

n=number of edges

Where (S, S) denotes all edges that go from a node in the set S to another node in the set S.

Equation (1c) enforces the constraint that the edges in E_T can't form cycles.

3.3 .0 METHODS OF SOLVING MINIMUM CONNECTOR PROBLEM

3.3.1 Kruskal's algorithm

The Kruskal Algorithm starts with a forest which consists of n trees. Each and every one tree consists only by one node and nothing else. In every step of the algorithm, two different trees of this forest are connected to a bigger tree. Therefore, we keep having less and bigger trees in our forest until we end up in a tree which is the minimum genetic tree (m.g.t.) .In every step we choose the side with the least cost, which means that we are still under greedy policy. If the chosen side connects nodes which belong to the same tree the side is rejected, and not examined again because it could produce a circle which will destroy our tree .Either this side or the next one in order of least cost will connect nodes of different trees, and this we insert connecting two small trees into a bigger one. . If an edge (u, v) connects two different trees, then (u, v) is added to the set of edges of the MST, and two trees connected by an edge (u, v) are merged into a single tree on the other hand, if an edge (u, v) connects two vertices in the same tree, then edge (u, v) is discarded

A little more formally, given a connected, undirected, weighted graph with a function $w : E \to R$.

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

This minimum spanning tree algorithm was first described by Kruskal in 1956 in the same paper where he rediscovered Jarnik's algorithm. This algorithm was also rediscovered in 1957 by Loberman and Weinberger, but somehow avoided being renamed after them.

Algorithm of kruskal can be stated as follow

Step 1 Choose the unused edge with the lowest value.

Step 2 Add this edge to your tree.

Step 3 If there are $n - 1$ edges in your tree, stop. If not, go to Step 1.

NB : At each step remember to make sure you do not make a cycle

Worked example

Use Kruskal's algorithm to find the minimum spanning tree for the following network.

Figure 3.2 represents roads network between towns.

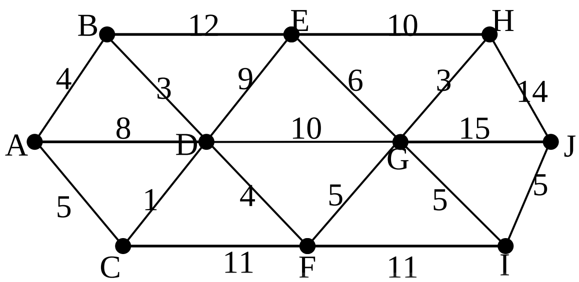


Figure 3.2 roads network of towns

Solution

There are 10 vertices so the minimum spanning tree will have nine edges.

Step 1 Choose the lowest edge CD, a value of 1.

Step 2 Add CD to the tree.

Figure 3.2.1 shows the edge of towns C and D

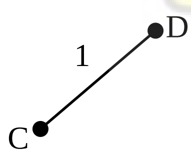


Figure 3.2.1: edge of towns C and D

Step 1 Choose the lowest edge BD or GH, both have a value of 3. It doesn't matter which is chosen.

Step 2 Add BD to the tree.

Figure 3.2.2 represents the roads network of towns B, D and C

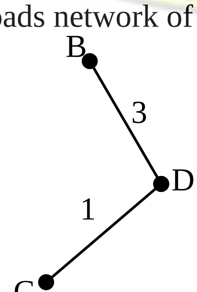


Figure 3.2.2: roads network of towns B, D and C

Step 1 Choose the lowest edge GH, with a value of 3.

Step 2 Add GH to the tree.

Figure 3.2.3 represents roads network of towns B, D, C, G and H

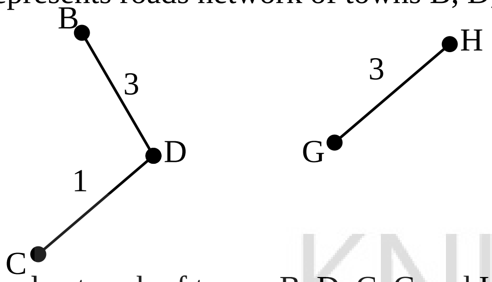


Figure 3.2.3 road network of towns B, D, C, G and H

Step 1 Choose the lowest edge AB or DF, both have a value of 4.

Step 2 Add AB to the tree.

Figure 3.2.4 represents road network of towns A, B, D, C, G and H

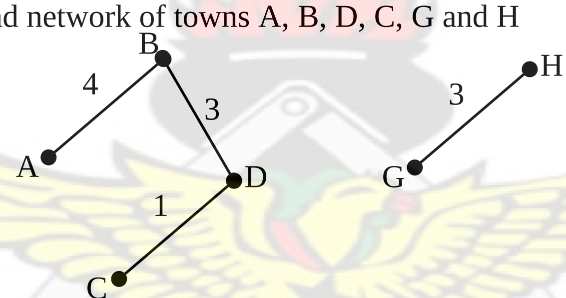


Figure 3.2.4 road network of towns A, B, D, C, G and H

Step 1 Choose the lowest edge DF, with a value of 4.

Step 2 Add DF to the tree.

Figure 3.2.5 represents road network of towns A, B, C, D, F, G and H

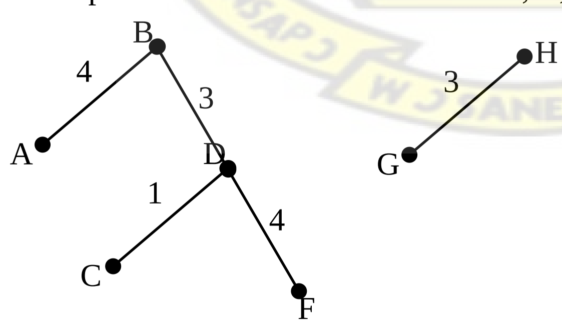


Figure 3.2.5 road network of towns A, B, C, D, F, G and H

Figure 3.2.6 represents the complete minimum spanning of the road network of the towns

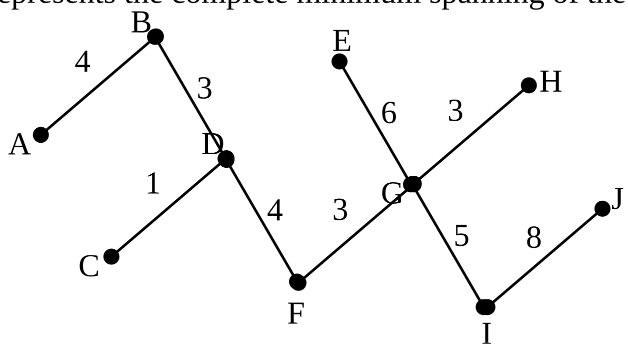


Figure 3.2.6: The complete minimum spanning of the road network of the towns

The length of the minimum spanning tree is 37 units.

3.3.3 Prim's algorithm

Given is an undirected graph G= (*V*, *E*), where V is the set of nodes and E is the set of edges. Each edge is defined by a node pair (a, b), each edge has a non-negative length. Then the prim's algorithm works as follows:

Step 1 Create a tree containing a single vertex, chosen arbitrarily from the graph

Step 2 Create a set containing all the edges in the graph

Step 3 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

Thus Prim's algorithm works from a starting point and builds up the spanning tree step by step, connecting edges into the existing solution. 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 can be applied directly to the distance matrix, as well as to the network itself.

The algorithm applying to the network itself can be stated as follows:

Step 1: choose a starting vertex

Step 2: From the starting vertex draw the lowest valued edge to start your tree. (Any vertex can be chosen as the start vertex; however, it will always be given in an exam question.)

Step 3: From any vertex on your tree, add the edge with the lowest value.

Step 4: If there $n - 1$ edges in your tree, you have finished. If not, go to step 3.

NB : At each step remember to make sure you do not make a cycle.

A key aspect of Prim's algorithm is that as the tree is being built, the tree is always connected. You are always adding a new vertex to your current tree.

Illustrative example

Use Prim's algorithm, starting from A, to find the minimum spanning tree for the network below.

Figure 3.3 represents road network between towns.

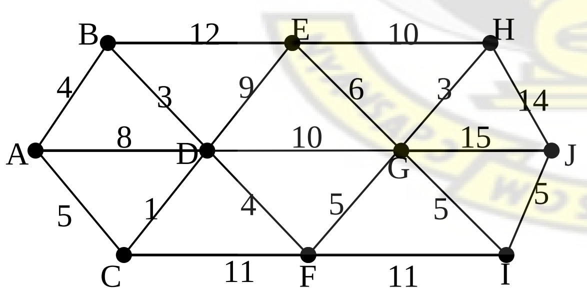


Figure 3.3 road networks of towns

Solution

Step 1 Choose the lowest edge from A, which is AB, with a value of 4.

Figure 3.3.1 represents roads network between towns A and B.

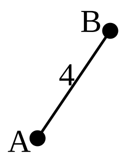


Figure 3.3.1 roads network between towns A and B.

Step 2 Choose the lowest edge from A or B, which is BD, with a value of 3. Add BD to the tree.

Figure 3.3.2 represents road network of towns AB and BD.

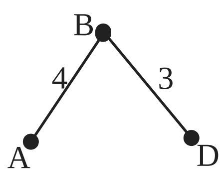


Figure 3.3.2: roads network of towns AB and BD

Choose the lowest edge from A, B or D, which is DC, with a value of 1. Add DC to the tree.

Figure 3.3.3: shows roads network of towns A, B, D and C

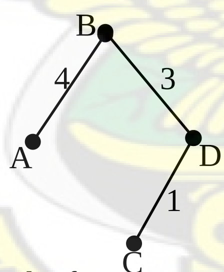


Figure 3.3.3: roads network of towns A, B, D and C

Choose the lowest edge from A, B, C or D, which is DF, with a value of 4. Add DF to the tree.

Figure 3.3.4: shows roads network of towns A, B, D, C, D and F

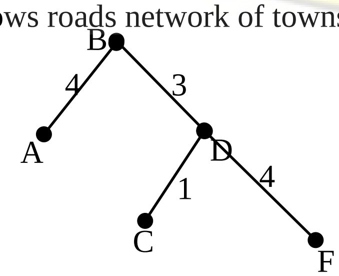


Figure 3.3.4: roads network of towns A, B, D, C and F

Step 2 Choose the lowest edge from A, B, C, D or F, which is either

AC or FG, with a value of 5. **DO NOT ADD** AC to the tree as this would make a cycle of ADCA. Add FG to the tree.

Figure 3.3.5: shows road network of towns A, B, D, C, F and G

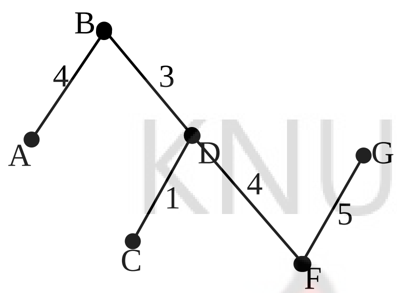


Figure 3.3.5: shows road network of towns A, B, D, C, F and G

Figure 3.3.6: shows minimum spanning tree of the road network of the towns

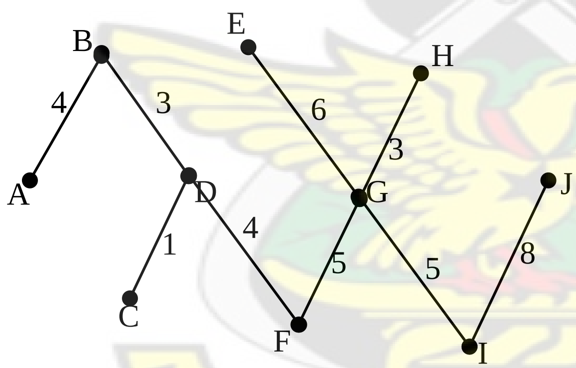


Figure 3.3.6: shows minimum spanning tree of the road network of the towns

There are now $n - 1$ (which equals 9, in this case) edges. The minimum spanning tree is your final answer. The length of the minimum spanning tree is 39 units.

Apart from Prim's algorithm being applied directly to the network itself, it can also be applied directly to the distance matrix.

Matrix formulation of Prim's algorithm

- Select any node to be the first node of T
- Circle the new node of T in the top row, and cross out the row corresponding to this new node.
- Find the smallest weight left in the columns with circled headings. Circle this weight. Then choose the node whose weight the row is in to join T.
- Repeat until T contains every node.

Illustrative example

It is required to install a computer network linking up the offices of the head quarters of a commercial organization.

The network drawn below shows the costs associated with various runs of cable represented by arcs in the diagram. What is the minimum cost of the cabling for the computer network?

Figure 3.4 shows the cost of cabling a computer network

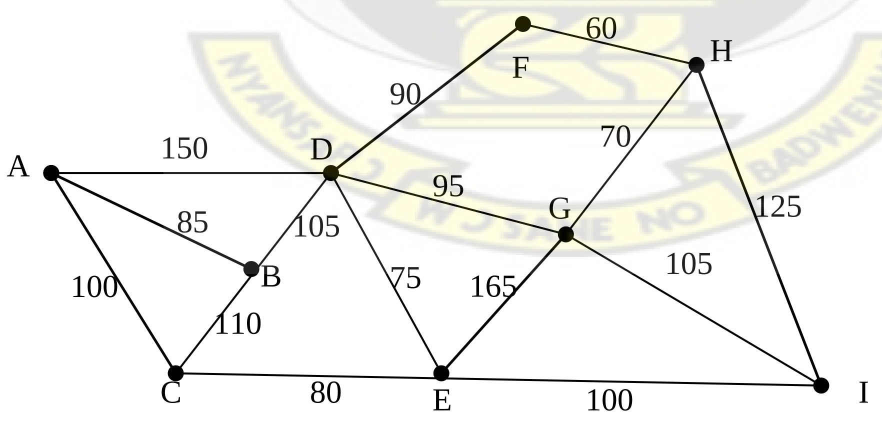


Figure 3.4 network of the cost of cabling a computer network

Table 3.1 shows the n x n matrix of the cost of cabling a computer network

Table 3.1 n x n matrix of the cost of cabling a computer network

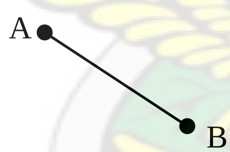
	A	B	C	D	E	F	G	H	I
~~**A**~~	~~∞~~	~~85~~	~~100~~	~~150~~	~~∞~~	~~∞~~	~~∞~~	~~∞~~	~~∞~~
B	85	∞	110	105	∞	∞	∞	∞	∞
C	100	110	∞	∞	80	∞	∞	∞	∞
D	150	105	∞	∞	75	90	95	∞	∞
E	∞	∞	80	75	∞	∞	165	∞	100
F	∞	∞	∞	90	∞	∞	∞	60	∞
G	∞	∞	∞	95	165	∞	∞	70	105
H	∞	∞	∞	∞	∞	60	70	∞	125
I	∞	∞	∞	∞	100	∞	105	125	∞

AB is the smallest edge joining A to the other vertices. Look for the smallest entry in columns A and B.

Table 3.1.1 shows the edge AB being selected.

Table 3.1.1 the edge A and B.

	A	B	C	D	E	F	G	H	I
~~**B**~~	~~85~~	~~∞~~	~~110~~	~~105~~	~~∞~~	~~∞~~	~~∞~~	~~∞~~	~~∞~~
C	100	110	∞	∞	80	∞	∞	∞	∞
D	150	105	∞	∞	75	90	95	∞	∞
E	∞	∞	80	75	∞	∞	165	∞	100
F	∞	∞	∞	9O	∞	∞	∞	60	∞
G	∞	∞	∞	95	165	∞	∞	70	105
H	∞	∞	∞	∞	∞	60	70	∞	125
I	∞	∞	∞	∞	100	∞	105	125	∞

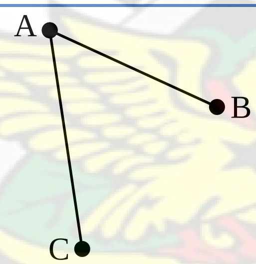


AC is the smallest edge joining A and B to the other vertices. Put edge AC into the solution. Look for the smallest entry in columns A, B and C

Table 3.1.2 represents the edges AB and AC being selected.

Table 3.1.2 the edges AB and AC

	A	B	C	D	E	F	G	H	I
~~**C**~~	~~100~~	~~110~~	~~∞~~	~~∞~~	~~80~~	~~∞~~	~~∞~~	~~∞~~	~~∞~~
D	150	105	∞	∞	75	90	95	∞	∞
E	∞	∞	80	75	∞	∞	165	∞	100
F	∞	∞	∞	9O	∞	∞	∞	60	∞
G	∞	∞	∞	95	165	∞	∞	70	105
H	∞	∞	∞	∞	∞	60	70	∞	125
I	∞	∞	∞	∞	100	∞	105	125	∞

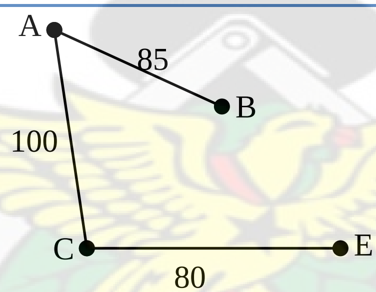


CE is the smallest edge joining A, B and C to the other vertices. Put CE into the solution. Look for smallest entry in the column A, B, C and E.

Table 3.1.3 represents the edges AB, BC and CE being selected

Table 3.1.3 the edges of AB, AC and CE

	A	B	C	D	E	F	G	H	I
D	150	105	∞	∞	75	90	95	∞	∞
~~**E**~~	~~∞~~	~~∞~~	~~80~~	~~75~~	~~∞~~	~~∞~~	~~165~~	~~∞~~	~~100~~
F	∞	∞	∞	9O	∞	∞	∞	60	∞
G	∞	∞	∞	95	165	∞	∞	70	105
H	∞	∞	∞	∞	∞	60	70	∞	125
I	∞	∞	∞	∞	100	∞	105	125	∞



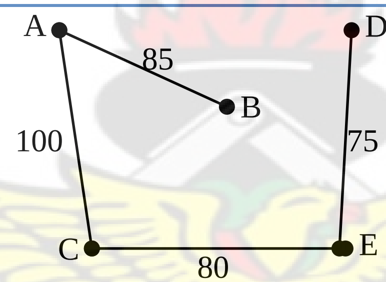
ED is the smallest edge joining A, B, C and E to the other vertices. Put ED into the solution.

Look for smallest entry in the column A, B, C E and D.

Table 3.1.4 represents the edges AB, CC, DE and CE being selected

Table 3.1.4 the edges of AB, CC, DE and CE

	A	B	C	D	E	F	G	H	I
D	150	105	∞	∞	75	90	95	∞	∞
F	∞	∞	∞	9O	∞	∞	∞	60	∞
G	∞	∞	∞	95	165	∞	∞	70	105
H	∞	∞	∞	∞	∞	60	70	∞	125
I	∞	∞	∞	∞	100	∞	105	125	∞



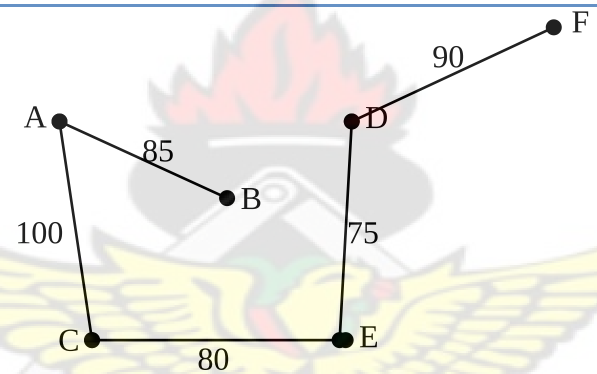
DF is the smallest edge joining A, B, C, E and D to the other vertices. Put DF into the solution.

Look for smallest entry in the column A, B, C, E, D and F

Table 3.1.5 represents the edges AB, CC, DE, CE and DF being selected

Table 3.1.5 the edges of AB, CC, DE, CE and DF

	A	B	C	D	E	F	G	H	I
~~**F**~~	~~∞~~	~~∞~~	~~∞~~	~~9O~~	~~∞~~	~~∞~~	~~∞~~	~~60~~	~~∞~~
G	∞	∞	∞	95	165	∞	∞	70	105
H	∞	∞	∞	∞	∞	60	70	∞	125
I	∞	∞	∞	∞	100	∞	105	125	∞



EI is the smallest edge joining A, B, C, E, D, F, G and H to the other vertices. Put EI into the solution. Look for smallest entry in the column A, B, C, E, D, F, H, G and I

	A	B	C	D	E	F	G	H	I
~~**I**~~	~~∞~~	~~∞~~	~~∞~~	~~∞~~	~~100~~	~~∞~~	~~105~~	~~125~~	~~∞~~

Figure 3.4.1 shows the minimum spanning tree of the cost of cabling the computer network of the offices

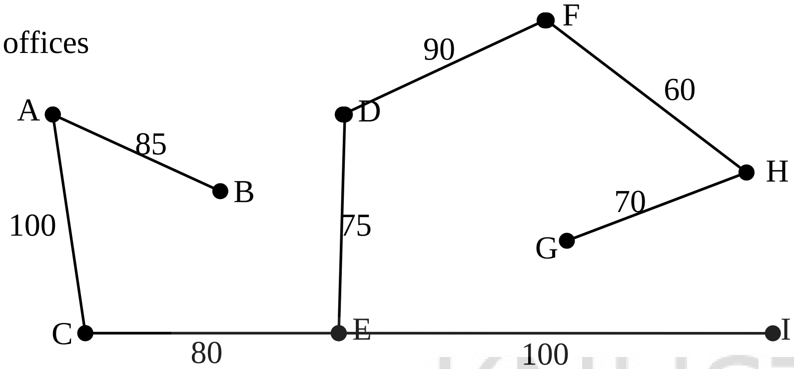


Figure 3.4.1 minimum spanning tree of the cost of cabling the computer network of the offices

Minimum cost = 85 + 100 + 80 + 75 + 90 + 60 + 70 + 10

= Gh¢660

3.3.3 Reverse – delete algorithm

The reverse-delete algorithm is an algorithm in graph theory used to obtain a minimum spanning tree from a given connected, edge-weighted graph. It first appeared in Kruskal (1956), but it should not be confused with Kruskal's algorithm which appears in the same paper. If the graph is disconnected, this algorithm will find a minimum spanning tree for each disconnected part of the graph. The set of these minimum spanning trees is called a minimum spanning forest, which contains every vertex in the graph.

This algorithm is a greedy algorithm, choosing the best choice given any situation. It is the reverse of Kruskal's algorithm, which is another greedy algorithm to find a minimum spanning tree. Kruskal’s algorithm starts with an empty graph and adds edges while the Reverse-Delete algorithm starts with the original graph and deletes edges from it.

The algorithm works as follows:

- Start with graph G, which contains a list of edges E.
- Go through E in decreasing order of edge weights.
- For each edge, check if deleting the edge will further disconnect the graph.
- Perform any deletion that does not lead to additional disconnection

Illustrative example

Figure 3.5shows the road network of towns

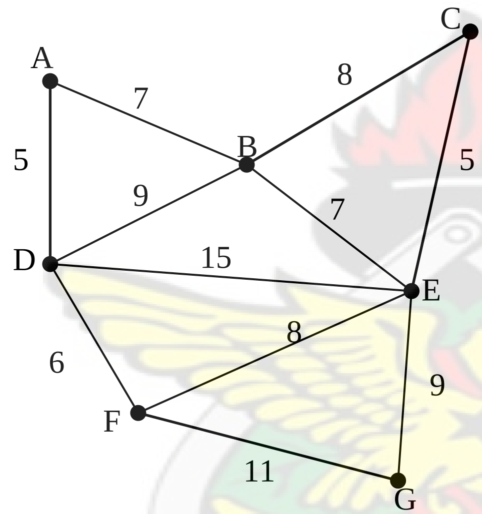


Figure 3.5 road networks of towns

The algorithm will start with the maximum weighted edge, which in this case is DE with an edge weight of 15. Since deleting edge DE does not further disconnect the graph it is deleted.

Figure 3.5.2 shows the next maximum weighted edges of the road network of towns being deleted

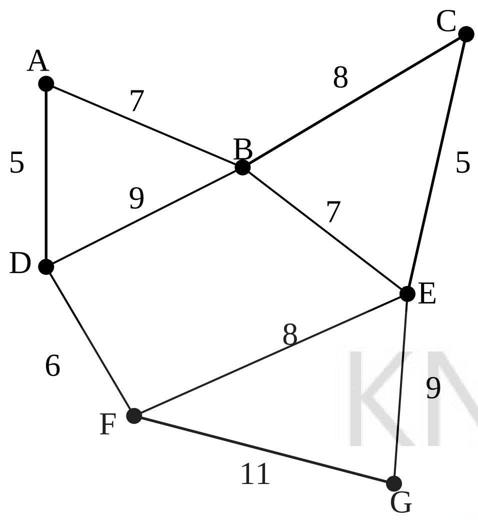


Figure 3.5.2 deleted maximum weighted edges of the road network of towns

The next largest edge is FG so the algorithm will check if deleting this edge will further disconnect the graph. Since deleting the edge will not further disconnect the graph, the edge is then deleted.

Figure 3.5.3 shows the next maximum weighted edges of the road network of towns being deleted

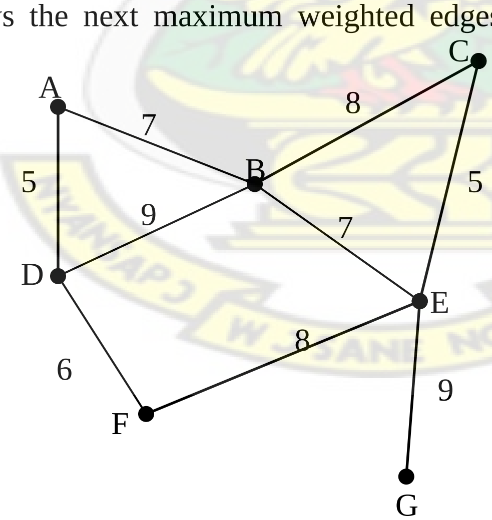


Figure 3.5.3 deleted maximum weighted edges of the road network of towns

Figure 3.5.4 shows the minimum spanning tree of the road network of the towns

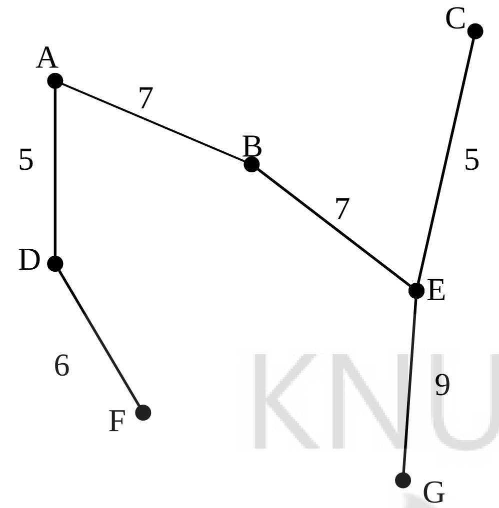


Figure 3.5.4 minimum spanning tree of the road network of the towns

The algorithm will then search the remaining edges and will not find another edge to delete; therefore this is the final graph returned by the algorithm.

Total distance is = 6 + 5 + 7 + 7 + 9 + 5 = 37

3.4.0 METHODS OF SOLVING SHORTEST PATH PROBLEMS

3.4.1 SINGLE SOURCE PATHS, NON-NEGATIVE WEIGHTS (DIJKSTRA'S ALGORITHM)

Dijkstra's algorithm, conceived by Dutch computer scientist Edsger Dijkstra in 1959, is a graph search algorithm that solves the single-source shortest path problem from a graph with nonnegative edge path costs, producing a shortest path tree.

The algorithm finds the shortest paths from a source nodes S to all other nodes in a network with non – negative arc lengths. Dijkstra's algorithm maintains a distance label d(i) with each node i, which is ad upper bound on the shortest path length from the source node 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 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 distances from the node S. Initially, node S is assigned a permanent label of zero, and each 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 select a node i with minimum temporary label (breaking ties arbitrarily), makes it permanent, and reaches out from that 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 node permanent.

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

Step 1 : assign the permanent label O to the starting vertex.

Step 2 : assign temporary labels to all the vertices that are connected directly to the most recently permanent labeled vertex.

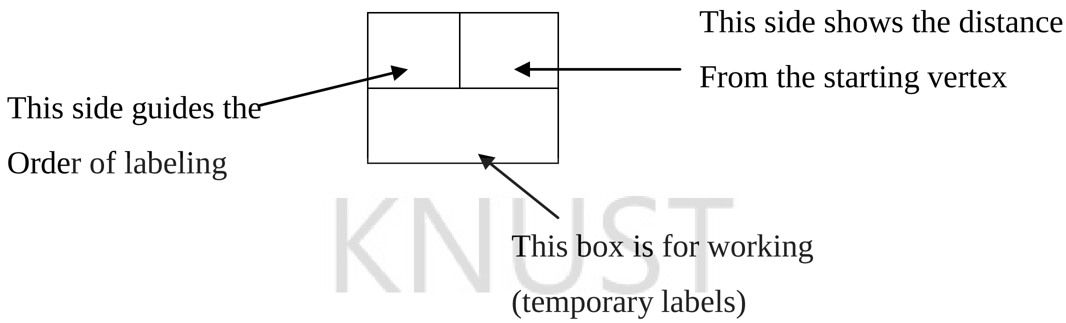
Step 3: Choose the vertex with the smallest temporary label and assign a permanent labeled vertex.

Step 4: Repeat Steps 2 and 3 until all vertices have permanent labels.

Step 5: 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.

Illustrative example

Figure 3.6 shows the road network of eight towns

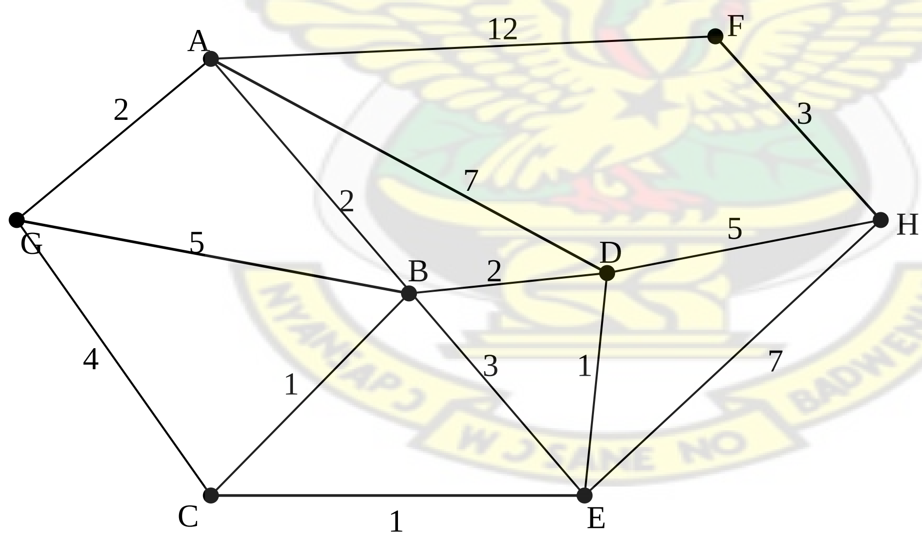


Figure 3.6 road networks of eight towns

Figure 3.6.1 shows the starting vertex G and it vertices that are directly connected to it

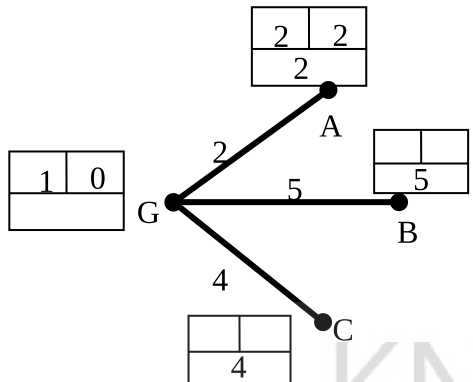


Figure 3.6.1 vertex G and it directly connected vertices

Figure 3.6.2 shows permanent vertices G and A and their temporary vertices that are directly connected to them

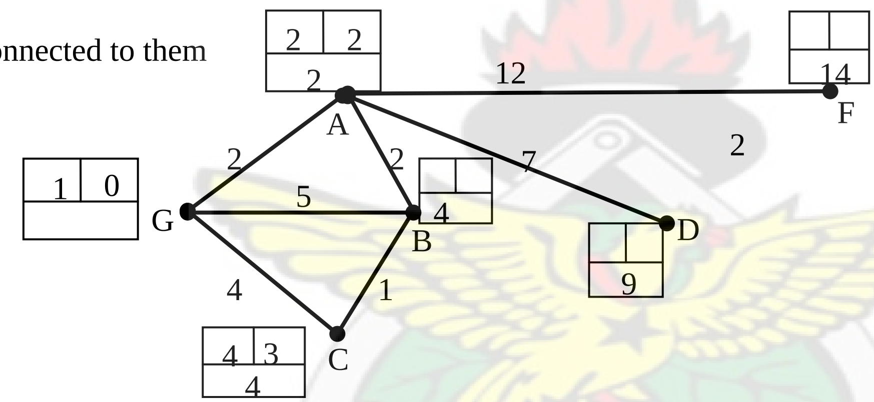


Figure 3.6.2 permanent vertices G and A and their directly connected vertices

Figure 3.6.3 represents permanent vertices G, A and C their temporary vertices that are directly connected to them

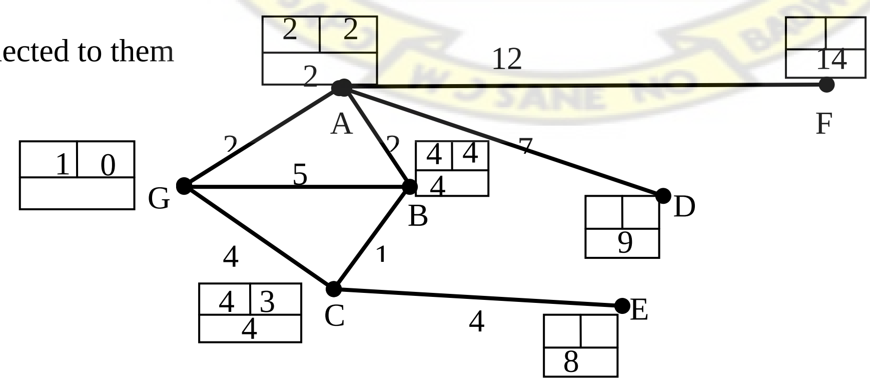


Figure 3.6.3 permanent vertices G and A and C their directly connected vertices

Figure 3.6.4 represents permanent vertices G, A, C and B and temporary their temporary vertices

that are directly connected

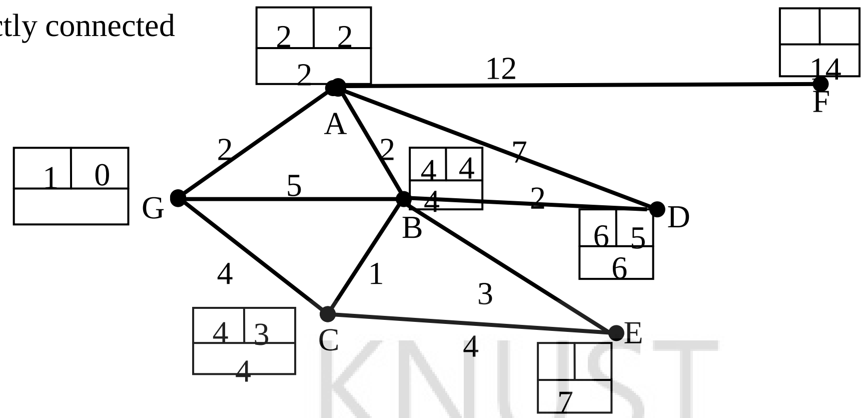


Figure 3.6.4 permanent vertices G, A, C and B and their directly connected vertices

Figure 3.6.5 shows all the permanent vertices

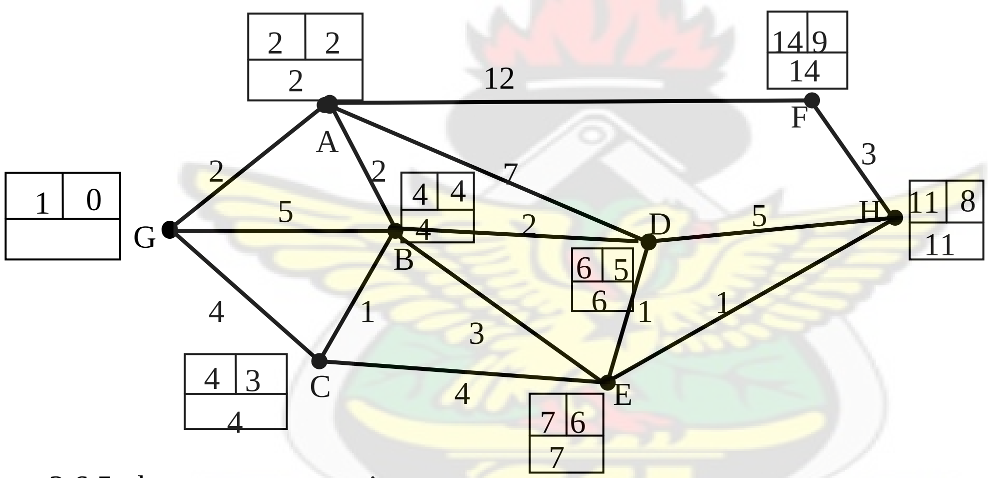


Figure 3.6.5 the permanent vertices

Now all the vertices have permanent labels and we can see that the shortest distance from G to H is 11.

We find the shortest path by working backward from H, ie. H → D → B→A→G.

Hence the shortest path from G to H is G→ A → B → D→H with length 11.

3.5.0 ALL – PAIRS SHORTEST PATH PROBLEM

3.5.1 FLOYD – WARSHALL ALGORITHM

In computer science, the Floyd–Warshall algorithm (also known as Floyd's algorithm, Roy–Warshall algorithm, Roy–Floyd algorithm, or the WFI algorithm) is a graph analysis algorithm for finding shortest paths in a weighted graph with positive or negative edge weights (but with no negative cycles,) and also for finding transitive closure of a relation R. A single execution of the algorithm will find the lengths (summed weights) of the shortest paths between all pairs of vertices, though it does not return details of the paths themselves. The algorithm is an example of dynamic programming. It was published in its currently recognized form by Robert Floyd in 1962. However, it is essentially the same as algorithms previously published by Bernard Roy in 1959 and also by Stephen Warshall in 1962 for finding the transitive closure of a graph. The modern formulation of Warshall's algorithm as three nested for-loops was first described by Peter Ingerman, also in 1962.

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 nodes i and j. It repeats the process until it obtains $d^{n+1}(i, j)$ for all node pairs i and j when it terminates. Given $d^k(i, j)$, the algorithm computes $d^{k+1}(i, j) = \min\{ d^k(i, k) , d^k(i, j)$ 1. The Floyd-Warshall algorithm remains of interest because it handles negative weight edges correctly.

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.

Illustrative example

Table 3.2 represents the matrix of the distances (in km) of direct roads between six towns

Table 3.2 Matrix of the distances (in km) of direct roads between six towns

	A	**B**	**C**	**D**	**E**	**F**
A	-	12	3	∞	7	10
B	12	-	6	6	18	8
C	3	6	-	4	12	∞
D	∞	6	4	-	9	∞
E	7	18	12	9	-	12
F	10	8	∞	∞	12	-

Note: the symbol ∞ means that there is no direct connection between these vertices.

From A to B: (the direct distance from A to B is 12)

Since the values of the distances between A and C, A and E and A and F are all less than 12, we compute the following distances for the detours;

$A \xrightarrow{3} C \xrightarrow{6} B = 9$

$A \xrightarrow{7} E \xrightarrow{18} B = 25$

$A \xrightarrow{10} F \xrightarrow{8} B = 18$

since the minimum detour {9, 25,18} = 9 is less than 12 the direct distance between A and B we replace 12 by 9 in the table and likewise the distance between B and A in the table.

From A to C: the direct connection between A and C is 3 and no direct connection between A and any other node is less than 3, so we retain the value 3 as the minimum distance between A and C and C and A.

A to D: $A \xrightarrow{12} B \xrightarrow{6} D = 18$

$A \xrightarrow{3} C \xrightarrow{4} D = 7$

$A \xrightarrow{7} E \xrightarrow{9} D = 16$

$A \xrightarrow{10} F \xrightarrow{\infty} B = \infty$

Minimum {18, 7, 16,∞} = so we replace ∞ from A to D and D and A by 7

A to E: (the direct connection between A and E is 7) since only the direct distance between A and C is less 7,we compute

$A \xrightarrow{3} C \xrightarrow{12} E = 15$, Which is less than 7 so we retain the 7 in the cell AE

A to F: (the direct connection between A and F is 10) but the distances between A and C, A and D and A and E are less than 10 so we compute the distances for the detours,

$A \xrightarrow{3} C \xrightarrow{\infty} F = \infty$

$A \xrightarrow{7} D \xrightarrow{\infty} F = \infty$

$A \xrightarrow{7} E \xrightarrow{12} F = 19$

Min $\{\infty, \infty, 19\} > 10$, so retain the values in cell AF and FA.

From A to the other nodes, the table is

Table 3.2.1 represents the matrix of the minimum distances (in km) of direct roads between town A and the other towns

Table 3.2.1 the matrix of the minimum distances (in km) of direct roads between town A and the other towns

	A	B	C	D	E	F
A	-	9	3	7	7	10
B	9	-	6	6	18	8
C	3	6	-	4	12	∞
D	7	6	4	-	9	∞
E	7	18	12	9	-	12
F	10	8	∞	∞	12	-

From the node B.

From B to A: (the direct distance from B to A is 9)

$B \xrightarrow{3} C \xrightarrow{6} A = 12$

$B \xrightarrow{6} D \xrightarrow{7} A = 13$

$B \xrightarrow{8} E \xrightarrow{7} F = 15$

Min{12, 13, 15} = 12 > 9 so, we retain the value of 9 in the cell BA and AB

For the cells BC and BD each with value 6, no other value in the cells beginning with B is less than 6, so we retain the value 6 in the cells BC, CD, BD and DB.

B to E: (the direct connection between B and E is 18)

$B \xrightarrow{9} A \xrightarrow{7} E = 16$

$B \xrightarrow{6} C \xrightarrow{12} E = 18$

$B \xrightarrow{6} D \xrightarrow{9} E = 15$

$B \xrightarrow{8} F \xrightarrow{12} E = 20$

Min{16,18,15,20} = 15 $<$ 18, we replace the values of cells BE and EB with 15

B to F: (the direct connection between B and F is 8)

$B \xrightarrow{6} C \xrightarrow{12} F = \infty$

$B \xrightarrow{7} D \xrightarrow{12} F \quad \infty$

Min $\{\infty, \infty, \} = \infty > 8$, so retain the values in the cell BF and FB.

From B to the other nodes, the table is

Table 3.2.2 represents the matrix of the minimum distances (in km) of direct roads between towns A and B and other towns

Table 3.2.2 the matrix of the minimum distances (in km) of direct roads between town A and B and other towns

	A	B	C	D	E	F
A	-	9	3	7	7	10
B	9	-	6	6	15	8
C	3	6	-	4	12	∞
D	7	6	4	-	9	∞
E	7	15	12	9	-	12
F	10	8	∞	∞	12	-

From the node C.

From C to A: (the direct distance from C to A is 3) since no direct connection between C and any other node is less than 3, so we retain the value 3 as the minimum

distance between C and A and A and C.

Min{12, 13, 15} = 12 $>$ 9 so, we retain the value of 9 in the cell BA and AB

From C to B: (the direct distance from C to B is 6), but those in the cells CA and CD are less than 6, so we compute the distances of the detours

$$C \xrightarrow{3} A \xrightarrow{9} B = 12$$

$$C \xrightarrow{4} D \xrightarrow{6} B = 10$$

Min{12,10}=10 > 6 , we retain the values in cells CB and BC

C to D: (the direct connection between C and D is 4)

$C \xrightarrow{3} A \xrightarrow{7} D = 10 > 4$ so we retain 4 in cells CD and DC

C to E: (the direct connection between C and E is 12)

$$C \xrightarrow{3} A \xrightarrow{7} E = 10$$

$$C \xrightarrow{6} B \xrightarrow{8} E = 14$$

$$C \xrightarrow{4} D \xrightarrow{9} E = 13$$

Min{10,14,13} = 10 < 12, we replace the values of cells CE and EC with 10

C to F: (the direct connection between C and F is ∞)

$$C \xrightarrow{3} A \xrightarrow{10} F = 13$$

$$C \xrightarrow{6} B \longrightarrow F = 15$$

$$C \xrightarrow{4} D \xrightarrow{12} F = 20$$

Min $\{13,14,\infty,24,\} = 13 < \infty$, so we replace the values in the cell CF and FC.

Table 3.2.3 represents the matrix of the minimum distances (in km) of direct roads between towns A, B and C and the other towns

Table 3.2.3 the matrix of the minimum distances (in km) of direct roads between town A,B and C and the other towns

	A	**B**	**C**	**D**	**E**	**F**
A	-	9	3	7	7	10
B	9	-	6	6	15	8
C	3	6	-	4	10	13
D	7	6	4	-	9	∞
E	7	15	10	9	-	12
F	10	8	13	∞	12	-

From the node D.

From D to A: (the direct distance from D to A is 7)

$$D \xrightarrow{6} B \xrightarrow{9} A = 15$$

$$D \xrightarrow{4} C \xrightarrow{3} A = 7$$

Min{15,7}=7 $=$ 6 , we retain the values in cells DA and AD

From D to B: (the direct distance from D to B is 6

$D \xrightarrow{4} C \xrightarrow{6} B = 10 > 6$ so we retain 6 in cells DB and BD

D to C: (the direct connection between D and C is 4) since no direct connection between D and any other node is less than 4, so we retain the value 4 as the minimum

distance between D and C and C and D.

D to E: (the direct connection between D and E is 9)

$D \xrightarrow{7} A \xrightarrow{7} E = 14$

$D \xrightarrow{6} B \xrightarrow{15} E = 21$

$D \xrightarrow{4} C \xrightarrow{10} E = 14$

Min{14,21,14} = 14 > 9, we retain the value 9 in the cells DE and ED

D to F: (the direct connection between D and F is ∞)

$D \xrightarrow{7} A \xrightarrow{10} F = 17$

$D \xrightarrow{6} B \xrightarrow{8} F = 14$

$D \xrightarrow{4} C \xrightarrow{13} F = 17$

$D \xrightarrow{9} E \xrightarrow{12} F = 21$

Min {17,14,17,21, } = 14 < ∞ , so we replace the values in the cell DF and FD by 14.

From D to the other nodes, the table is

	A	**B**	**C**	**D**	**E**	**F**
A	-	9	3	7	7	10
B	9	-	6	6	15	8
C	3	6	-	4	10	13
D	7	6	4	-	9	14
E	7	15	10	9	-	12
F	10	8	13	14	12	-

From the node E.

From E to A: (the direct distance from E to A is 7)) since no direct connection between E and any other node is less than 7, so we retain the value 7 as the minimum

distance between E and A and A and E.

E to B

$$E \xrightarrow{7} A \xrightarrow{9} B = 16$$

$$E \xrightarrow{10} C \xrightarrow{6} B = 16$$

$$E \xrightarrow{9} D \xrightarrow{6} B = 15$$

$$E \xrightarrow{12} F \xrightarrow{8} B = 20$$

Min{16,16,15,20}=15 $= 15$, we retain the values in cells EB and BE

From E to C: (the direct distance from E to C is 10

$$E \xrightarrow{7} A \xrightarrow{3} C = 10$$

$$E \xrightarrow{9} C \xrightarrow{4} E = 13$$

Min{10,13}=10 which is not less than 10 so we retain it

E to D: (the direct connection between E and D is 9)

$E \xrightarrow{7} A \xrightarrow{7} D = 14 > 9$, so we retain 9 in the cells ED and DE

E to F: (the direct connection between E and F is 12)

$$E \xrightarrow{7} A \xrightarrow{10} F = 17$$

$$E \xrightarrow{10} C \xrightarrow{13} F = 23$$

$$E \xrightarrow{9} D \xrightarrow{14} F = 23$$

Min $\{17,23,23\} = 17 > 12$, so we retain the value12 in the cells EF and FE.

Since the values in the tables are symmetric, after computing the value EF we can stop. Hence the shortest distances between all pairs of node is as shown in the table below:

Table 3.2.4 shows the minimum distances (in km) of direct roads between the towns

Table 3.2.4 the minimum distances (in km) of direct roads between the towns

	A	B	C	D	E	F
A	-	9	3	7	7	10
B	9	-	6	6	15	8
C	3	6	-	4	10	13
D	7	6	4	-	9	14
E	7	15	10	9	-	12
F	10	8	13	14	12	-

3.6.0 P – Median Problem

One of the important measures of the effectiveness of a given locational configuration is the average distance or time that is traveled by those who utilize the facilities. The smaller this quantity, the more accessible the system is to its users. This is appealing, since the smaller the average distance of travel, the less one is inconvenienced in getting to his closest facility. Therefore, one approach to public facility location planning could be to locate a given number of facilities such that the resulting average travel distance is minimized. The objective of minimizing average travel distance is equivalent to an objective of minimizing total weighted travel distance. Within a network context, this location problem can be defined in the following way: Minimize the total weighted travel distance associated with a network of demand nodes by locating p-facilities on the network (on arcs or at nodes) where each demand node is served by its closest facility.

This problem is called the p-median problem and was first introduced by Hakimi **. A** set of p points that yield the smallest possible weighted distance is called an "optimal" p-facility solution set, and the points in the set that minimize the sum of the weighted distances are said to be medians of the network. For any value of p (i.e., number of facility points), it has been shown that there is at least one optimal p-median solution set which consists entirely of nodes of the network. Therefore, an optimal solution to the p-median problem could be achieved by minimizing the total weighted distance with the facilities restricted to nodes of the network.

3.6.1 Formulation of P – median problem

Consider a set L of m facilities (or location points), a set U of n users (or customers or demand points), and a $n \times m$ matrix D with the distances travelled (or costs incurred) for satisfying the demand of the user located at i from the facility located at j, for all $j \in L$ and $i \in U$. The objective is to minimize the sum of these distances or transportation costs

$$(\min) \sum_{i \in U, j \in J} \min d_{ij}$$

where $J \subseteq L$ and $|J| = p$. PMP can be defined as a purely mathematical problem: given an $n \times m$ matrix D, select p columns of D in order that he sum of minimum coefficients in each line within these columns be smallest possible.

The p-median problem and its extensions are useful to model many real word situations, such as the location of industrial plants, ware-houses and public facilities.

Beside this combinatorial formulation, the PMP has also an integer programming one.

ReVelle and Swain (1970) proposed an optimal procedure for the *p*-median, based on linear programming and branch and bound. Their formulation is now well known and used profusely, in a slightly different form:

***p*-median:**

$$min \sum_{j} h_i d_{ij} x_{ij} \qquad (1)$$

s.t.

$$\sum_{j} x_{ij} = 1 \qquad i = 1,2, \dots, n \qquad (2)$$

$$x_{ij} \leq y_j \qquad i, j = 1,2, \dots, n \qquad (3)$$

$$\sum_{j} y_j = p \qquad (4)$$

$$x_{ij}, y_j \in \{0,1\} \qquad i, j = 1,2, \dots, n \qquad (5)$$

Where:

i Index of demand points

m Total number of demand points in the space of interest

j Index of potential facility sites

n Total number of potential facility locations

h_i Weight associated to each demand point.

d_{ij} Distance between demand area i and potential facility at j.

x_{ij} Variable that is equal to 1 if demand area i is assigned to a facility at j, and 0 otherwise

y_j Variable that is equal to 1 if there is an open facility at j, and 0 otherwise.

The first set of constraints forces each demand point to be assigned to only one facility. The second set of constraints allows demand point *i* to assign to a point *j* only if there is an open facility in this location. Finally, the last constraint sets the number of facilities to be located.

Algorithm of p – median

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.

Illustrative example

Figure 3.7 shows distances between localities and their populations

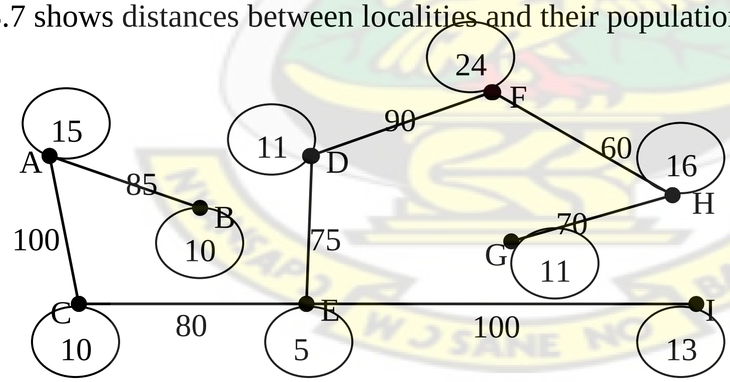


Figure 3.7 network of localities and population

Table 3.3 shows distances between localities

Table 3.3 distances between localities

localities	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**
A	**-**	**85**	**100**	**255**	**180**	**345**	**475**	**405**	**280**
B	**85**	**-**	**185**	**340**	**265**	**355**	**485**	**415**	**365**
C	**100**	185	**-**	**155**	**80**	**245**	375	**305**	**180**
D	**255**	340	155	-	75	**90**	220	150	175
E	**180**	265	80	75	**-**	165	295	225	**100**
F	**345**	355	245	90	**165**	-	130	**60**	**265**
G	**475**	485	375	**220**	**295**	130	**-**	**70**	**395**
H	**405**	415	305	**150**	225	**60**	70	-	**325**
I	**280**	365	**180**	**175**	**100**	265	395	325	-

Table 3.4 shows the Population of the localities

Table 3.4   Population of localities

offices	A	B	C	D	E	F	G	H	I
population	15	10	10	18	5	24	11	16	13

Figure 3.7.1 shows the distances between A and the localities and their population

Total weight distance W_A of Locality A is calculated as

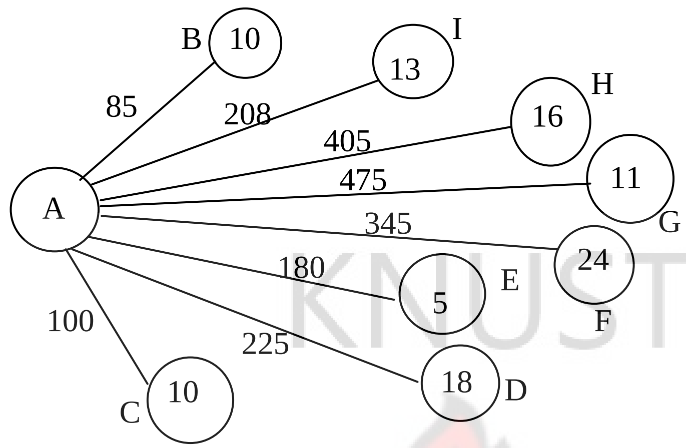


Figure 3.7.1 distances between A and the localities and their population

$$W_A = (85 \times 10) + (475 \times 11) + (345 \times 24) + (180 \times 5) + (225 \times 18) + (100 \times 10) + (405 \times 16) + (208 \times 13)$$

$= 29489m$

Therefore if W_A which is office A is to be selected, it means that for all the peoples from other offices to access the facility at town A, a total distance of 20305m must be covered.

Figure 3.7.2 shows the distances between B and the localities and their population

The total weighted distance (W_B) of office B is calculated as follows:

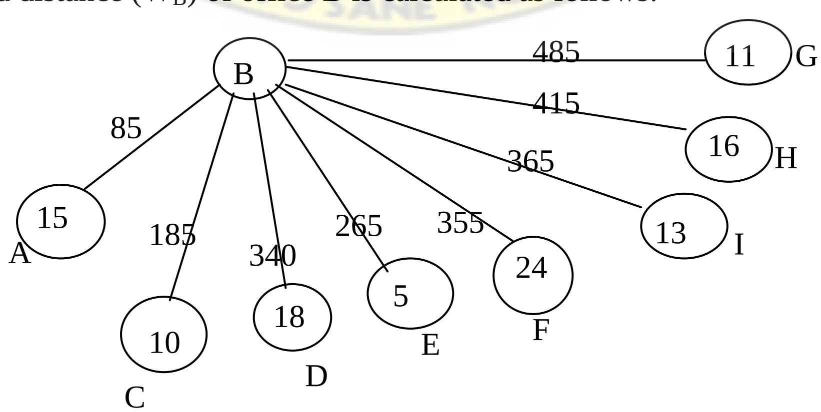


Figure 3.7.2: distances between B and the localities and their population

$W_B = (85 \times 15) + (185 \times 10) + (340 \times 18) + (265 \times 5) + (355 \times 24) + (365 \times 13) +$

$(415 \times 16) + (485 \times 11) = 35810m$

Therefore if W_B which is office B is to be selected, it means that for all the peoples from other offices to access the facility at town B, a total distance of 35810m must be covered.

Figure 3.7.3 shows the distances between B and the localities and their population

The total weighted distance (W_C) of office C is calculated as follows:

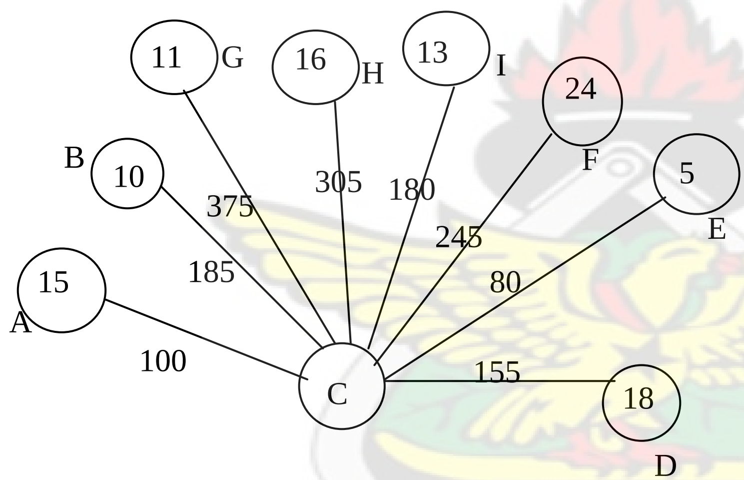


Figure 3.7.2: distances between C and the localities and their population

$W_C = (100 \times 15) + (185 \times 10) + (155 \times 18) + (80 \times 5) + (245 \times 24) + (180 \times 13) +$

$(305 \times 16) + (375 \times 11) = 23765m$

Therefore if W_C which is office C is to be selected, it means that for all the peoples from other offices to access the facility at town C, a total distance of 23765m must be covered.

Figure 3.7.4shows the distances between D and the localities and their population

The total weighted distance (W_D) of office D is calculated as follows:

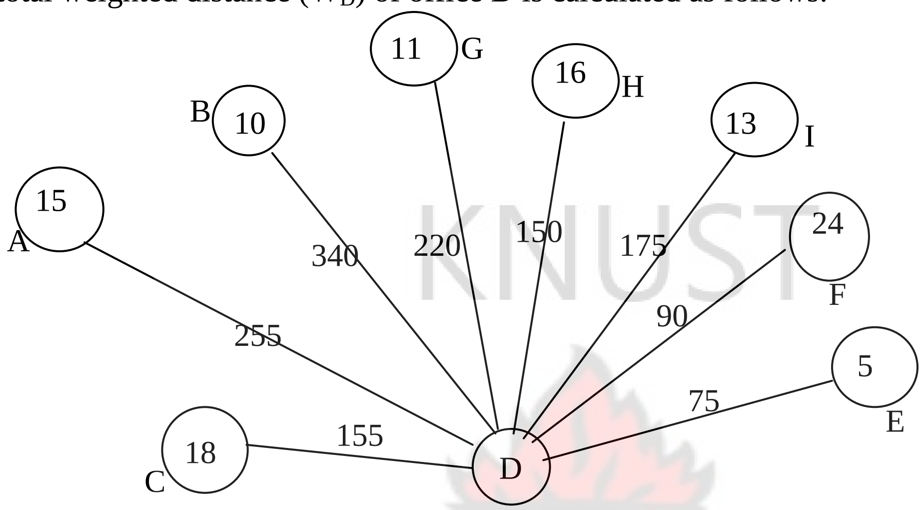


Figure 3.7.4: distances between D and the localities and their population

$$W_D = (255 \times 15) + (340 \times 10) + (155 \times 18) + (75 \times 5) + (90 \times 24) + (175 \times 13) + (150 \times 16) + (220 \times 11) = 19645\text{m}$$

Therefore if W_D which is office D is to be selected, it means that for all the peoples from other offices to access the facility at town D, a total distance of 19645m must be covered.

Figure 3.7.5 shows the distances between I and other localities and their population

The total weighted distance (W_I) of office I is calculated as follows:

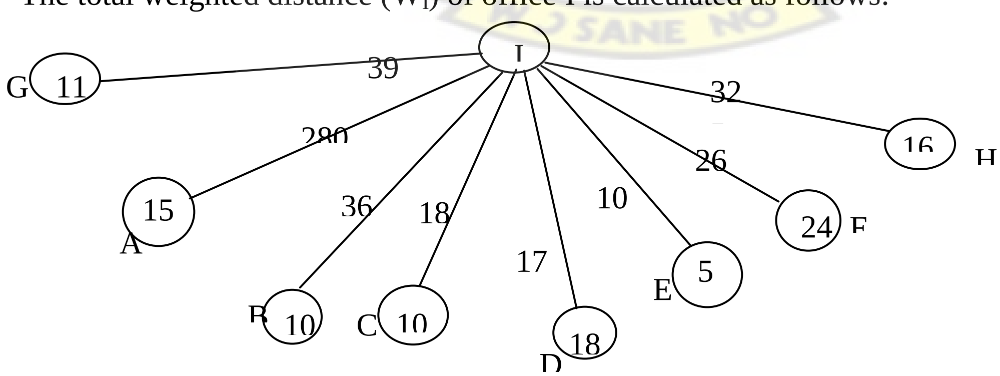


Figure 3.7: distances between I and other localities and their population

$W_I = (280 \times 15) + (365 \times 10) + (175 \times 18) + (10 \times 180) + (100 \times 5) + (395 \times 11) +$

$(325 \times 16) + (265 \times 24) = 29205m$

Table 3.5 shows the weighted distances of the localities

Table 3.5: weighted distances of localities

Office	A	B	C	D	E	F	G	H	I
Weight Length	29489	35810	23765	19645	19605	19635	30625	23885	29205

The minimum of the weighted distance above is 19605m which corresponds to E. Therefore Locality E is the best location for the conference hall.

Figure 3.8 represents minimum spanning tree and the location point

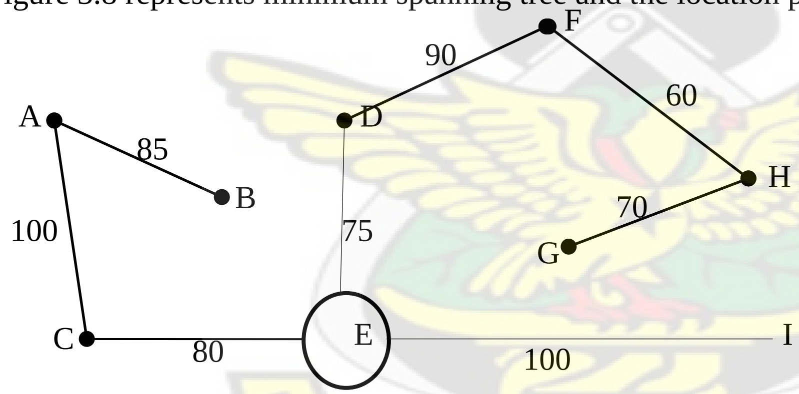


Figure 3.8 minimum spanning tree and location point

CHARPTER FOUR

DATA COLLECTION AND ANALYSIS

4.0 Data collection

Information and Data needed for the analyses were gathered from the school administration. These are

1. Pipe network system of the school
2. Population of various destinations

4.1 Data

Table 4. 1 shows the location of the nodes of the school and their representative alphabets

Table 4.1 localities and their respective alphabet

Sites	**localities**	**population**
A	**Sources**	**5**
B	Girls dormitory	370
C	Masters bungalow 1	41
D	Masters bungalow 2	35
E	Classroom block 1	680
F	Kitchen	17
G	Classroom block 2	420
H	Administration block	62
I	Boys dormitory	255
J	Headmaster's dormitory	8

Source: St. Augustine's Senior High School- Bogoso administration

Figure 4.1 represent the network system of St. Augustine’s Senior High School- Bogoso (in metres)

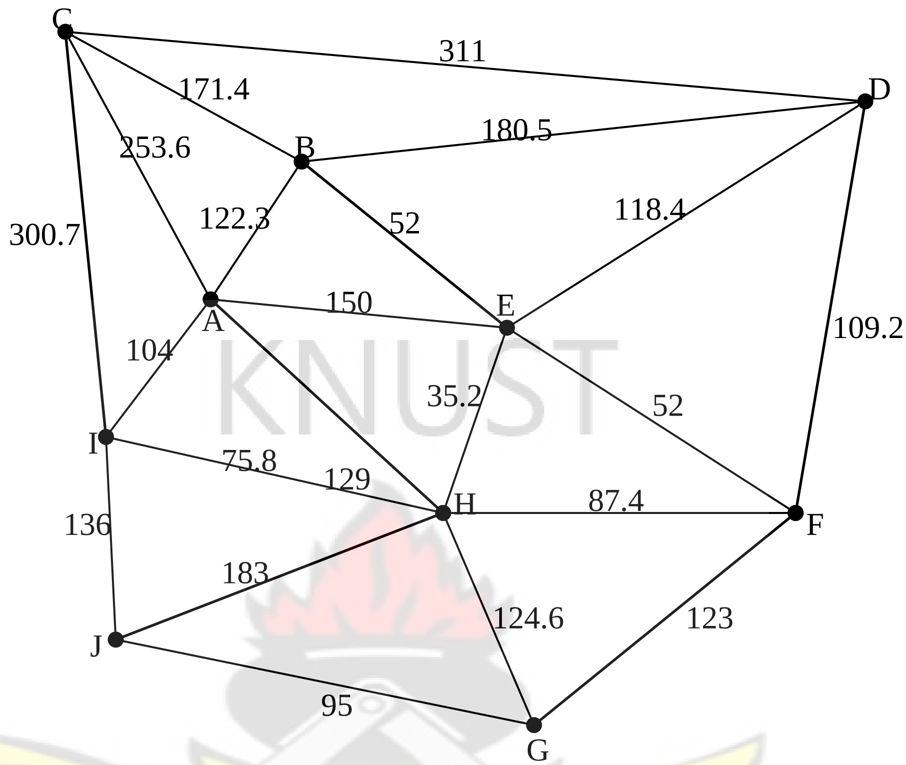


Figure 4.1 network system of St. Augustine’s Senior High School- Bogoso

4.2 Minimum connector fomulation

Given a connected undirected graph G = (V, E), with the node set V, the edge set E, weight ω_e for all edge in E and x_e for all edge in E_T and $x_e = \begin{cases} 1 & if\ edge\ e \in E_T \\ 0 & otherwise \end{cases}$.
Find a spanning tree $G_T = (V_T, E_T)$ of minimum total weight with the node set V_T and the edge set E_T. The problem can be model as

$$min \quad \sum_{e \in E} \omega_e x_e$$

$$s.t \quad \sum_{e \in E} x_e = n - 1$$

$$\sum_{e \in (S,S)} x_e \leq |S| - 1 \quad \forall S \subseteq V \qquad (1c)$$

$$x_e \in \{1, 0\} \quad \forall e \subseteq E$$

Where

(S, S) = all edges that go from a locality in the set S to another locality in the set S,

V = the set of nodes (localities)

E = the set of edges (distances)

x_e = edges of the minimum spanning tree of the localities

$$x_e = \begin{cases} 1 & if\ edge\ e \in E_T \\ 0 & otherwise \end{cases}$$

$\omega_e x_e$ = weight of the edges of the minimum spanning tree of the localities

n=number of edges of the localities

Equation (1c) enforces the constraint that the edges in E_T can't form cycles.

4.3 Prim's algorithm

Given is an undirected graph G= (*V*, *E*), where V is the set of nodes (localities) and E is the set of edges (distances). Each edge is defined by a node pair (a, b), each edge has a non-negative length. Then the prim's algorithm works as follows:

Step 1 Create a tree containing a single vertex, chosen arbitrarily from the graph

Step 2 Create a set containing all the edges in the graph

Step 3 Loop until every edge in the set connects two vertices in the tree

- O removes from the set an edge with minimum weight that connects a vertex in the tree with a vertex not in the tree

- O add that edge to the tree

Thus Prim's algorithm works from a starting point and builds up the spanning tree step by step, connecting edges into the existing solution. 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.

The algorithm can be stated as follows:

Step 1: choose a starting vertex

Step 2 : From the starting vertex draw the lowest valued edge to start your tree. (Any vertex can be chosen as the start vertex; however, it will always be given in an exam question.)

Step 3 : From any vertex on your tree, add the edge with the lowest value.

Step 4: If there *n* – 1 edges in your tree, you have finished. If not, go to step 3.

NB : At each step remember to make sure you do not make a cycle.

A key aspect of Prim's algorithm is that as the tree is being built, the tree is always connected. You are always adding a new vertex to your current tree.

4.4 COMPUTATIONAL PROCEDURE FOR PRIM'S ALGORITHM

The computer brand used in running the programming code was HP 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 the data shown in Appendix B. the program code was ran twice and all yielded the same result.

4.5 DATA USED FOR PRIM'S ALGORITHM

Table 4.2 depicts distances between the localities.

Table 4.2 distances between localities

	A	B	C	D	E	F	G	H	I	J
A	0	122.3	253.6	0	150	0	0	129	104	0
B	122.3	0	171.4	180.5	152	0	0	0	0	0
C	253.6	171.4	0	311	0	0	0	0	300.7	0
D	0	180.5	311	0	118.4	109.2	0	0	0	0
E	150	152	0	118.4	0	52	0	35.2	0	0
F	0	0	0	109.2	52	0	123	87.4	0	0
G	0	0	0	0	0	123	0	124.6	0	95
H	129	0	0	0	35.2	87.4	124.6	0	75.8	183
I	104	0	300.7	0	0	0	0	75.8	0	136
J	0	0	0	0	0	0	95	183	136	0

4.6 RESULTS FOR PRIM'S ALGORITHM

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

Table 4.3 shows the edges in minimum spanning tree and their distances

Table 4.3 : edges and their minimum distances

FROM	**TO**	**DISTANCE (m)**
A	I	104
I	H	75.8
H	E	35.2
E	F	52.0
F	D	109.2
A	B	122.3

F	G	123.0
G	J	95.0
B	C	171.4

The total distance for the spanning tree is **887.9m**

Figure 4.2 shows minimum spanning tree of the network system of St. Augustine's Senior High School- Bogoso

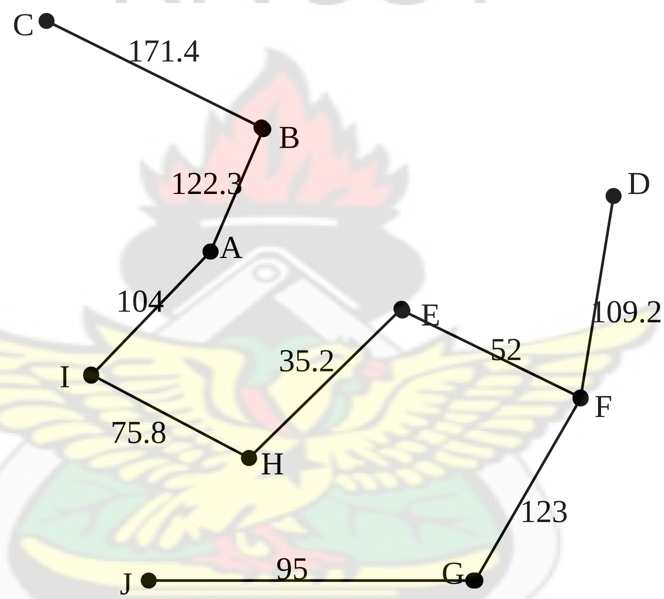


Figure 4.2: minimum spanning tree of the network system of St. Augustine's Senior High School- Bogoso

4.7 Formulation of P – Median

Given m location points (or facilities), n users (or customers or demand points), and a $n \times m$ matrix D with the distances travelled (or costs incurred) for satisfying the demand of the user located at i from the facility located at j as indicated in table 4.1 and 4.3. The objective is to minimize the sum of these distances or transportation costs. The model can be represented mathematically as follows.

$$min \sum_{ij} h_i d_{ij} x_{ij} \qquad (1)$$

s.t.

$$\sum_{j} x_{ij} = 1 \qquad i = 1,2,\ldots,10 \qquad (2)$$

$$x_{ij} \leq j_j \qquad i,j = 1,2,\ldots,10 \qquad (3)$$

$$\sum_{j} y_j = p \qquad (4)$$

$$x_{ij}, j_j \in \{0,1\} \qquad i,j = 1,2,\ldots,10 \qquad (5)$$

Where:

i Index of demand points

j Index of potential facility sites

n Total number of the localities

h_i the population of the localities.

d_{ij} Distance between the other localities and the selected locality

x_{ij} Variable that is equal to 1 if demand area i is assigned to a facility at j, and 0 otherwise

y_j Variable that is equal to 1 if there is an open facility at j, and 0 otherwise.

P = 1 which signifies 1 median

The first set of constraints forces each demand point to be assigned to only one facility. The second set of constraints allows demand point i to assign to a point j only if there is an open facility in this location. Finally, the last constraint sets the number of facilities to be located.

The algorithm of p – median can be stated 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.

4.8 COMPUTATIONAL PROCEDURE FOR 1-MEDIAN

The computer brand used in running the programming code was HP 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 ***p*** and it was done twice with the same result.

4.9 DATA FOR 1- MEDIAN ON A TREE

D is n x n matrix representing the distances between the nodes (localities) for the minimum connector after using Prim's algorithm. **P** is a row vector representing the population.

Table 4.5 shows the direct distances between the localities

Table 4.5: direct distances between the localities

D =

0	122.3	0	0	0	0	0	0	104	0
122.3	0	171.4	0	0	0	0	0	0	0
0	171.4	0	0	0	0	0	0	0	0
0	0	0	0	0	109.2	0	0	0	0
0	0	0	0	0	52	0	35.2	0	0
0	0	0	109.2	52	0	123	0	0	0
0	0	0	0	0	123	0	0	0	95
0	0	0	0	35.2	0	0	0	75.8	0
104	0	0	0	0	0	0	75.8	0	0
0	0	0	0	0	0	95	0	0	0

P is a row matrix representing the population of the localities

P= [5 370 41 35 680 17 420 62 255 8]

4.10 RESULTS FOR 1-MEDIAN ON A TREE

Using the Matlab Codes for ReVelle and Swain algorithm for 1-Median problem on the minimum spanning tree, the following results were obtained. The codes are listed in appendix A.

Tables 4.5 shows the various localities and their weighted distances

Table 4.5: localities and their weighted distances

S/N	LOCALITIES	WEIGHTED DISTANCE
1	A	426546.3
2	B	557529.6
3	C	867925
4	D	507293.3
5	E	259406.1
6	F	307922.1
7	G	369693.1
8	H	274352.5
9	I	316202.3
10	J	615188.1

From the above results, the minimum weighted distance is **259406.1m** which occurs at the Classroom block 1. Therefore the central standpipe should be located closer to the Classroom block 1

Figure 4.3 shows the minimum spanning tree and location point.

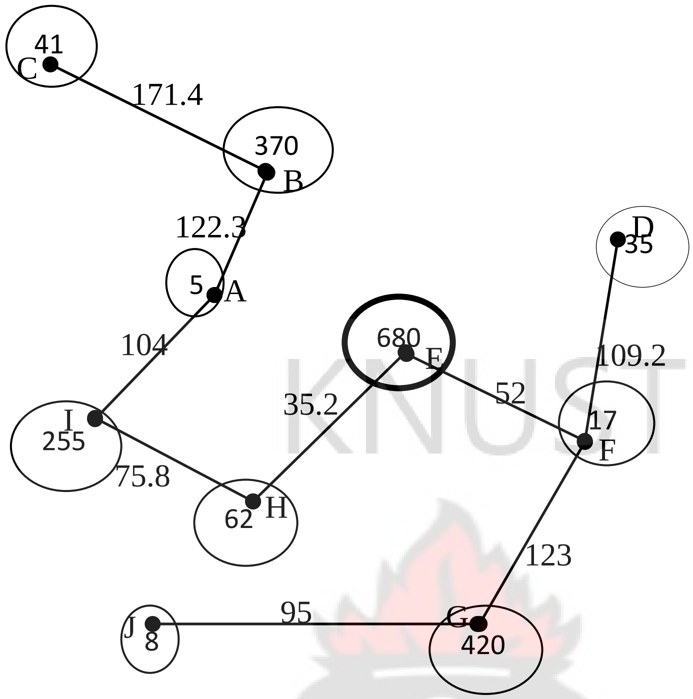


Figure 4.3 Minimum Spanning tree and location point

4.11 Discussion

The optimal total distance covered by the layout of the network is 887.9 metres. The layout for the pipe network starts from the main source (A) of water supply. The source has two outlets. One passes through the Girls' dormitory and leads to the masters' bungalow 1 (C) while the other passes through Boys' Dormitory (I), administration block (H), class room block 1 (E) to the kitchen (F). From the Kitchen (F) it branches to the masters' Bungalow 2 (D) also to Classroom Block 2 (G) and finally to the headmaster's flat

From the above results in table 4.5, the minimum weighted distance is **259406.1m** which occurs at the Classroom block 1. The central standpipe should be located closer to the Classroom block 1.

CHAPTER FIVE

5.0 CONCLUSIONS

The network data provided by the school administration was analyzed and modeled to suit P-median which was solved by ReVelle and Swain algorithm and minimum connector which was solved by Prim's Algorithm in determining the location of standpipe and the optimal layout of the pipe network system of St. Augustine's Senior High school. The total distance for the minimum spanning tree is **887.9m** and the layout is indicated in figure 4.2 in page 72. Also the location of the central standpipe on the minimum connection network was modeled as 1 median problem to locate the central standpipe. The minimum weighted distance is **259406.1m** which occurs at the Classroom block 1. The central standpipe should be located closer to the Classroom block 1.

5.1 RECOMMENDATIONS

- The school administration should employ the finding from this research to lay the pipelines system in the school.
- The school should employ the services of computer expert or programmer to help them locate a suitable place of the standpipe on the minimum connector.

REFERENCES

1. Adu-Ahyiah, Maxwell; Anku, Romi Ernest. "Small Scale Wastewater Treatment in Ghana (a Scenario)" (PDF). Lund. Retrieved 2008-03-28., p. 1-2

2. Agodzo, SK; Huibers, FP; Chenini, F.; van Lier, JB; Duran A. "Use of wastewater in irrgigated agriculture. Country studies from Bolivia, Ghana and Tunisia. Volume 2: Ghana" (PDF). Wageningen: WUR. ISBN 90-6754-704-2. Retrieved 2008-03-28. , p. 16-17

3. Amponsah, S.K. and Darkwah, F.K. (2007). Operation Research, pp 23-26.

4. Arogundade, O.T. and Akinwale, A.T. (2009). Application of Prim's Algorithm to a Profit Oriented Transportation System in Rural Areas of Nigeria. Proceedings of World Academy of Science, Engineering and Technology. Vol. 37, pp 1295-1306.

5. Berman, O. and Larson, R. C. (1982), The Median Problem with Congestion, *Computers and Operations Research* 4(2) 119-126

6. Brimberg, J. Mladenović, N. (2003). Oil pipeline design problem. Operation Research, Vol. 51, pp. 228-239

7. Divoky, James J., Hung, Ming S (1990). *Performance of shortest path algorithms in network flow problems.* Institute for Operations Research and the Management Sciences.

8. Densham, P.J. and Rushton, G,(1992). A more efficient heuristic for solving large *p*-median problems. *Papers in Regional Science* 71 (3), 307-329.

9. Dijkstra, E.W., 1959. "A note on two problems in connexion with graphs". Numerische Mathematik, 1, 269 – 271

10. 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.

11. Gloor, P. A., Laubacher, R.. Dynes, S.B.C. and 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.

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

13. Goldman, A J. (1969) Optimal locations for centers in a network, *Transportation Science* 3(2), 352-360.

14. Hakimi S .L. (1964) Optimal location of switching centers and the absolute centers and medians of a graph, *Operations Research* 12, 450– 459.

15. Hillsman, E. and Rhoda, R.(1978) Errors in measuring distance from populations to service centers, *Annals of the Regional Science Association*, 12, 74-88.

16. Holmes, J., Williams, F. and Brown, L. (1972) Facility location under maximum travel restriction: An example using day care facilities, *Geographical Analysis*, 4, 258-266.

17. Kariv O and Hakimi S L. (1979) An Algorithmic Approach to Network Location Problems. II: The P medians, *SIAM Journal on Applied Mathematics*, 37, 539–560.

18. McCarthy (2009).The application of two well known graph algorithms; Edmond's algorithm and Prim's algorithm

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

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

21. 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.

22. Pettie , Vijaya R. and Srinath, S. (2002). Experimental Evaluation of a New Shortest Path Algorithm (Extended Abstract)

23. ReVelle, C. S. and Swain, R .W. (1970) Central Facilities Location, *Geographical Analysis*, 2, 30-42.

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

25. Thomas, H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein (2001). Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill. ISBN 0-262-03293-7. Section 23.2: The algorithms of Kruskal and Prim, pp. 567–574.

26. Voices.yahoo.com/the – importance – water health – human – life 1691455.html

26. World Health Organization; UNICEF. "Joint Monitoring Program for Water Supply and Sanitation (JMP)". Retrieved 2012-08-15.

27. Wesolowsky, G. O. and Truscott, W. G. (1975) The Multiperiod Location-Allocation Problem with Relocation of Facilities, *Management Science*, 22(1), 57-65.

APPENDIX A

MATLAB CODES FOR P-MEDIAN

```
%P is a row vector indicating the population of the suburbs

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

D=input('Enter D;')

P=input('Enter P;')

mn=size(D)

m=mn(1)

n=mn(2)

i=1:n

j=1:n

w=(P*D)'
```

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

MATLAB CODES FOR PRIM'S ALGORITHM

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

% User supplies adjacency matrix A.  This program uses Prim's algorithm
% to find a minimum spanning tree.  The edges of the minimum spanning
% tree are returned in array mst (of size n-1 by 2), and the total cost
% is returned in variable cost.  The program prints out intermediate
% results and pauses so that user can see what is happening.  To continue
% after a pause, hit any key.

A=input('Enter A:')
[n,n] = size(A);                        % The matrix is n by n, where n = # nodes.
A, n, pause,
if norm(A-A','fro') ~= 0 ,              % If adjacency matrix is not symmetric,
  disp(' Error:  Adjacency matrix must be symmetric ') % print error message and quit.
  return,
end;

% Start with node 1 and keep track of which nodes are in tree and which are not.
intree = [1];  number_in_tree = 1;  number_of_edges = 0;
notintree = [2:n]';  number_notin_tree = n-1;
in = intree(1:number_in_tree),              % Print which nodes are in tree and which
out = notintree(1:number_notin_tree), pause, % are not.
% Iterate until all n nodes are in tree.
while number_in_tree < n,
```

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

```
  out = notintree(1:number_notin_tree), pause,% which are not.


end;

disp(' Edges in minimum spanning tree and their costs: ')

[mst  costs]                    % Print out edges in minimum spanning tree.

cost = sum(costs)
```