KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

OPTIMAL VISITATION OF BASE TRANSCIEVER STATION SITE (CASE STUDY OF VODAFONE NETWORK; NEW JUABENG MUNICIPALITY, EASTERN REGION, GHANA)

BY

ISAAC ANSAH

A thesis submitted in partial fulfillment of the requirements

For the degree of master of Philosophy in industrial mathematics

DEPARTMENT OF MATHEMATICS

SCHOOL OF GRADUATE STUDIES

INSTITUTE OF DISTANCE LEARNING

JUNE, 2012

KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

OPTIMAL VISITATION OF BASE TRANSCIEVER SATATION SITE

(CASE STUDY OF VODAFONE NETWORK; NEW JUABENG MUNICIPALITY, EASTERN REGION, GHANA)

BY

ISAAC ANSAH

JUNE 2012

DECLARATION

I hereby declare that this submission is my own work towards the Master of Philosophy (MPhil) and that to the best of my knowledge, it contains no material previously published by another person nor material which has been accepted for the award of any other degree of the University, except where due acknowledgement has been made in the text.

Isaac Ansah (PG4063010)	…………………………	……………………
Name of Student and ID	Signature	Date
Certified by:		
S.K. Amponsah (Dr.)	……………………	……………………..
Supervisor	Signature	Date
Certified by:		
Mr. K. Darkwah	……………………..	…...………………….
Head of Department	Signature	Date

ACKNOWLEDGEMENT

I would like to acknowledge the advice and guidance of Dr. S.K Amponsah and Dr. Lord Mensah for their helpful discussions during the preparation of this thesis. I would also like to thank the other lecturers in the Department of Mathematics who have taught me during my study for the degree of Master of Philosophy at Kwame Nkrumah University of Science and Technology. Appreciation is also extended to the Graduate Institute of Distance Learning of this University who has made it possible for the award of this degree.

Lastly, I would like to express my gratitude to all my family members and my friends who have always been supportive of what I do. Thank you for all that you have provided for success to be achieved.

ABSTRACT

This research work presents a case study of the optimal visitation of Base Transceiver Station (BTS) site in a mobile communication network. The main objective is to formulate a mathematical model that takes into consideration the actual distances between the twenty four (24) BTS Site. Also to determine the optimal route for visiting the entire twenty four (24) BTS Site within the New Juabeng Municipality to minimize travelling cost, time and distance for Vodafone maintenance Engineers and entire sub-contractors working on the BTS sites. The problem is formulated as an Integer Programming Model and solution is presented via the simulated Annealing Algorithms based Meta-heuristic for the Travelling Salesman Problem. Data on distances were collected from potential length Between the BTS and with a Matlab implementation codes, results are obtained. In comparison with the existing routing system at Vodafone Network the results evince the outperformance of the simulated Annealing algorithm in terms of efficiency. In fact, the simulated Annealing results reveal a tremendous improvement in the total route length by approximately 45%.

TABLE OF CONTENTS PAGE

CHAPTER 2 LITERATURE REVIEW

CHAPTER 3 METHODOLOGY

CHAPTER 4 ANALYSIS AND RESULTS

CHAPTER 5 CONCLUSION

APPENDIX

LIST OF FIGURE

LIST OF TABLES

CHARPTER ONE

1.0 INTRODUCTION

Overview of Telecommunications in the Ghana

The first telegraph line in Ghana (then known as the Gold Coast) was a ten mile link installed in 1881 between the castle of the colony, then governor in Cape Coast and Elmina. The line was then extended to Christiansburg Castle near Accra, which became the seat of government, and later extended still further to Aburi, 26 miles outside Accra.

In 1882, the first public telegraph line stretching over a distance of 2.5 miles, was erected between Christiansburg and Accra. Between 1887 and 1889, these telegraph lines were extended to cover Accra, Prampram, Winneba, Saltpond, Sekondi, Ankobra, Dixcove, and Shama. All colonial castles or fort towns as well as commercial ports and fishing centers. In 1886, telegraph lines were extended to the middle and northern parts of Ghana into the territory of the Ashantis.

In order to improve communications in the southern part of the country, the first manual telephone exchange (70 lines) was installed in Accra in 1892. Twelve years later, in 1904, a second manual exchange consisting of 13 lines was installed in Cape Coast.

Ghana's telecommunications infrastructure was laid down and expanded by the colonial administration mainly to facilitate the economic, social, and political administration of the colony. In 1901, for example, the Ashantis were brought under British colonial rule, and telegraph lines were accordingly extended from Accra to the capital of the Ashanti

Kingdom and beyond. By the end of 1912, 1,492 miles of telegraph lines had been constructed to link forty-eight telegraph offices spread throughout the country. Before the beginning of World War I in 1914, 170 telephone subscribers had been served in Ghana, but it was between World War I and 1920 that the backbone of the main trunk telephone routes--Accra-Takoradi, Accra-Kumasi, Kumasi-Takoradi, and Kumasi-Tamale--was built using unshielded copper wires. By 1930, the number of telephone exchange lines in Ghana had grown to 1,560, linking the coastal region with the central and northern parts of the country.

Due to the depressed global economy of the 1940s, there was little or no growth in telecommunications in Ghana during and immediately after the Second World War. Nevertheless, during that period carrier equipment (1+1) was installed on the Accra-Takoradi, Accra-Kumasi, Kumasi-Takoradi, and Kumasi-Tamale physical trunks. These were later augmented with carrier equipment (3+1), thus increasing the trunks connecting these towns threefold.

In 1953, the first automatic telephone exchange with 200 lines was installed in Accra to replace the manual one erected 63 years earlier. Three years later, in 1956, the trunk lines connecting Accra, Kumasi, Takoradi, and Tamale were upgraded through the installation of a 48- and 12-channel VHF network. The attainment of independence by Ghana in 1957 brought new dynamism to the country's telecommunications development. A seven-year development plan launched just after independence hastened the completion of a second new automatic exchange in Accra in 1957. By the end of 1963, over 16,000 telephone subscribers and 32,000 rotary-type telephones were in use in Ghana.

Due to the rapid growth in commercial activities in mining, timber, cocoa, shear butter, and the like in outlying parts of the country, new manual exchanges were installed at Cantonments, Accra, Swedru, Koforidua, Ho, Tamale, Sunyani, and Kumasi during the post-independence years. The installed exchanges were Strowger (step-by-step) and PhilipUR49 Switch. The management of Ghana's telecommunication institutions was initially assigned to the Public Works Department but was transferred to the post office following the enactment of the Post Office Ordinance in 1886. Telecommunications was later administered by the government's Post and Telecommunications (P&T) Department until the early 1970s.

A new chapter in the development of Ghana's telecommunications system began in November 1974, when the Post and Telecommunication Department was declared a public corporation by National Redemption Council Decree No. 311. The department was placed under the authority of the Ministry of Transport and Communication, which is still responsible today for policy formulation and the control of Ghana's telecommunications sector. (Akorli & Allotey, 1999)

In 1975, the P & T contracted loans from many multilateral and bilateral financial institutions in order to undertake a number of development projects to modernize and expand both national and international telecommunication services in Ghana.

Even though the project suffered delays due to changes in government, economic recession and other factors, it was completed in 1985. As part of Ghana's long-term telecommunication development program, a Second Telecommunication Project (STP) with an eight-year time frame was initiated in 1987. This project was intended to provide further rehabilitation, modernization, and expansion of the telecommunications facilities not covered by the FTP, as well as, the restructuring of the P & T.

The components of the STP were as follows:

(i) Expansion of the microwave transmission network and provision of coast station facilities at Tema for maritime telecommunication services;

(ii) Rehabilitation and expansion of Ghana's switching network in thirteen urban centers and twenty-six rural communities;

(iii) Rehabilitation and expansion of the external cable network to match the switching component described above;

(iv) Rehabilitation of Ghana's satellite earth station;

(v) Provision of 330-line international telephone switch and a 1,000-line telex switch;

(vi) Procurement of subscriber terminal equipment, spare parts, vehicles, and personal computers; and separation and restructuring of the P & T into two entities.

The impact of the completed portion of the STP on telecommunication services in Ghana was modest but appreciable between 1986 and 1990. There was 20% growth in subscription. . In addition, the number of direct exchange lines in working order increased from 60 % in 1987 to 89 % in March 1992. Also the availability since October 1988 of international direct dial service In twelve (12) exchange areas resulted in the promotion of international business and trade. The number of international satellite circuits also grew--from 41 satellite circuits in 1988 to 193 satellite circuits and 84 terrestrial circuits in 1992 (World Bank Project Report, 1995).

At the end of 1992 the project had introduced AT&T direct service; airlines became capable of making reservations through SITA (Societé International Telecommunication Aeronautique) facilities; a meteorological department was created to send meteorological and seismological data from various locations throughout Ghana to Accra; and press agencies/news houses, commodity markets, and financial institutions gained increased information and data transfer capabilities for transmission to and from the outside world.

Other benefits of the completed project were the capability of the Ghana Broadcasting Corporation (GBC) to transmit voice cast (radio commentary) and live television telecasts via satellite and simultaneous TV transmission from all GBC transmitters in the country.

Despite the achievements of the STP, Ghana's telephone density, in 1994, (0.31 per one hundred inhabitants), is still among the lowest in Africa. Typical telephone densities for other African nations include 9 % for Libya, 1.3 % for Zimbabwe, 0.5 % for the Ivory Coast, 0.33 % for Togo, 0.2 % for Nigeria, and 0.1 % for Burkina Faso. The enormity of the task facing Ghana and other African nations attempting telecommunications modernization becomes apparent when African telephone density rates are compared to those of selected nations in Europe and Asia: 62.4 telephones per one hundred inhabitants in Sweden, 43 in the United Kingdom, 42 in Japan, 41 in France, and 8 in Malaysia, (Akorli and Allotey, 1999).

The deregulation of the telecommunication sector by the Government of Ghana in 1987 saw the emergence of private companies in the sector. By 1992, about forty telephone companies were in operation, including a local cellular company and a paging company.

Other companies supply, install, and maintain terminal equipment such as facsimiles, telephones, Private branch exchange and Private automatic branch exchange (PBX, and PABX). In spite of the improvements that resulted from this development, telecommunications in Ghana was still extremely inadequate. In 1995, only 37 of the 110 administrative districts of the country had telephone exchange facilities, and there were only 35 payphones in the entire country with 32 in Accra, (Salia, 1995).

In 1995, the Post and Telecommunications Corporation was split into two autonomous divisions by the government of Ghana, Ghana Postal Services and Ghana Telecom. This was done in order for the company to function as a commercially viable entity (Akorli and Allotey, 1999).

According to Missing Link, (the 1994 report of the International Telecommunication Union's (ITU) Independent Commission for World Telecommunication), “penetration of telephones in Ghana in 1992 was only 0.32 per one hundred inhabitants”. In 1996 Ghana privatized its incumbent telecommunication firm by selling 30 percent of Ghana Telecom to the G-Com consortium, in which Telekom Malaysia (TM) holds an 85% stake, for USD 38 million, (Haggarty, 2002).

Ghana licensed a second network operator, and allowed multiple mobile firms to enter the market. The reforms yielded mixed results. Landline telephone penetration increased dramatically while the number of mobile subscribers surpassed even this higher level of fixed line subscribers. On the other hand, the network did not reach the levels the government hoped, the second network operator, Western Telecommunication Company

Ltd (Westel), never really got off the ground, and the regulator remained weak and relatively ineffective.

In February, 2002 government of Ghana contracted Telenor Management Partners (TMP) of Norway to provide a management consultancy to Ghana Telecom. A Ghanaian management later took over the affairs of Ghana Telecom prior to its acquisition by Vodafone.

In 2006 it had around 400,000 customers for fixed and mobile telephony and internet services. On 16th April, 2009 Ghana Telecom was rebranded as Vodafone (Akorli and Allotey, 1999).

1.1 Background of Eastern region and Koforidua municipality

The Eastern Region, with an area of 19,323 square kilometers, occupying 8.1 per cent of the total land area of Ghana, is the sixth largest region of the country. A total of 2,106,696 populations for the region, representing 11.1 per cent of Ghana's population. It is the third most populous region, after the Ashanti and Greater Accra. The population is made up of 49.2 per cent males and 50.8 per cent females, giving a sex ratio of 96.8 males to 100 females, which comprises of Seventeen (17) districts.

Koforidua, also popularly known as Kof-town, is a city in the West African republic of Ghana, about two hours by road from the capital city, Accra. It is the administrative capital of the Eastern Region of Ghana and has a total population of one hundred and sixty eight thousand (168 000)(2011 census) with a land area of one hundred and ten(110) square kilometers which serves as a commercial center for the region and New-Juabeng Municipal District.

The businesses in the municipality are mostly small and medium scale enterprises (SMEs). The municipality has 20 second cycle institutions, 35 basic schools, 1 college of education, 2 nursing training schools and 1 polytechnic and 1 university. All of these institutions are potential customers of Vodafone network. (Wikipedia)

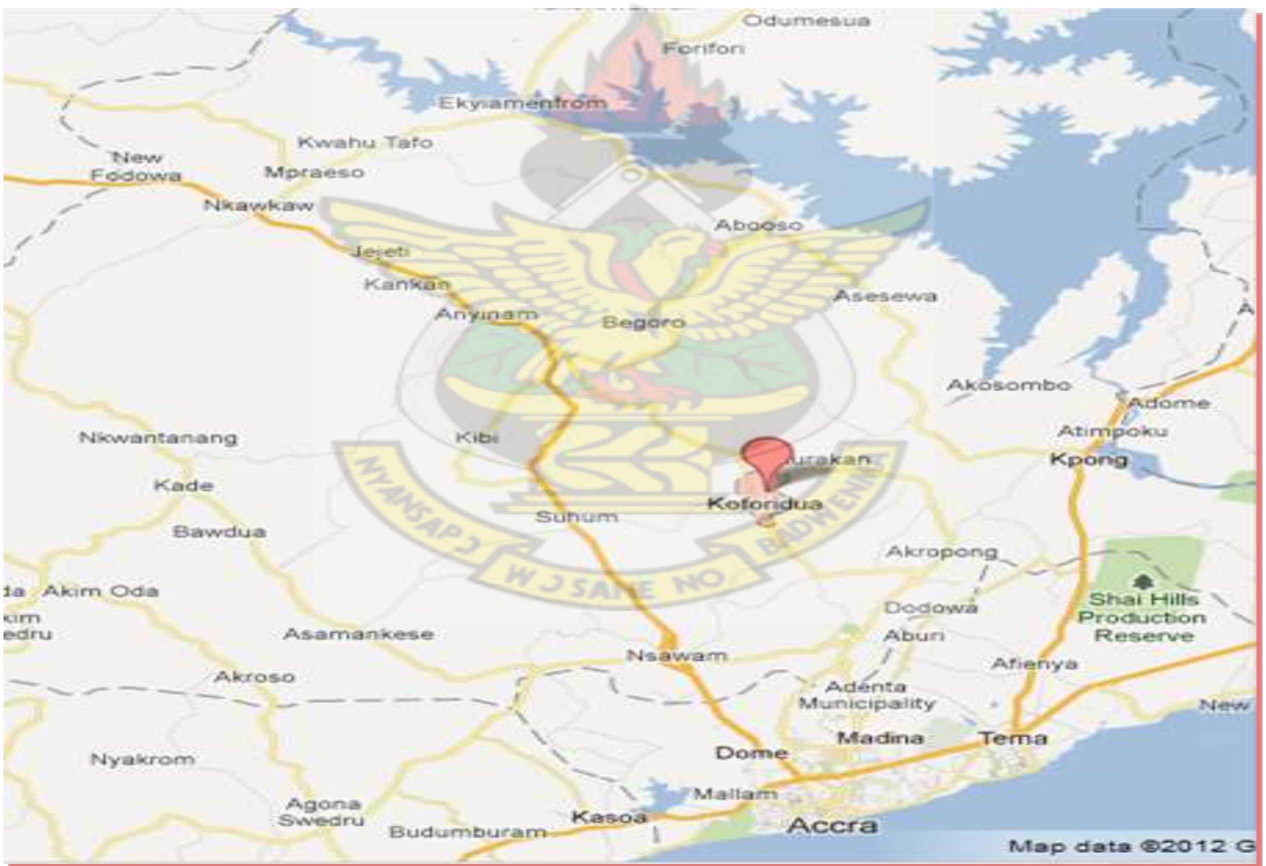

Figure 1.1 Map of New Juabeng Municipality Eastern Region .(Google images)

The use of Base Transceiver Station provides a fast establishment and expansion of the connection between operator and end-user The Base Transceiver Station is logically connected to the mobile stations through the air interface and one base station can be connected to a number of end-users (Point to Multipoint access - PMP). One base station can host up to 6 sectors and each sector has a capacity of 37Mbps (Megabits per second Gross bit rate full duplex. One sector covers the end-users in an area within an angle of $90^0$with a maximum transmission range at approximately 5 km. This means that a base station with 4 sectors has a total potential coverage area that can be approximated by a circle with centre at the base station and a maximum radius of five kilometers (5 km).

The major challenge in Telecommunication Network Planning is to identify the location of Base Stations for optimal visitation. This enhances effective preventive and routine maintenance on the BTS equipment, fault restoration, installations, refueling of standby plant, BTS integration , running pre-acceptance test and for the commission of (BTS).

The Vodafone Network in the New Juabeng Municipality has twenty four (24) Base transceiver Station sites. The BTS location and coordinates are as shown in Table 1.1.

Table 1.1: BTS location and number allocated in the New Juabeng municipality

NUMBER ALLOCATED	BTS LOCATION	NUMBER ALLOCATED	
		Longitude	Latitude
1	Koforidua radio station	-0.243330	6.088360
2	Nyerede Adawso	-0.256840	6.063335
3	Adukrom	-0.075240	6.020460
4	Somanya	-0.014630	6.104280
5	Huhunya_Ex	-0.167640	6.174700
6	Krobo Odumasi	-0.000740	6.134670
7	Akuse_Ex	-0.114660	6.093550
8	Akosombo_2	-0.054870	6.288470
9	Asesewa	-0.147490	6.401820
10	Begoro	-0.378790	6.376920
11	Kukurantumi	-0.370831	6.185835
12	New Tafo	-0.372489	6.232176
13	Apedwa	-0.488680	6.112400
14	Kibi	-0.554130	6.170800
15	Akim-Asafo	-0.473681	6.178264
16	Asiakwa	-0.496510	6.266440
17	Osino	-0.480660	6.341860
18	Akrade	-0.077105	6.203555
19	Koforidua_Ex_1	-0.259600	6.091700
20	Koforidua_Ex_2	-0.254270	6.096790
21	Koforidua_North	-0.256996	6.078823
22	Koforidua_High-Court	-0.027965	6.129740
23	Akosombo radio station	-0.040190	6.244540
24	Juapong	-0.134800	6.250090

1.2 Statement of the Problem

Visitation to Base transceiver stations Site is one the major task affecting the Vodafone Operation within the Eastern Region of Ghana, as a result, aiding to a lot of outages within the network and also impede the preventive maintenance of Base Transceiver Stations, increasing mean time to restored (MTTR), fault incident rate and refueling of standby plants.

The town and country planning Survey report for (2010-2011) reveals that the Eastern Region has land sides of 19,323 square kilometers, occupying 8.1 per cent of the total land area of Ghana and is the sixth largest region in the country.

Taking the optimal route by visiting all the Base transceiver Station Site is one sure way by which cost can be reduced. This is what this thesis seeks to

1.3 Objectives of the Study

The objectives of this thesis are:

(i) To formulate a mathematical model that takes into consideration the actual distance between the twenty four (24) Base Transceiver Station Site within the New Juabeng Municipality

(ii) To determine the optimal route for visiting the entire twenty four (24) Base Transceiver Station Site within the New Juabeng Municipality to minimize travelling cost, time and distance for Vodafone maintenance Engineers and entire sub-contractors working on the BTS site. Especially Huawei, Eton and ICK Company.

1.4 Justification of Study

This study introduces a more proactive approach in the dealing with the visitation of Base transceiver station (BTS) site. An algorithm that proposes the optimal visitation of BTS Site, will help increase a faster response to;

(i) Faults and help increase customer satisfaction.

(ii) Enhance effective preventive Maintenance on the Base Transceiver Station Site.

(iii) Refueling Base Transceiver Station Site Standby Plant (BTS).

(iv) Co-location of the Base Transceiver Station Site (BTS).

(v) Integration Base Transceiver Station Site (BTS).

(vi) Pre- Acceptance Test of Base Transceiver Station Site (BTS).

(vii) Commission of Base Transceiver Station Site (BTS).

1.5.0 Methodology

In this research work, realistic mathematical models for the visiting of Base transceiver station site are formulated and solved using the Meta-heuristics, Simulated Annealing Algorithm. It goes beyond developing Meta-heuristic to solve simple strategies to optimize the visitation tour.

The idea is to create a core program that, with the correct input data files is able to assist in the optimal tour. The visualization of the result is made using function developed in the program Matlab. The sources of data for the thesis are the internet and libraries for relevant literature, Vodafone Company Ghana on current information on Base Transceiver Station (BTS) and New Juabeng Municipal Assembly was also consulted for information on the demarcation as well as distances of the network routes between suburbs, towns within the municipality and the region.

1.5.1 Synopsis of some heuristics algorithms

These are adhoc, trial-and-error methods which do not guarantee to find the optimal solution but are designed to find near-optimal solutions in a fraction of the time required by optimal methods. A heuristic is typically a simple intuitively designed procedure that exploits the problem structure and does not guarantee an optimal solution. Because most of practical problems and many interesting theoretical problems are *NP*-hard, heuristics and approximation algorithms play an important role solving high level optimization problems.

Such algorithms are used to find suboptimal solutions when the time or cost required to find an optimal solution to the problem would be very large. A meta-heuristic ("meta" means "beyond") is a general high-level procedure that coordinates simple heuristics and rules to find good approximate (or even optimal) solutions to computationally difficult combinatorial optimization problems. A meta-heuristic does not automatically terminate once a locally optimal solution is found, (Keuthen, 2003).

1.5.2 Greedy heuristics

These are simple iterative heuristics specifically designed for a particular problem structure. A greedy heuristic starts with either a partial or infeasible solution and then constructs a feasible solution step by step based on some measure of local effectiveness of the solutions. In each iteration, one or more variables are assigned new values by making greedy choices. The procedure stops when a feasible solution is generated. As an extension of greedy heuristics, a large number of local search approaches have been developed to improve given feasible solutions, (Tian and Yang, 1993).

1.5.3 Lagrangian heuristics

This exploits the solution process of the Lagrangian dual in order to obtain feasible solutions to the original problem. Most Lagrangian heuristics proposed so far attempt to make the optimal solution to the Lagrangian relaxation feasible, e.g., by means of a simple heuristic, (Fisher, 1981).

1.5.4 Local search.

This is a family of methods that iteratively search through the set of solutions. Starting from an initial feasible solution, a local search procedure moves from one solution optimal within a neighboring set of solutions; this is in contrast to a global optimum, which is the optimal solution in the whole solution space solution to a neighboring solution with a better objective function until a local optimum is found or some stopping criteria are met. The next two algorithms simulated annealing and tabu search, enhance local search mechanisms with techniques for escaping local optima,
(Kolohan and Liang, 2000).

1.5.5 Simulated annealing

This is a probabilistic meta-heuristic derived from statistical mechanics (Bradley Buckham and Casey Lambert, 1999). This iterative algorithm simulates the physical process of annealing, in which a substance is cooled gradually to reach a minimum-energy state. The algorithm generates a sequence of solutions and the best among them becomes the output. The method operates using the neighborhood principle, i.e., a new solution is generated by modifying a part of the current one and evaluated by the objective function (corresponding to a lower energy level in physical annealing). The new solution is accepted if it has a better objective function value. The algorithm also allows occasional non-improving moves with some probability that decreases over time, and depends on an algorithm parameter and the amount of worsening. A non-improving move means to go from one solution to another with a worse objective function value. This type of move helps to avoid getting stuck in local optimum. It has been proved that with a sufficiently large number of iterations and a sufficiently small final temperature, the simulated algorithm converges to a global optimum with a probability close to one. However, with these requirements, the convergence rate of the algorithm is very low. Therefore, in practice it is more common to accelerate the algorithm performance to obtain fast solution approximations, (Tian and Yang, 1993).

1.5.6 Tabu search

This is a meta-heuristic technique that operates using the following neighborhood principle. To produce a neighborhood of candidate solutions in each iteration, a solution is perturbed a number of times by rules describing a move. The best solution in the neighborhood replaces the current solution. To prevent cycling and to provide a mechanism for escaping locally optimal solutions, some moves at one iteration may be classified as tabu if the solutions or their parts, or attributes, are in the tabu list (the short-term memory of the algorithm), or the total number of iterations with certain attributes exceeds a given maximum (long-term memory). There are also aspiration criteria which override the tabu moves if particular circumstances apply, (Kolohan and Liang, 2000).

1.5.7 Genetic algorithm

These are probabilistic meta-heuristics that mimic some of the processes of evolution and natural selection by maintaining a population of candidate solutions, called individuals, which are represented by strings of binary genes. A genetic algorithm starts with an initial population of possible solutions and then repeatedly applies operations such as crossover, mutation, and selection to the set of candidate solutions. A crossover operator generates one or more solutions by combining two or more candidate solutions, and a mutation operator generates a solution by slightly perturbing a candidate solution. Thus, the population of solutions evolves via processes which emulate biological processes. The basic concept is that the strong species tend to adapt and survive while the weak ones tend to die out, (Goldberg, 1989).

1.5.8 The Travelling Salesman Problem

The Travelling Salesman Problem (TSP) is a problem in combinatorial optimization studied generally in some field of engineering, operations research and computer science. Given a number of nodes/location/ports/cities and their pair wise distances, the major task is to find the shortest possible tour that visits each node/location/port/city exactly once.

The Travelling Salesman Problem (TSP) has been studied during the last five decades and many exact and heuristic algorithms have been proposed and used to solve problem which otherwise have no direct ways of having an optimal solutions. Notable among such algorithms used include construction algorithms, iterative improvement algorithms, branch-and-cut exact branch-and-bound and algorithms and many metaheuristic algorithms, such as tabu search (TS), simulated annealing (SA), genetic algorithm (GA) and ant colony (AC), (Lin and Kernighan, 1973).

1.6.0 The Scope of the Study

The study partly parallels with the travel salesman problem but more closely, it mimics a type of the Chinese Postman Problem (CPP), the heuristic procedure consists of cluster first, route second method.

1.7 Limitations of the study

Unplanned nature of the town makes the determination of the distances very difficult.

The limited time at the researcher‘s disposal lead to few document being reviewed as a literature. The quality of data depends not only on the amount of time one spends in gathering them but partially on how much money one is prepared to spend in gathering them.

The researcher also encountered certain difficulties in connection with data collection. Other information, like the distance at a particular point with reference to a given geographical direction, etc.

1.8 Organization of the thesis

Chapter one covers the introduction to the thesis. In chapter two, we shall put forward relevant literature on simulated Annealing. Chapter three denoted for the methodology of the study. Chapter four presents data collection and analysis. Chapter five deals with the conclusion and recommendation of the study.

1.9 Summary

In this chapter, we presented brief history of Vodafone Ghana, background to the study, statement of the problem, objectives of the study, methodology, importance of a visit to the Base Transceiver Station site and the organization of the thesis.

In the next Chapter, we shall put forward relevant literature on simulated annealing on the Travelling Salesman problem.

CHAPTER TWO

LITERATURE REVIEW

The Travelling Salesman Problem (TSP) is a problem in combinatorial optimization studied in operations research and theoretical computer science. Given a list of cities and their pairwise distances, the task is to find the shortest possible tour that visits each city exactly once.

The Travelling Salesman Problem (TSP) has been studied during the last fifty years and many exact and heuristic algorithms have been developed. These algorithms include construction algorithms, iterative improvement algorithms, branch-and-bound and branch-and-cut exact algorithms and many metaheuristic algorithms, such as Simulated Annealing (SA), Tabu search (TS), Ant Colony (AC) and Genetic Algorithm (GA).

Some of the well-known tour construction procedures are the nearest neighbor procedure by Rosenkratz et al, (1993) and the Clark and Wright, (1974), savings algorithm, the insertion procedures, the partitioning approach by Karp and the minimal spanning tree approach by Christotides.

The branch exchange is perhaps the best known iterative improvement algorithm for the TSP. The 2-opt and 3-opt heuristics described in Lin. Lin and Kernighan, (1973) made a great improvement in quality of tours that can be obtained by heuristic methods. Even today, their algorithm remains the key ingredient in the successful approaches for finding high quality tours and is widely used to generate initial solutions for other algorithms or developed a simplified edge exchange procedure requiring only Q (n^2)

operations at each step , but producing tour nearly as good as the average performance of 3-opt algorithm.

One of the earliest exact algorithms is due to (Dantzig, 1954) in which linear programming (LP) relaxation is used to solve the integer formulation by suitably chosen linear inequality to the list of constraints continuously. Branch and bound (B & B) algorithm are widely used to solve the TSP‘s. Several authors have proposed B &B algorithm based on assignment problem (AP), relaxation of the original TSP formulation. These authors include (Eastman, 1958), (Held and Karp, 1970).

Besides the above mentioned exact and heuristic algorithms, metaheuristic algorithms have been applied successfully to the TSP by a number of researchers. SA algorithms for the TSP were developed by Bonomi and Lutton, Golden and Skiscim and Nahr et al. Lo and Hus etc. Tabu search metaheuristic algorithms for TSP have been proposed by Knox and Fiechter. The AC is a relative new metaheuristic algorithm, which is applied successfully to solve the TSP.

(Applegate et. al 1994) solved a traveling salesman problem which models the production of printed circuit boards having 7,397 holes (cities) and in (1998) the same authors solved a problem over the 13,509 largest cities in the U.S. For problems with large number of nodes as cities the TSP becomes more difficult to solve.

In Homer's Ulysses problem of a 16 city traveling salesman problem, one finds that there are 653,837,184,000 distinct routes, (Grötschel and Padberg, 1993). Enumerating all such roundtrips to find the shortest one took 92 hours on a powerful workstation.

The TSP and its solution procedures have continued to provide useful test grounds for many combinatorial optimization approaches. Classical local optimization techniques Rossman,(1958) ; Applegate,(1999) ;Riera-Ledesma,(2005) ;Walshaw,(2002) ; Walshaw, (2001) as well as many of the more recent variants on local optimization, such as simulated annealing by Tian and Yang, (1993), tabu search by Kolohan and Liang, (2003) neural networks by Potvin, (1996) and genetic algorithms have all been applied to this problem, which for decades has continued to attract the interests of researchers.

Although a problem statement posed by Karl Menger on February 5, 1930, at a mathematical colloquium in Vienna, is regarded as a precursor of the TSP, it was Hassle Whitney, in 1934, who posed the traveling salesman problem in a seminar at Princeton University, (Flood, 1956).

In 1949 Robinson, with an algorithm for solving a variant of the assignment problem is one of the earliest references that use the term "traveling salesman problem" in the context of mathematical optimization. (Robinson, 1949), However, a breakthrough in solution methods for the TSP came in 1954, when Dantzig, (1954) applied the simplex method (designed by George Dantzig in, 1947) to an instance with 49 cities by solving the TSP with linear programming.

There were several recorded contributions to the TSP in 1955. Heller, (1955) discussed linear systems for the TSP polytope, and some neighbor relations for the asymmetric TSP polytope. Also Kuhn, (1955) announced a complete description of the 5-city asymmetric TSP polytope. Morton and Land, (1955) presented a linear programming approach to the TSP, alongside the capacitated vehicle routing problem. Furthermore, Robacker, (1955) reported manual computational tests of some 9 cities instance using the Dantzig-Fulkerson-Johnson method, with average computational times of about 3 hours. This time became the benchmark for the next few years of computational work on the TSP (Robacker, 1955).

Flood (1956) discussed some heuristic methods for obtaining good tours, including the nearest-neighbor algorithm and 2-opt while Kruskal, (1956) drew attention to the similarity between the TSP and the minimum-length spanning trees problem. The year 1957 was a quiet one with a contribution from Barachet, (1957) described an enumeration scheme for computing near-optimal tours.

Croes, (1958) proposed a variant of 3-opt together with an enumeration scheme for computing an optimal tour. He solved the Dantzig-Fulkerson-Johnson 49-city example in 70 hours by hand. He also solved several of the Robacker examples in an average time of 25 minutes per example. Bock, (1958) describes a 3-opt algorithm together with an enumeration scheme for computing an optimal tour. The author tested his algorithm on some 10-city instance using an IBM 650 computer.

By 1958, work related to the TSP had become serious research to attract Ph.D. students. A notable work was a Ph.D. thesis Eastman, (1958) where a branch-and-bound algorithm using the assignment problem to obtain lower bounds was described. The algorithm was tested on examples having up to 10 cities. Also that same year, Rossman and Twery, (1958) solved a 13-city instance using an implicit enumeration while a step-by-step application of the Dantzig-Fulkerson-Johnson algorithm was also given for Barachet's 10-city example. Bellman, (1960) showed the TSP as a combinatorial problem that can be solved by dynamic programming method.

In Miller et al, (1960), an integer programming formulation of the TSP and its computational results of solving several small problems using Gomory's cutting-plane algorithm was reported.

Lambert, (1960) solved a 5-city example of the TSP using Gomory cutting planes. Dacey, (1960) reported a heuristic, whose solutions were on average 4.8 percent longer than the optimal solutions. TSP in 1960 achieved national prominence in the United States of America when Procter & Gamble used it as the basis of a promotional context. Prizes up to $10,000.00 were offered for identifying the most correct links in a particular 33-city problem. A TSP researcher, Gerald Thompson of Carnegie Mellon University won the prize in Applegate et al, (2007).

Müuller- Merbach, (1961) proposed an algorithm for the asymmetric TSP; he illustrated it on a 7-city example.

Ackoff et al, (1961) gave a good survey of the computational work on the TSP that was carried out in the 1950‘s. By 1962, when the computer was becoming a useful tool in exploring TSP, the dynamic programming approach gained attention. Gonzales solved instances with up to 10 cities using dynamic programming on an IBM 1620 computer by Gonzales, (1962), similarly, Held and Karp, (1962) described a dynamic programming algorithm for solving small instances and for finding approximate solutions to larger instances.

Little et al, (1963) coined the term branch-and-bound. Their algorithm was implemented on an IBM 7090 computer and they gave some interesting computational tests including the solution of a 25-city problem that was in the Held and Karp test set. Their most cited success is the solution of a set of 30-city asymmetric TSPs having random edge lengths.

In an important paper, (Lin, 1965) a heuristic method for the TSP was published. The author defined *k*-optimal tours, and gave an efficient way to implement 3-opt, extending the work of Croes, (1958) with computational results given for instances with up to 105 cities.

The year 1966 was another fruitful one for the TSP in terms of published works. Roberts and Flores, (1966) described an enumerative heuristic and obtained a tour for Karg and Thompson's 57-city example, having cost equal to the best tour found by Karg and Thompson. Also, in a D.Sc. thesis at Washington University, St. Louis, Shapiro, (1966) describes an algorithm similar to Eastman's branch-and-bound algorithm. Gomory1966 gave a very nice description of the methods contained in Dantzig et al, (1954), Held and Karp, (1962) and Little et al, (1963). Similarly, in Lawler and Wood, (1966) descriptions of the branch-and-bound algorithms of Eastman, (1958) and Little et al, (1963) were given. The authors suggested the use of minimum spanning trees as a lower bound in a branch-and-bound algorithm for the TSP.

Bellmore and Nemhauser, (1968) presented an extensive survey of algorithms for the TSP. They suggested dynamic programming for TSP problems with 13 cities or less, Shapiro‘s branch-and-bound algorithm for larger problems up to about 70-100 and Shen Lin's `3-opt' algorithm for problems that cannot be handled by Shapiro's algorithm. Raymond, (1969) is an extension to Karg and Thompson‘s, (1964) heuristic for the TSP where computational results were reported for instances having up to 57 cities.

Held and Karp in there, (1970) paper introduced the 1-tree relaxation of the TSP and the idea of using node weights to improve the bound given by the optimal 1-tree. Their computational results were easily the best reported up to that time. Another notable work on the TSP in the 70s is the Hong, Ph.D. Thesis, at The Johns Hopkins University in 1972 written under the supervision of Bellmore, and the work was the most significant computational contribution to the linear programming approach to the TSP since the original paper of Dantzig et al, (1959).

The Hong‘s algorithm, (1972) had most of the ingredients of the current generation of linear-programming based algorithms for the TSP. The author used a dual LP algorithm for solving the linear-programming relaxations; he also used the Ford-Fulkerson max-flow algorithm to find violated subtour inequalities.

The algorithm of Held and Karp, (1971) was the basis of some major publications in 1974. In one case, Hansen and Krarup, (1974) tested their version of Held-Karp, (1971) on the 57-city instance of Karg and Thompson, (1964) and a set of instances having random edge lengths. In 1976 a linear programming package written by Land and Powell was used to implement a branch-and-cut algorithm using subtour inequalities. Computational results for the 48-city instance of Held and Karp and the 57-city instance of Karg and Thompson, (1964) were given.

Smith and Thompson, (1977) presented some improvements to the Held-Karp algorithm tested their methods on examples which included the 57-city instance of Karg and Thompson 1964 and a set of ten 60-city random Euclidean instances. In 1979, Land described a cutting-plane algorithm for the TSP. The decade ended with a survey on algorithms for the TSP and the asymmetric TSP in Buckard, (1979).

A very impressive work heralded the 1980s. Crowder and Padberg, (1980) gave the solution of a 318-city instance described in Lin and Kernighan, (1973). The 318-city instance would remain until 1987 as the largest TSP solved. Also, in 1980, Grötschel gave the solution of a 120-city instance by means of a cutting-plane algorithm, where subtour inequalities were detected and added by hand to the linear programming relaxation in Grötschel, (1980).

In 1982, Volgenant and Jonker described a variation of the Held-Karp algorithm, together with computational results for a number of small instances by Volgenant and Jonker, (1982). A very important work of 1985 is a book (Lawler et al., 1985) containing several articles on different aspects of the TSP as an optimization problem. Padberg and Rinaldi, (1987) solved a 532-city problem using the so-called branch and cut method.

The approach for handling the subtours elimination constraints of the TSP integer LP is another area for re-examination. Researchers have identified the issue of feasibility or subtour elimination as very crucial in the formulation of the TSP or similar permutation sequence problem. —No one has any difficulty understanding subtours, but constraints to prevent them are less obvious, says Radin, (1998). Methodologies or theoretical basis for handling these constraints within the context of algorithm development has been the basis of many popular works on the TSP.

A classical example of this approach is in Crowder and Paderg, (1980) where a linear programming relaxation was adopted such that if the integral solution found by this search is not a tour, then the subtour inequalities violated by the solution are added to the relaxation and resolved.

Grötschel, (1980) used a cutting-plane algorithm, where cuts involving subtour inequalities were detected and added by hand to the linear programming relaxation. Hong, (1972) used a dual LP algorithm for solving the linear-programming relaxations, the Ford-Fulkerson max-flow algorithm, for finding violated subtour inequalities and a branch-and-bound scheme, which includes the addition of subtour inequalities at the nodes of the branch-and-bound tree. Such algorithms are now known as "branch-and-cut". The problem of dealing with subtour occurrences algorithm development has been a major one in the in the TSP studies in the literature.

The works in the 1990‘s were mostly application in nature. A large number of scientific/engineering problems and applications such as vehicle routing, parts manufacturing and assembly, electronic board manufacturing, space exploration, oil exploration, and production job scheduling, etc. have been modeled as the Machine Setup problem (MSP) or some variant of the TSP are found in (Al-Haboub-Mohamad and Selim Shokrik, (1993), Clarker and Ryan, (1989), Crama et al, (2002), Ferreir, (1995), Foulds and Hamacher, (1993), G¨unther et al, (1998), Keuthen, (2003), Kolohan and Liang, (2000), Mitrovic-Minic and Krishnamurti, (2006).

One of the ultimate goals in computer science is to find computationally feasible exact solutions to all the known NP-Hard problems; a goal that may never be reached. Feasible exact solutions for the TSP have been found, but there are restrictions on the input sizes. An exact solution was found for a 318-City problem by Crowder and Padberg in, (1980). The basic idea in achieving this solution involves three phases. In the first phase, a true lower bound on the optimal tour is found. In the second phase, the result in the first phase is used to eliminate about ninety-seven percent of all the possible tours. Thus, only about three percent of the possible tours need to be considered. In the third phase, the reduced problem is solved by brute force. This solution has been implemented and used in practice. Experimental results by Apple Gate et al, (1998) showed that running this algorithm, implemented in the C programming language and executed on a 400MHz machine, would produce a result in 24.6 seconds of running time.

Other exact solutions have been found, as mention in, (1998), a 120-city problem by Grötschel, (1980), a 532-city problem by Padberg and Rinaldi, (1987). However, none of the algorithms that provide an exact solution for input instances of over a thousand cities are practical for everyday use. Even with today's super computers, the execution time of such exact solution algorithms for TSPs involving thousands of cities could take days.

Computer hardware researchers have been making astonishing progress in manufacturing evermore powerful computing chips. Moores Law in (http://en.wikipedia.org/wiki/Moore's_law), which states that the number of transistors that can fit on a chip will double after every 18 months, has held ground since, (1965). This basically means that computing power has doubled every 18 months since then. Thus, we have been able to solve larger instances of NP-hard problems, but algorithm complexity has still remained exponential.

Moreover, it is highly speculated that this trend will come to an end because there is a limit to the miniaturization of transistors. Presently, the sizes of transistors are approaching the size of atoms. With the speeds of computer processors rounding the 5GHz mark, and talks about an exponential increase in speeds of up to 100GHz (http://en.wikipedia.org/wiki/Moore's_law), one might consider the possibility of us exceeding any further need of computational performance. However, this is not the case. Although computing speeds may increase exponentially, they are, and will continue to be, surpassed by the exponential increase in algorithmic complexity as problem sizes continue to grow. Moore‘s law may continue to hold true for another decade or so, but different methods of computing are being researched.

2.2 Summary

In this chapter, we put forward relevant literature on travel salesman program (TSP) and some meta-heuristic algorithms.

In the next chapter we shall consider the methodology of the study.

CHAPTER 3

METHODOLOGY

The travelling salesman problem is combinatorial optimization problem. According to Hillier and Lieberman, (2005) it has been given this picturesque name because it can be described in terms of a salesman (or saleswoman) who must travel to a number of cities during one tour. Starting from his or her home city, the salesman follows to visit each city exactly once before returning to his home city as to minimize the total length of the tour.

The figure below shows an example of a small travelling salesman problem with seven cities.

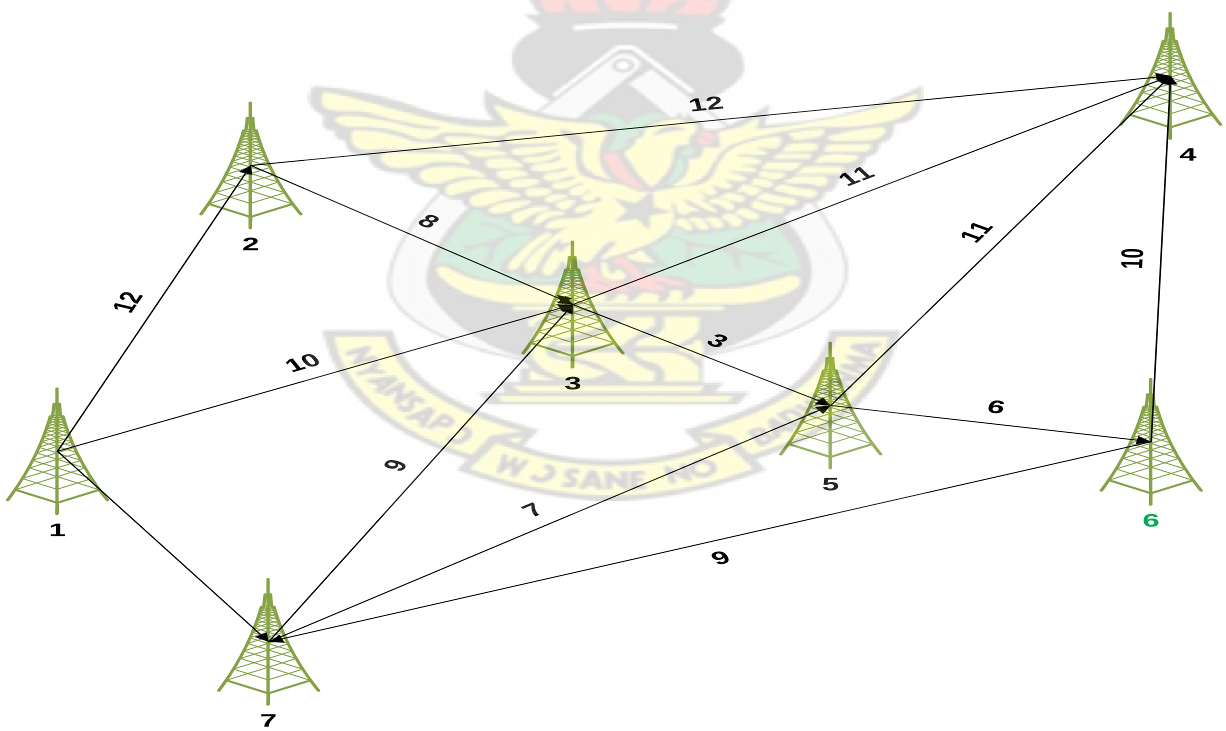

Figure 3.1 Traveling Salesman Problem

City 1 being the salesman's home city, he starts from this city, and must choose a route to visit each of the other cities exactly once before returning to city 1. The number next to each link between each pair of cities represents the distance between
these cities. The objective is to determine which route will minimize the total distance that the salesman must travel. The Sub-tour Reversal Algorithm, the Tabu Search and the simulated annealing will each be used to find the optimal solution for the travelling salesman problem above later in this chapter.

There have been a number of applications of travelling salesman problems that have nothing to do with salesmen. For example, when a presidential aspirant leaves his home city and visits a number of cities campaigning and returns to his home city after a period, the problem of determining the shortest route for doing this tour is a travelling salesman problem. Another example involves the manufacture of printed circuit boards for wiring chips and other components. When many holes need to be drilled into a printed circuit board, the process of finding the most efficient drilling sequence is a travelling salesman problem.

The difficulty of travelling salesman problems increases rapidly as the number of cities increases. For a problem with n cities, the number of feasible routes to be considered is $(n-1)!/2$ since there are $(n-1)$ possibilities for the first city after the home city, $(n-2)$ possibilities for the next city, and so forth. The denominator of 2 arises because every route has an equivalent reverse route with exactly the same distance. Thus, while a 10-city travelling salesman problem require less than 200,000 feasible solutions to be considered, a 20-city problem has roughly 10^{16} feasible solutions while a 50-city problem has about 10^{62} feasible solutions.

3.1 Formulation of TSP model

The problem can be defined as follows: Let $G = (V,E)$ be a complete undirected graph with vertices V, $|V|$=n, where n is the number of cities, and edges E with edge length d_{ij} for (i,j). The focus is on the symmetric TSP in which case $d_{ij} = d_{ji}$ for all (i,j). This minimization problem can be formulated as an integer programming as shown below in Equations (1) to (5). The problem is an assignment problem with additional restrictions that guarantee the exclusion of sub-tours in the optimal solution. Recall that a sub-tour in V is a cycle that does not include all vertices (or cities). Equation (1) is the objective function, which minimizes the total distance to be travelled.

Constraints (2) and (3) define a regular assignment problem, where (2) ensures that each city is entered from only one other city, while (3) ensures that each city is only departed to on other city. Constraint (4) eliminates sub-tours. Constraint (5) is a binary constraint, where x_{ij} = 1 if edge (i,j) in the solution and x_{ij} = 0, otherwise.

$$\min \sum_{i \in v} \sum_{j \in v} c_{ij} x_{ij} \qquad (1)$$

$$\sum_{\substack{j \in v \\ j \neq i}} x_{ij} = 1 \qquad i \in v \qquad (2)$$

$$\sum_{\substack{i \in v \\ i \neq j}} x_{ij} = 1 \qquad j \in v \qquad (3)$$

$$\sum_{i \in s} \sum_{j \in s} x_{ij} \leq [s] - 1 \qquad \forall \mathrm{s} \subset \mathrm{v}, \mathrm{s} \neq \varnothing \qquad (4)$$

$$x_j \;\; or \;\; 1 \qquad i, j \in v \qquad (5)$$

However, the difficulty of solving TSP is that sub-tour constraints will grow exponentially as the number of cities grows large, so it is not possible to generate or store these constraints. Many applications in real world do not demand optimal solutions.

3.2 The Sub-Tour Reversal Algorithm

This adjusts the sequence of cities visited in the current trial solution by selecting a sub-sequence of the cities and simply reversing the order in which that sequence of cities is visited.

Initialization: Start with any feasible tour as the initial trial solution.

Iteration: For the current trial solution, consider all possible ways of performing a sub-tour reversal except reversal of the entire tour. Select the one that provides the largest decrease in the

Stopping Rule: Stop when no sub-tour reversal will improve the current trial solution. Accept this solution as the final solution.

Applying this algorithm to the problem above and starting with **1 – 2 – 3 – 4 – 5 – 6 – 7 - 1** as the initial trial solution, there are four possible sub-tour reversals that would improve upon this solution as shown below

3.3 Tabu Search

According to Hillier and Lieberman Tabu Search is a widely used metaheuristic that uses some common sense ideas to enable the search process to escape from a local optimum. The concept of tabu search (TS) is derived from artificial intelligence where intelligent use of memory helps in exploiting useful historical information. The restrictions put on the information in the memory reminiscent of the definition of the word 'tabu' as a set apart as charged with a dangerous supernatural power and forbidden to profane use or contact.. Tabu search can also incorporate some more advanced concepts. One is intensification, which involves exploring a portion of the feasible region more thoroughly than usual after it has been identified as a particularly promising portion for containing very good solutions. Another concept is diversification, which involves forcing the search into previously unexplored areas of the feasible region. The focus will however be on the basic form of tabu search summarized below.

(i) **Initialization** : A starting solution generated by choosing a random solution, $x \in S$. The evaluating function f(x) is used to evaluate x. The solution is stored in the algorithm memory called the tabu list.

(ii) Neighborhood exploration: All possible neighbours $\mu(x)$ of the solution x are generated and evaluated. Solutions in the tabu list are considered unreachable neighbours, they are taboo (tabu). An immediate neighbor can be reached by making a sub-tour reversal.

(iii) New Solution: A new solution is chosen from the explored neighborhood. This solution should not be found in the tabu list before it is discovered and has to have the best move evaluation value of f(x) for all reachable neighbors of x.

Do tabu check on the new solution. If it is successful replace the current

solution and update the tabu list and other tabu attributes. Here the new solution evaluation value can be worse compared with that of current solution. This enables the solution not to be trapped at local optimum.

The tabu check applied based on the move being the best move.

(i) If the solution is in the tabu list then check the aspiration level. If successful replace the current solution and update the tabu list and other tabu attributes. The aspiration check uses the function evaluation and the success of the check depends on the function evaluation of the new solution being better than that of the current best solution.

(ii) If checks **(i)** and **(ii)** are not successful then keep the current solution otherwise replace the current solution by the new solution.

(iii) Compare the best solution to the current solution, if the current solution is better than the best solution, replace the best solution.

(iv) Until loop condition is satisfied go to step Until termination condition is satisfied go to step1.

(v) Stop after three consecutive iterations without an improvement in the best objective function value. Also stop at any iteration where the current trial solution has no immediate neighbours that are not ruled out by their tabu status.

To apply this tabu search algorithm to the problem above,

Let initial trial **solution = 1 – 2 – 3 – 4 – 5 – 6 – 7 - 1** Distance **= 69**

Tabu list : Blank at this point

Iteration 1:

reverse 3 - 4

Delete Links: **2 - 3 and 4 - 5**

Added links: **2 - 4 and 3 - 5**

Tabu list : Links **2 - 4** and **3 - 5** :New trial solution: **1 – 2 – 4 – 3 – 5 – 6 – 7 - 1** Distance **= 65**

Iteration 2

Reverse 3-5-6

Delete links: 4 - 3 and 6 - 7

Added links: 4 - 5 and 3 - 7

Tabu list: links 2 - 4, 3 - 5, 4 - 6 and 3 - 7

New trial solution: 1 – 2 – 4 – 6 – 5 – 3 – 7 - 1 Distance = 64

The tabu search algorithm now escapes from this local optimum by moving next to the best immediate neighbor of the current trial solution even though its distance is longer. Considering the limited availability of links between pairs of cities in fig….., the current trial solution has only the two immediate neighbors listed below.

Reverse 6 – 5 - 3: 1 – 2 – 4 – 3 – 5 – 6 – 7 - 1 Distance = 65

Reverse 3-7: 1 – 2 – 4 – 6 – 5 – 7 – 3 - 1 Distance = 66

Reversing **2 – 4 – 6 – 5 – 3 - 7** to obtain **1– 7 – 3 – 5 – 6 – 4 – 2 - 1** is ruled out since it is simply the same tour in the opposite direction. However the of these immediate neighbours must be ruled out because it would require deleting links **4 - 6 and 3 - 7,** which is tabu since both of these links are on the tabu list. This move could still be allowed if it would improve upon the best trial solution found so far but it does not.

Ruling out this immediate neighbor does not allow cycling back to the preceding trial solution. Therefore by default, the second of these immediate neighbours is chosen to be the next trial solution as summarized below.

Iteration 3

Reverse 3 - 7

Delete links: 5 - 3 and 7 - 1

Add links: 5 - 7 and 3 - 1

Tabu List: 4 - 5, 3 - 7, 5 - 7 and 3 - 1

New trial solution: 1 – 2 – 4 – 6 – 5 – 7 – 3 - 1 Distance = 66

The sub- tour reversal for this iteration can be seen in the fig……., where the dashed lines show the links being deleted (on the left) and added (on the right) to obtain the new trial solution.

The new trial solution has the four immediate neighbours listed below.

Reverse **2 – 4 - 6 – 5 - 6: 1**- 7 – **5 – 6 – 4 – 2 – 3 - 1** Distance = **65**

Reverse **6 - 5: 1** – 2 – 4 – 5 – **6 – 7 – 3 - 1** Distance **= 69**

Reverse **5 - 7: 1 – 2 – 4 – 6 – 5 – 7 – 3 - 1** Distance **= 63**

Reverse **7 - 3: 1 – 2 – 4 – 6 – 5 – 3 – 7 - 1**

Both of the deleted links **4 - 6** and **5 - 7** are on the tabu list. The second of these immediate neighbours is therefore tabu. The fourth immediate neighbor is also tabu. Thus, there are only two options, the first and the third immediate neighbours. The third immediate neighbor is chosen since it has shorter distance.

Iteration 4

Reverse 5 - 7

Delete links: 6 - 5 and 7 - 3

Add links**: 6 - 7** and **5 - 3**

Tabu list: **5 - 7, 3 - 1, 6 - 7** and **5 - 3**

(**4 - 6** and **3 - 7** are now deleted from the list)

New trial solution: **1 – 2 – 4 – 6 – 7 – 5 – 3 - 1** Distance **= 63**

The only immediate neighbor of the current trial solution would require deleting links **6 - 7** and **5 - 3,** both of which are on the tabu list so cycling back to the preceding trial solution is prevented. Since no other immediate neighbours are available, the stopping rule terminates the algorithm at this point with **1 – 2 – 4 – 6 – 7 – 5 – 3 - 1** as the final solution with Distance **= 63.**

3.4 Model assumptions

(i) The traffic situation does not affect the weight (distance) on each edge of the graph (street).

(ii)The model considers the weight of each edge of the graph in terms of distance instead of time.

(iii)Road conditions are the same between all the towns within which the BTS are located.

(iv) (OBTS which are less than one km are exempted and represented by the BTS centrally

3.5 Simulated Annealing

According to Amponsah and Darkwah (2007) the concept of Simulated Annealing is derived from Statistical mechanics in the area of natural sciences. A piece of regular metal in its natural state has the magnetic direction of its molecules aligned in uniform direction. As the metal is heated, the kinetic energy of the molecules increases and the cohesive force decreases till when the molecules are free to move about randomly. The magnetic directions of the molecules are oriented randomly.

To achieve regularity of alignment of the magnetic direction so as to make the metal stable for use, it must be cooled slowly. This slow cooling of the metallic material is called annealing.

In 1953 Metropolis and others recognised the use of Boltzmann's law to stimulate the efficient equilibrium condition of a collection of molecules at a given temperature and thus facilitate annealing. When the metal is heated to higher temperature and it is being cooled slowly it is assumed that for a finite drop in temperature the system state change in the sense that the molecules assume new configuration of arrangement. The configuration depends on parameters like temperature, the energy of the system and others. An energy function can be obtained by combining the parameters. In 1983 Kirk Patrick showed how Simulated Annealing of Metropolis could be adapted to solve problems in Combinatorial Optimization. The following analogy was made

(i) Annealing looks for system state at a given temperature.

(ii) Optimization looks for feasible solution of the combinatorial problems

(iii) Cooling of the metal is to move from one system state to another b) Search procedure (algorithm scheme) tries one solution after another in order to find the optimal solution.

(iv) Energy function is used to determine the system state and energy

(v) Objective (cost) function is used to determine a solution and the objective function value.

(vi) Energy results in evaluation of energy function and the lowest energy state corresponds to stable state.

(vii) Cost results in evaluation of objective function and the lowest objective function value corresponds to the optimal solution

(viii) Temperature controls the system state and the energy

(ix) A control parameter is used to control the solution generation and the objective function value

Simulated annealing (SA) is a generic probabilistic meta-heuristic for the global optimization problem of applied mathematics, namely locating a good approximation to the global minimum of a given function in a large search space. It is often used when the search space is discrete (e.g., all tours that visit a given set of cities). For certain problems, simulated annealing may be more effective than exhaustive enumeration provided that the goal is merely to find an acceptably good solution in a fixed amount of time, rather than the best possible solution.

3.5.1 Using simulated Annealing to solve TSP

The TSP was one of the first problems to which simulated annealing was applied, serving as an example for both Kirk Patrick et al. (1983) and Cerny (1985). Since then the TSP has continued to be a prime test bed for the approach and its variants. Most adaptations have been based on the simple schematic, with implementations differing as to their methods for generating starting solutions (tours) and for handling temperatures, as well as in their definitions of equilibrium, frozen, neighbor and random.

3.5.2 General scheme for a simulated annealing algorithm

(a) Generate a starting solution S and set the initial solution $S^* = S$

(b) Determine a starting temperature T.

(c) While not yet at equilibrum for this temperature, do the following:

(d) Choose a random neoghbor S^* of the current solution

(e) Set Δ = Length(S*), Length(S)

(f) if Length(S*) $\leq$ 0 (downhill move): Set $S = S^*$

(g) if Length(S) < Length(S*), set $S^* = S$

(h) if Length(S) < length(S*) (uphill move):

Choose a random number r uniformly from [0,1]

$if\ \ r < e^{\left(-\Delta/T\right)},\ set\ S = S^*$

i) End "while not yet equilibrum" loop.

(j) Lower the temperature T

(k) End "While not yet frozen loop"

(l) Return S^*

3.1 Prototype Example

Considering Figure 3.1

Taking the initial solution to be in the tour in the order:

1- 2 - 3 - 4 - 5 - 6 - 7 - 1 using the parameters;

$T_o = 20$, $T_{(k+1)} = \alpha T_k$, $\alpha = 0.5$

Stop when T<0.1

First Iteration

Assuming x^o = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 1

d(x^o) = d(1,2)+d(2,3)+d(3,4)+d(4,5)+d(5,6)+d(6,7)+(7,1) = 69

Using the sub-tour reversal as local search to generate the new solution

x^1 = 1 - 3 - 2 - 4 - 5 - 6 - 7 - 1

d(x^1) = d(1,3)+d(3,2)+d(2,4)+d(4,5)+d(5,6)+d(6,7)+d(7,1) = 68

δ=d(x^1)-d(x^0)

Since δ<0, set $x^0 \leftarrow x^1$

Updating the temperature T1= αT_0 = 0.5(20) = 10

Second Iteration

d(x^0) = 68

By the sub-tour reversal as local search to generate the new

solution 1 - 2 - 3 - 4 - 5 - 6 - 7 - 1

x^1= 1 - 2 - 3 - 4 - 5 - 6 - 7 - 1

d(x^1) = d(1,2)+d(2,3)+d(3,5)+(5,4)+(4,6)+d(6,7)+d(7,1) = 65

δ = d(x^1) - d(x^0) = 65 - 68 = 3

Since δ<0, set $x^0 \leftarrow x^1$

Updating the temperature, T_2 = 0.5(10) = 5

Third Iteration

d(x^0) = 65

Using the sub-tour reversal as local search to generate the new solution 1 - 2 - 3 - 4 - 6 - 5 - 7 - 1

x^1 = 1 - 2 - 3 - 4 - 6 - 5 - 7 - 1

d(x^1) = d(1,2)+d(2,3)+d(3,4)+d(4,6)+d(6,5)+d(5,7)+d(7,1) = 66

$\delta = d(x^1) - d(x^1) = 65 - 64 = 1$

Since $\delta > 0$, apply Boltzmann's condition m=$e^{\left(-\delta/T_2\right)}$ = 0.82

A random number will be generated from a computer say θ

If m > θ, then set $x^0 \leftarrow x^1$ otherwise $x^0 \leftarrow x^1$

Updating the temperature, T_3 = 0.5(5) = 2.5

This process will continue until the final temperature and the optimal solutions are obtained.

3.6 Genetic Algorithm

The genetic algorithm (GA) is an evolutionary algorithm inspired by Darwin, (1859) and recently discussed by Dawkins, (1986) .Holland 1975 invented Genetic Algorithm as an adaptive search procedure. Generalized chromosome genetic algorithm (GCGA) was proposed for solving generalized traveling salesman problems (GTSP).

Theoretically, the GCGA could be used to solve classical traveling salesman problem (CTSP) by (Yang, 2008).

3.6.1 Genetic Algorithm

The GA has the following simulations of the evolutionary principles;

Table 3.1 The Relationship between Evolution and Genetic Algorithm

EVOLUTION	GENETIC ALGORITHM
An individual is a genotype of the species.	**An individual** is a solution of the optimization problem
Chromosomes defined the structure of an individual.	**Chromosomes** are used to represent the data structure of the solution.
Chromosomes consist of sequence of cells called genes which contain the structural information.	**Chromosomes** consist of sequence of gene species which **placeholder boxes** containing string of data whose unique combination gives the solution value.
The genetic information or traits in each gene is called an **allele.**	**An allele** is an element o data structure stored in a gene place holder.
Fitness of an individual is an interpretation of how the chromosomes have adopted to competition environment.	Fitness of a solution consists in evaluation of measures of the objective function for the solution and comparing it to the evaluations for other solutions.
A population is a collection of species found in a given location.	A population is a set of solution that forms domain search space.
A generation is a given number of individual of the population identified over a period of time.	**A population** is a set solution taken from the population (domain) and generated at an instant of time or in iteration.

Selection is pairing of individual as parent for reproduction.	**Selection** is the operation of selecting parents from the generation to produce offspring.
Crossover is mating and breeding of offspring by chromosomes characteristics are exchanged to form new individuals.	**Crossover** is the operation where by pairs of parents exchange characteristics of their data structure to produce two new individuals as offspring.
Mutation is a random chromosomal process of modification where by the inherited genes of the offspring from their parents are distorted.	**Mutation** is a random operation whereby the allele of a gene in a chromosomes of the offspring is changed by a probability pm.
Recombination is a process of nature's survival of the fittest.	Recombination is the operation whereby elements of the generation and elements of the offspring form an intermediate generation and less fit chromosomes are taken from the generation.

Given a population at t, genetic operators are applied to produce a new population at time ($t+1$). A stepwise evolution of the population from the time (t) to ($t+1$) is called generation. The GA for a single generation is based on the general framework of selection, crossover, Mutation and Recombination.

3.6.2 Representation of individuals

For the purpose of crossover and mutation operations the variables in the genetic algorithm may be represented by an amenable data structure.

Suppose we have the search space $x = 0,1,2,.....10$ then the x values form the individual. The elements of the search space in a binary sequence are encoded by expressing $x = 10$ *and* $x = 0$ in binary sequence to obtain $10 = 1010_2$ and $0 = 0000_2$

Thus $x = 10$ is an individual and 1010 is its chromosome representation. The chromosome has 4 genes placeholder for the alleles. The allele information in the genes will be the binary numbers 0 and 1.the chromosome for $x = 9$ is therefore

Table 3.2

1	0	0	1

There are 2^4 permutations for a binary string of length 4.These 2^4 permutation consist of both infeasible and feasible solutions. There are 11 feasible solutions which constitute the search space and the rest for the infeasible set. Since the solution set is restricted to the integers we look for suboptimal solution. In general the data structure used for the representation of individual depends on variables of the problem at hand.

3.6.3 Fitness function

This the measure associated with the collective objective functions of the optimization problem. The measure indicates the fitness of a particular chromosome representation of a particular individual solution. In the TSP, the fitness function is the sum of the path between the cities.

$$\max\left[f_2 \Big/ \sum_{j=4,i=2}^{5} g_{ij} \,,\; f_2 \Big/ \sum_{j=4,i=3}^{5} g_{ij} \right] = \max\left[4/7, 4/7 \right] = 4/7$$

3.6.4 Mutation

Mutation operation is performed on individual chromosome whereby the alleles are changed probabilistically.

3.6.5 Random swap mutation

In random swap two loci (position) are chosen at random and their values swapped.

3.6.6 Move-and-insert gene mutation

Using move-and-insert, a locus is chosen at random and its value is inserted before or after the value at another at another randomly chosen locus.

3.6.7 Move-and-sequence mutation

Sequence mutation is very similar to the gene move-and-insert but instead of a single locus a sequence loci is moved and inserted before or after the value at another randomly chosen locus.

3.7 Vehicle Routing Problem (VRP)

With regards to a particular number of vehicles, vehicle routing is the problem of determining which customer should be served by which vehicles, and in what order each vehicle should visit its customers. The constraints may include the available fuel, Capacity of each vehicle and available time windows for customers. TSP-based algorithms have been applied in this kind of problem and may also be applied to routing problems in computer networks, (Gerard, 1994).

The figure below shows an example of Vehicle Routing Problem (VRP) with four routes where the triangle in the middle denotes the source node

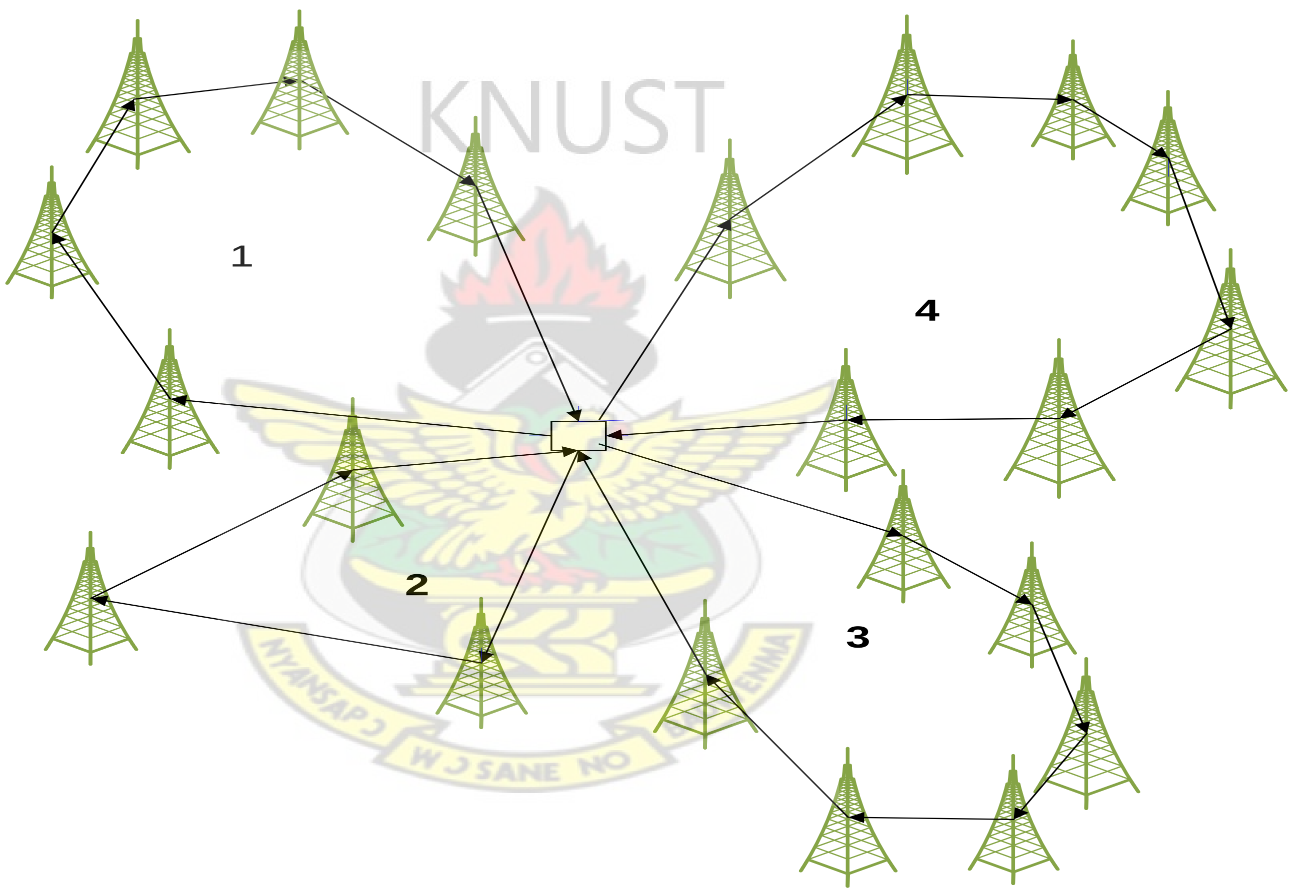

Figure 3.2: A typical solution for a VRP with 4 routes. (The square in the Middle denotes the source node)

3.8.0 Cutting Plane Method

Cutting plane methods are exact algorithms for integer programming problem. They have proven to be very useful computationally in the last few years, especially when combined with a branch and bound algorithm in a branch and cut framework. These methods work by solving a sequence of linear programming relaxations of the integer programming problem.

The relaxations are gradually improved to give better approximations to the integer programming problem, at least in the neighborhood of the optimal solution. For hard instances that cannot be solved to optimality, cutting plane algorithms can produce approximations to the optimal solution in moderate computation times, with guarantees on the distance to optimality.

Cutting plane algorithms have been used to solve many different integer programming problems, including the traveling salesman problem.

J'unger et al, 1995, contains a survey of applications of cutting plane methods, as well as a guide to the successful implementation of a cutting plane algorithm. Nemhauser and Wolse, (1992) provides an execellent and detailed description of cutting plane algorithms as wel as other aspects of integer programming. Research by Schrijver, 1986 and his article in (Schrijver, 1995) are excellent sources of cutting plane applications.

3.8.1 Using the fractional algorithm of cutting plane

In this algorithm all coefficients including the right hand side need to be integer. This condition is necessary as all variables (original, slack and artificial) are supposed to be integers as shown below.

In case a constraint with fractional coefficient exist then both sides of the inequality (equality) are multiplied by the least common multiple of the denominator (LCMD).

For instance $3x_1 + \frac{1}{5}x_2 \le \frac{2}{3}$ becomes $45x_1 + 3x_2 \le 10$

3.8.2 Procedure for cutting plane algorithm

(i) Solve the integer programming problem as a Linear Programming Problem.

(ii) If the optimal solution is integer stop else go to step (iii).

(iii) Introduce secondary constraints (cut) that will push the solution towards integrality (Return to (i).

We show how to the secondary constraints in the following sections.

3.8.3 The construction of the secondary constraints:

Given the integer problem

Minimize $Z = C^T x$

Subject $Ax \le b$

$x \ge 0$, integer x = Vector of decision variable.

C^T = Vector coefficients

A = the given matrix

B = vector coefficient

The optimal tableau of the Linear programming Problem is given in table below:

For simplicity of natation let us have $x = (x_B, x_{NB})$

$x_B = (x_1......x_M)$ *and* $x_{NB} = (x_1.....x_N)$

Table 3.2: The variables to be considered in the Cutting Plane Method.

	x	x_1…. x_i ….	W_1… W_J …. W_N	Solution
	1	x_i	C_1… C_j …. C_N	β_0
x_i	0	1 0	α_{11}… α_{1j} ….	β_1
		0	α_{1N}	
x_i				
	0			β_i
		0 1	α_{i1}… α_{ij} ….	
		0	α_{iN}	
x_i	0			β_M
		0 0	α_{M1}… α_{Mj} ….	
	0	1	α_{MN}	

Consider the *ith* equation where x_i was required to be integer but found not integer.

$$x_i = \beta_i - \sum_{J=1}^{N} (\alpha_{ij}.W_j) \quad \text{and } \beta_i \text{ non-integer} : i = 1,\ldots,M$$

(i) Any real number can be written as the sum of two parts, integer part and the fractional part.

Let $\beta_i = (\beta_i) + f_i$ and $\alpha_{ij} = (\alpha_{ij}) + g_{ij}$ (2)

Then

$$x_i = [\beta_i] + f_i - \sum_{J=1}^{N} ([\alpha_{ij}] + g_{ij}) w_j \quad and$$

$$f_i - \sum_{j=1}^{N} \left(g_{ij} w_j\right) = x_i - [\beta_i] + \sum_{j=1}^{N} \left(a_{ij}\right) w_{ij} \qquad (3)$$

Where $[\alpha] \leq \alpha$ is integer part of α); $0 < f_i > 1$; $0 \leq g_{ij} < 1$
($[\beta] \leq \beta$ and ($[\beta]$ is the integer part of β)
(note that $f_i > 0$ as x_i is presently not integer)
Since all $x_i (i = 1,\ldots, M)$ and all W_j $(j = 1,\ldots, N)$ must be integer,
the right hand-side is consequently integer and therefore the
left-hand side is also integer thus from the table above

$$f_i - \sum_{i=1}^{N} (g_{ij} w_j) \in \qquad \text{(Integer)} \qquad (4)$$

$g_{ij} \geq 0$ and $w_j \geq 0$ then from equation (3) with $X_i > (\beta_i)$ $f_i - \sum_{i=1}^{N} (g_{ij} w_j) \geq 0$

Therefore

$$f_i \geq f_i - \sum_{i=1}^{N} (g_{ij} w_j) \quad \textit{for all } i = 1,\ldots.N \qquad (5)$$

since 0 < f_i < 1 we have $f_i - \sum_{i=1}^{N}(g_{ij} w_j)$ < 1 and using (4) we obtain

$$f_i - \sum_{i=1}^{N}(g_{ij} w_j) \le 0 \tag{6}$$

Constraint (6) is the cut and can be expressed as a secondary constraints by adding slack variable:

This gives

$$f_i - \sum_{i=1}^{N}(g_{ij} w_j) + s_i = 0 \rightarrow s_i = \sum_{i=1}^{N}(g_{ij} w_j) - f_i \text{ for all } i = 1, \ldots M$$

where $s_i \ge 0$ (integer slack variable) (7)

3.8.4 Choice of the cut

Suppose two rows in table 2.0 gives non-integer solutions in x_i and x_k then there will be two cuts based on x_i and x_k having the following conditions:

(*i*) $f_k \le \sum_{i=1}^{N}(g_{ij} w_j)$

(ii) $f_k \le \sum_{i=1}^{N}(g_{kj} w_j)$

Cut (i) is stronger than cut (k) if

(iii) $f_i \ge f_k$ and g_{ij} for and j

With the strict inequality happening at least once.

In other words a cut is deeper in x_i direction as f_i increases and g_{ij} decreases

The Condition (iii) is difficult to implement computationally and therefore empirical rule that take into account the above definition have been developed.

Criterion (b) is more efficient as this represents the definition given by (iii) better.

$$(a)\quad fr \Big/ \sum_{j=1}^{N} g_{rk} = \max\left\{ fi \Big/ \sum_{j=1}^{N} grk \;; i = 1, ...M \,;\, x_i \, for \text{ a Specified k} \right\}$$

$$(b)\quad fr \Big/ \sum_{j=1}^{N} g_{rj} = \max\left\{ fi \Big/ \sum_{j=1}^{N} grj \;; i = 1, ...M \,;\, xi \notin but\ xi \text{ required to be integer} \right\}$$

$$(c)\quad fr \Big/ \sum_{j=1}^{N} g_{ik} = \max\left\{ fi \Big/ \sum_{j=1}^{N} gik \;; i = 1, ...M \,;\, x_i \, for \text{ a Specified k} \right\}$$

Phototype Example

Maximize $z = 7x_1 + 9x_2$

subject to $-x_1 + 3x_2 \le 6$

$7x_1 + x_2 \le 35$

$x_1 \ge 0, x_2 \ge 0$,integer

solution

Maximise $z = 7x_1 + 9x_2 + 0s_1 + 0s_2$

subject to

$-x_1 + 3x_2 + 1s_1 = 6$

$7x_1 + x_2 + 1s_2 = 35$

Table 3.3 Final tableau for first iteration

	c_j	7	9	0	0	
c_b	Basic variable	x_1	x_2	s_1	s_2	Solution
9	x_2	0	1	$-7/22$	$-1/22$	$-7/2$
7	x_1	1	0	$-1/22$	$1/22$	$9/2$
	z_j	7	9	0	0	63
	$c_j - z_j$	0	0	$-28/11$	$-15/11$	

Let $s_1 = x_3, s_2 = x_4, z = x_5$

from the tableau the optimal solution becomes $z = 63$,where $x_2 = 7/2$, and $x_1 = 9/2$

since x_2 and x_1 are not integers,we apply the concepts of cutting plane techniques

$x_2 + 7/22\, x_3 + 1/22\, x_4 = 2/7$ (1)

$x_1 + 0x_2 - 7/22\, x_3 + 1/22\, x_4 = 9/2$ (2)

Choice of Cut

Taking the equations(1) and (2)

$x_2 + \left(0 + 7/22\right)x_3 + \left(0 + 1/2\right)x_4 = \left(3 + 1/2\right)$ (1)a

$x_1 - \left(1 - 21/22\right)x_3 + \left(0 + 3/22\right)x_4 = \left(4 + 1/2\right)$ (2)b

$1/2 - 7/22\, x_3 - 1/2\, x_4 = x_2 + 0x_3 + 0x_4$ (3) integer (1a)

$1/2 - 21/22\, x_3 - 3/22\, x_4 = x_1 - x_3 + 0x_4$ (4) integer (2b)

$f_2 = 1/2, g_{23} = 7/22, g_{24} = 1/2$

$f_3 = 1/2, g_{23} = 21/22, g_{34} = 3/22$

using;

$$f_r \Big/ \sum_{j=1}^{N} g_{rj} = \max\left\{ f_i \Big/ \sum_{j=1}^{N} g_{rj} \;; i = 1, \ldots .M ; x_i \notin but\ x_i \text{ required to be integer} \right\}$$

where $t = 2,\ j = 3, 4$

$$f_2 = 1\!/\!2 , g_{23} = 7\!/\!22 , g_{24} = 1\!/\!22$$

$$\sum_{j=3}^{4} g_{ij} = 7\!/\!22 + 1\!/\!22 = 8\!/\!22$$

$when\ i = 3,\ j = 3, 4$

$$g_{33} = 21\!/\!22 , g_{34} = 3\!/\!22 .$$

$$\sum_{j=3}^{4} g_{ij} = 7\!/\!22 + 3\!/\!22 = 24\!/\!22$$

$$\max\left[\left(\frac{1/2}{8/22} \right), \left(\frac{1/2}{24/22} \right) \right]$$

$\max\left[\frac{22}{16},\frac{22}{48}\right]=\frac{22}{16}$

hence(la) would be considered to be part of the new constrains.

Thus $\frac{1}{2}-\frac{7}{22}x_3 - \frac{1}{22}x_4 \geq 0$

and $\frac{1}{2}-\frac{7}{22}x_3-\frac{1}{22}x_4+s_3=0$

$-\frac{7}{22}x_3-\frac{1}{22}x_4+s_3=-\frac{1}{2}$

The system of equations becomes;

$z=7x_1+9x_2+0x_3+0x_4+0x_5$

subject to;

$x_2+\frac{7}{22}x_3+\frac{1}{22}x_4=\frac{7}{2}$

$x_1+0x_2-\frac{1}{22}x_3+\frac{3}{22}x_4=\frac{9}{2}$

$-\frac{7}{22}x_3+\frac{1}{22}x_4+x_5=-\frac{1}{2}$

$s_3=x_5$

Table 3.4 Final tableau for second iteration

	C_j	7	9	0	0	0	
c_b	Basic variable	x_1	x_2	x_3	x_4	s_3	Solution
9	x_2	0	1	0	0	1	3
7		1	0	0			
0	x_3	0	0	1	$\frac{1}{7}$	$-\frac{22}{7}$	$-\frac{11}{7}$
	z_j	7	9	0	1	0	59
	c_j-z_j	0	0	0	-1	-8	

$z_{\max} = 59, x_2 = 3, x_1 = 32/7, and\ x_3 = 11/7$

since x_1 and x_3 are not integers we apply the cutting plane techniques.

using the fractional algorithm;

$x_1 + 1/7\, x_4 - 1/7\, x_5 = 32/7$ (1)*

$\rightarrow x_1 + \left(0 + 1/7\right) x_4 + \left(-1 + 6/7\right) x_5 = 4 + 4/7$

$x_1 + 0x_4 - 1x_5 - 4 = 4/7 - 1/7\, x_4 + 6/7\, x_5$ integer (1a)*

$x_3 + 1/7\, x_4 - 22/7\, x_5 = 11/7$ (2)*

$\rightarrow x_3 + \left(0 + 1/7\right) x_4 + \left(-4 + 6/7\right) x_5 = 1 + 4/7$

$x_3 + 0x_4 - 4x_5 - 1 = 4/7 - \left(1/7\, x_4 + 6/7\, x_5\right)$ integer (2a)*

Choice of Cut

From (1a)* $f_2 = 4/7, g_{24} = 1/7, g_{25} = 6/7$

From (2a)* $f_3 = 4/7, g_{34} = 1/7, g_{35} = 6/7$

using;

$$f_r \Big/ \sum_{j=1}^{N} g_{rj} = \max\left\{ f_i \Big/ \sum_{j=1}^{N} g_{rj} \ ; i = 1,....M ; x_i \notin but\ x_i \text{ required to be integer} \right\}$$

where $t = 2,\ j = 3,4$

$f2 = \ 4/7,\ g_{24}, = 1/7, g_{25} = 6/7$

Therefore

$$\sum_{j-4}^{5} g_{ij} = 1/7 + 6/7 = 1$$

when $i = 3,\ j = 4,5$

$f_3 = \ 4/7,\ g_{34}, = 1/7, g_{35} = 6/7$

$$\sum_{j=4}^{5} g_{ij} = 1/7 + 6/7 = 1$$

$$\max\left[f_2 \Big/ \sum_{j=4,i=2}^{5} g_{ij} \ , \ f_2 \Big/ \sum_{j=4,i=3}^{5} g_{ij} \right] = \max\left[4/7, 4/7\right] = 4/7$$

Tie will be broken arbitrary by choosing equation (2)*as the new constrains to be added.

where $s_3 = x_5$

The system of equations becomes;

$Z = 7x_1 + 9x_2 + 0x_3 + 0x_4 + 0x_5 + 0s_4$

subject to

$x_2 = 3$

$x_1 + 1/7\, x_4 - 1/7\, x_5 = 32/7$

$x_1 + 1/7\, x_4 - 1/7\, x_5 = 11/7$

$-1/7\, x_4 - 6/7\, x_5 + s_4 = -4/7$

Table 3.5 Final tableau for last iteration

	C_j	7	9	0	0	0	0	
c_b	Basic variable	x_1	x_2	x_3	x_4	x_5	s_5	Solution
9	x_2	0	1	0	0	0	0	3
7	x_1	1	0	0	0	-1	1	4
0	x_3	0	0	1	0	-4	1	1
0	x_4	0	0	0	1	6	-7	4
	z_j	7	9	0	0	-7	7	55
	$-c_j - z_j$	0	0	0	0	7	-7	

Now the $z_{\max} = 55,\ x_2 = 3,\quad x_1 = 4,\ x_3 = 1,\quad and\quad x_4 = 4$
since the variables are integers, we stop here

CHAPTER 4

4.0 DATA ANALYSIS AND RESULTS

For the purpose of this work, the twenty four (24) Base transceiver Station site are numerically represented within the New Juabeng Municipality, within which the arbitrary numbers have been allocated to each BTS. Number Allocated**,** Distance Matrix for New Juabeng Municipality, are as shown in Table 4.2

Table 4.1: BTS location and number allocated in the New Juabeng municipality

NUMBER ALLOCATED	BTS LOCATION	NUMBER ALLOCATED	
		Longitude	**Latitude**
1	Koforidua radio station	-0.243330	6.088360
2	Nyerede Adawso	-0.256840	6.063335
3	Adukrom	-0.075240	6.020460
4	Somanya	-0.014630	6.104280
5	Huhunya_Ex	-0.167640	6.174700
6	Krobo Odumasi	-0.000740	6.134670
7	Akuse_Ex	-0.114660	6.093550
8	Akosombo_2	-0.054870	6.288470
9	Asesewa	-0.147490	6.401820
10	Begoro	-0.378790	6.376920
11	Kukurantumi	-0.370831	6.185835
12	New Tafo	-0.372489	6.232176
13	Apedwa	-0.488680	6.112400
14	Kibi	-0.554130	6.170800
15	Akim-Asafo	-0.473681	6.178264
16	Asiakwa	-0.496510	6.266440
17	Osino	-0.480660	6.341860
18	Akrade	-0.077105	6.203555
19	Koforidua_Ex_1	-0.259600	6.091700
20	Koforidua_Ex_2	-0.254270	6.096790
21	Koforidua_North	-0.256996	6.078823
22	Koforidua_High-Court	-0.027965	6.129740
23	Akosombo radio station	-0.040190	6.244540
24	Juapong	-0.134800	6.250090

The table below shows the distance matrix obtained from distances between the Twenty four (24) Base Transceiver station within the New Juabeng Municipality. For BTS which have no direct link, the minimum distance along the edges is considered.

C_{ij} = The distance from BTS i to BTS j

$C_{ii} = C_{jj} = 0$ = There is no distance.

Table 4.2 Distance Matrix for 24 Base transceiver Stations (BTS) in the New Juabeng municipality in Kilometers.

C_{ij}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	0	5	30	45	9	50	74	80	39	54	28	38	35	56	35	49	46	58	3	2	6	4	78	79
2	5	0	35	50	14	55	79	85	44	59	33	43	40	61	40	54	51	63	2	3	7	1	83	84
3	30	35	0	16	32	21	30	47	72	84	50	57	64	79	74	89	78	28	27	28	33	26	45	46
4	45	50	16	0	42	5	14	31	56	88	62	72	79	94	85	104	98	13	42	43	48	44	29	30
5	9	14	32	42	0	34	53	76	33	64	31	42	49	64	59	79	60	42	7	8	12	9	74	50
6	50	55	21	5	34	0	19	25	51	83	67	77	74	89	80	99	93	8	47	48	53	49	23	85
7	74	79	30	14	53	19	0	44	86	102	64	86	93	108	99	118	112	14	71	72	77	73	92	76
8	80	85	47	31	76	25	44	0	76	108	92	102	99	114	105	124	118	17	77	78	83	79	2	7
9	39	44	72	56	33	51	86	76	0	93	67	74	74	95	74	88	85	59	36	37	42	38	74	75
10	54	59	84	88	64	83	102	108	93	0	35	26	46	42	37	34	33	91	51	52	57	53	105	107
11	28	33	50	62	31	67	64	92	67	35	0	10	43	39	34	31	30	75	25	26	31	27	90	91
12	38	43	57	72	42	77	86	102	74	26	10	0	33	29	24	21	20	85	35	36	42	37	100	100
13	35	40	64	79	49	74	93	99	74	46	43	33	0	14	11	22	31	82	32	33	38	34	97	98
14	56	61	79	94	64	89	108	114	95	42	39	29	14	0	25	15	26	97	53	54	59	55	102	113
15	35	40	74	85	59	80	99	105	74	37	34	24	11	25	0	21	20	88	32	33	38	35	103	104
16	49	54	89	104	79	99	118	124	88	34	31	21	22	15	21	0	19	107	46	47	52	48	120	123
17	46	51	78	98	60	93	112	118	85	33	30	20	31	26	20	19	0	101	43	44	49	45	115	117
18	58	63	28	13	42	8	14	17	59	91	75	85	82	97	88	107	101	0	55	56	61	57	15	16
19	3	2	27	42	7	47	71	77	36	51	25	35	32	53	32	46	43	55	0	1	3	1	75	81
20	2	3	28	43	8	48	72	78	37	52	26	36	33	54	33	47	44	56	1	0	4	2	76	79
21	6	7	33	48	12	53	77	83	42	57	31	42	38	59	38	52	49	61	3	4	0	5	81	86
22	4	1	26	44	9	49	73	79	38	53	27	37	34	55	35	48	45	57	1	2	5	0	77	84
23	78	83	45	29	74	23	92	2	74	105	90	100	97	102	103	120	115	15	75	76	81	77	0	6
24	79	84	46	30	50	85	76	7	75	107	91	100	98	113	104	123	117	16	81	79	86	84	6	0

4.1 Formulation of the TSP model

The problem can be defined as follows: Let $G = (V,E)$ be a complete undirected graph with Vertices V, $|V|=n$, where n is the number of BTS, and edges E with edge length d_{ij} for (i,j). We focus on the symmetric TSP case in which $C_{ij} = C_{ji}$, *for* all (i,j).

We formulate this minimization problem as an integer programming, as shown in Equations (1) to (5).

Subject to

$$\text{P1: } \min \sum_{i\in v} \sum_{j\in v} C_{ij} x_{ij} \quad (1)$$

$$\sum_{\substack{j\in v \\ j\neq i}} x_{ij} = 1 \quad i \in v \quad (2)$$

$$\sum_{\substack{j\in v \\ j\neq i}} x_{ij} = 1 \quad j \in v \quad (3)$$

$$\sum_{i\in s,\, j\in s} x_{ij} \leq |s| - 1 \quad \forall s \subset v, \neq \emptyset \;\; x_{ij} = 0 \text{ or } 1 \;\; i, j \in v \quad (4)$$

$$x_{ij} = 0 \text{ or } 1 \quad i, j \in v \quad (5)$$

The problem is an assignment problem with additional restrictions that guarantee the exclusion of sub tours in the optimal solution. Recall that a sub tour in V is a cycle that does not include all vertices (or BTS). Equation (1) is the objective function, which minimizes the total distance to be traveled.

Constraints (2) and (3) define a regular assignment problem, where (2) ensures that each BTS is entered from only one other BTS, while (3) ensures that each BTS is only departed to on other BTS. Constraint (4) eliminates sub tours. Constraint (5) is a binary constraint, where $x_{ij} = 1$ if edge (i,j) in the solution and = 0, otherwise.

4.2 Data Analysis

To satisfy the constrains (2) and (3) we choose the random

initial tour (x^o) =18 – 2 – 5 – 6 – 17 – 7 – 10 – 9 – 11 – 12 – 15 – 4 – 3 – 13 – 8 – 16 – 14 - 19 -20 – 21 – 22 – 23 - 24

From the objective function (1) the initial distance = d(x^o) =

10-4-3-6-8-15-21-18-24-20-7-9-17-14-11-5-2-1-13-12-16-19-22-23

D(10,4)+d(4,3)+d(3,6)+d(6,8)+d(8,15)+d(15,21)+d(21,18)+d(18,24)+d(24,20)+d(20,7)+ d(7,9)
+d(9,17)+d(17,14)+d(14,11)+d(11,15)+d(5,2)+d(2,1)+d(1,13)+d(13,16)+d(16,19)+d(19,22)+d (22, 23) = 988km

The initial temperature is taken to be $(T_o) = 4069.00, \alpha = 0.99$.
Temperature is updated by using the formula;
$T_{(k+1)} = \alpha T_k$ where k is the number of iterations.
Stop when $T \leq 42.00$

4.3 Computational Procedure

Matlab 2009 software version was installed on Dell Latitude| E6400 Laptop Computer with Intel® Core ™ 2 Duo CPU P8700@ 2.53GHZ, 783MHZ, 1.95GHZ of RAM. Simulated annealing algorithm was used to obtain the final solution. After 1500 iterations in 370.25 seconds . The execution time was varied with number of iterations

4.4 Results

After performing 1600 iterations the optimal tour = 1- 20 – 19 – 22 – 2 – 21 – 5 – 11 - 12 – 17 – 16 – 14 – 13 – 15 – 10 – 3 – 4 - 6 – 18 – 23 – 8 – 24 – 7 – 9

Thus

d(1,20)+d(20,19)+d(19,22)+d(22,2)+d(2,21)+d(21,5)+d(5,11)+d(11,12)+d(12,17)+d(17,16)+d(16,14)+d(14,13)+d(13,15)+d(15,10)+d(10,3)+d(3,4)+d(4,6)+d(6,18)+d(18,23)+d(23,8)+d(8,24)+d(24,7)+d(7,9) = 453km

The optimal tour was found to be the same after it was run nine times.

The optimal tour is as follows:

$$
\begin{aligned}
&\text{Koforidua\_RS} \rightarrow \text{Koforidua\_2} \rightarrow \text{Koforidua\_1} \rightarrow \text{Koforidua\_High} - \text{Court} \rightarrow \\
&\rightarrow \text{Nyerede Adewso} \rightarrow \text{Koforidua\_North} \rightarrow \text{Huhunya} \rightarrow \text{Kukurantumi} \rightarrow \\
&\rightarrow \text{New Tafo} \rightarrow \text{Osino} \rightarrow \text{Asiakwa} \rightarrow \text{Kibi} \rightarrow \text{Apedwa} \rightarrow \text{Akim} - \text{Asafo} \rightarrow \\
&\rightarrow \text{Begoro} \rightarrow \text{Adukrom} \rightarrow \text{Somanya\_Ex} \rightarrow \text{Krobo Odumasi} \rightarrow \text{Akrade} \rightarrow \\
&\rightarrow \text{Akosombo\_RS} \rightarrow \text{Akosombo\_2} \rightarrow \text{Juapong} \rightarrow \text{Akuse} \rightarrow \text{Asesewa}
\end{aligned}
$$

CHAPTER 5

OBSERVATION, CONCLUSION AND RECOMMENDATION

5.0 Conclusion

The simulated annealing algorithm can be a useful tool to apply to hard combinatorial problems like that of TSP. Using simulated annealing as a method in solving the symmetric TSP model has been proved that it is possible to converge to the best solution.

It could therefore be concluded that the objective of finding the minimum tour from the symmetric TSP model by the use of simulated annealing algorithm was successfully achieved. The study shows clearly that, any visit to the Base transceiver station site within the Vodafone Network New Juabeng Municipality must undertake the visit in the order below to minimize cost of Traveling and mean Time to restored (MTTR) in order to reduced the Operation cost. Also, recommended that similar exercise should be replicated in the other region. The order is as follows:

$$\begin{aligned}
&\text{Koforidua_RS} \rightarrow \text{Koforidua_2} \rightarrow \text{Koforidua_1} \rightarrow \text{Koforidua_High} - \text{Court} \rightarrow \\
&\rightarrow \text{Nyerede Adewso} \rightarrow \text{Koforidua_North} \rightarrow \text{Huhunya} \rightarrow \text{Kukurantumi} \rightarrow \\
&\rightarrow \text{New Tafo} \rightarrow \text{Osino} \rightarrow \text{Asiakwa} \rightarrow \text{Kibi} \rightarrow \text{Apedwa} \rightarrow \text{Akim} - \text{Asafo} \rightarrow \\
&\rightarrow \text{Begoro} \rightarrow \text{Adukrom} \rightarrow \text{Somanya_Ex} \rightarrow \text{Krobo Odumasi} \rightarrow \text{Akrade} \rightarrow \\
&\rightarrow \text{Akosombo_RS} \rightarrow \text{Akosombo_2} \rightarrow \text{Juapong} \rightarrow \text{Akuse} \rightarrow \text{Asesewa}
\end{aligned}$$

5.1 Recommendations

After a comprehensive study of TSP and Simulated annealing algorithm, the following

a. Recommendation should be considered. Vodafone engineer or contractor who visit the Base Transceiver Station site within the New Juabeng Municipality for fault restoration, preventive maintenance, installation, integration of site, co-location of site, pre-acceptance test(PAT) etc, should consider the routes below in other to minimize their cost.

$$\begin{aligned}
&\text{Koforidua_RS} \rightarrow \text{Koforidua_2} \rightarrow \text{Koforidua_1} \rightarrow \text{Koforidua_High}-\text{Court} \rightarrow \\
&\rightarrow \text{Nyerede Adewso} \rightarrow \text{Koforidua_North} \rightarrow \text{Huhunya} \rightarrow \text{Kukurantumi} \rightarrow \\
&\rightarrow \text{New Tafo} \rightarrow \text{Osino} \rightarrow \text{Asiakwa} \rightarrow \text{Kibi} \rightarrow \text{Apedwa} \rightarrow \text{Akim}-\text{Asafo} \rightarrow \\
&\rightarrow \text{Begoro} \rightarrow \text{Adukrom} \rightarrow \text{Somanya_Ex} \rightarrow \text{Krobo Odumasi} \rightarrow \text{Akrade} \rightarrow \\
&\rightarrow \text{Akosombo_RS} \rightarrow \text{Akosombo_2} \rightarrow \text{Juapong} \rightarrow \text{Akuse} \rightarrow \text{Asesewa}
\end{aligned}$$

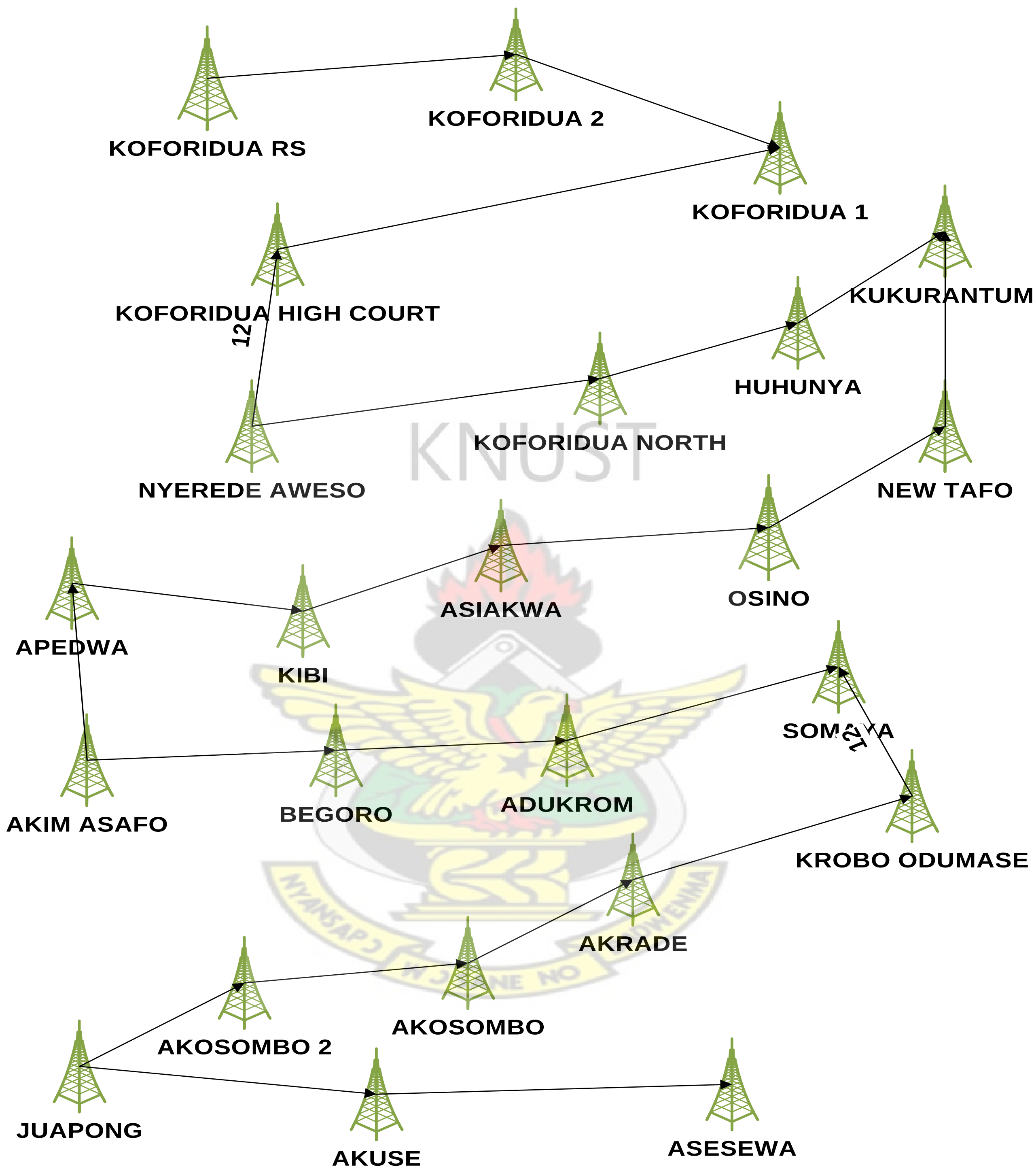

Figure 5.1 Interconnection of BTS within the New Juabeng municipality

Students can use this work for further research covering the other regions in Ghana.

REFERENCES

1. Ackoff R.L Arnoff, E.L Sengupta, S.S. (1961). Mathematical Programming. In progress in Operations Research. R.L. Ackoff, editor. John Wiley and Sons: New York: NY. 105-210.

2. Akorli K. F and Allotey F. K. A. (1999). Telecommunications in Ghana, Journal of communications and Information, Paper VII. (www.vii.org/papers/ghana.htm)

3. Al-Hboub-Mohamad H. and Selim Shokrik, Z. (1993). A Sequencing Problem in the weaving Industry. European Journal of Operational Research (The Netherlands). 66(1):6571.

4. Applegate D., Bixby, R., Chv'atal, V, and Cook, W. (1999). .Finding Tours in the TSP.Technical Report 99885. Research Institute for Discrete Mathematics, Universitaet Bonn: bonn, Germany.

5. AppleGate D, Bixby R., Chvatal V and Cook W. (1998). On the Solution of the Traveling Salesman Problems. Documenta Mathematica – Extra Volume ICM, chapter 3, pp. 645-656

6. Applegate, D., Bixby, R., Chv'atal, V. and Cook, W. (2007): The Traveling Salesman problem. Princeton University Press: Princeton, NJ.

7. Applegate, D., Bixby, R., Chv´atal, V., and Cook, W. (1994): Finding Cuts in the TSP (A preliminary report), Tech. rep., Mathematics, AT&T Bell Laboratories, Murray Hill,

8. Amponsah, S .K and F.K Darkwah (2007): Lecture notes on operation Research, IDL KNUST 62-67

9. Bellmore, M. and Nemhauser, G.L. (1968). The Traveling Salesman Problem: A Survey. operations Research. 16:538-558.

10 Bellmore .M and. Nemhauser G. L, (1968) The Traveling Salesman Problem: A Survey operations Research, Vol. 16, No. 3 pp. 538-558. http://www.jstor.org/stable/168581

11. Bock, F. (1958). An Algorithm for Solving Traveling-Salesman' and Related Network optimization Problems . Research Report, Operations Research Society of America Fourteenth National Meeting: St. Louis, MO. Problems. Research Report, Operations Research Society of America Fourteenth National Meeting: St. Louis, MO.

12. Burkard, R.E. (1979). Traveling Salesman and Assignment Problems: A Survey. In: discrete Optimization 1. P.L. Hammer, E.L. Johnson, and B.H. Korte, eds. Annals of Discrete mathematics Vol. 4, North-Holland: Amsterdam. 193-215.

13. Carpaneto, G., Dell'Amico, M. and Toth, P., (1995), .Exact Solution of Large-scale asymmetric Traveling Salesman Problems. ACM Transactions on Mathematical Software, 21, pp.394–409.

14.Cerny, V. (1985) Thermodynamical Approach to the Traveling Salesman Problem: An efficient Simulation Algorithm, J. Opt. Theory Appl., 45, 1, 41-51.

15. Crama,Y. ,Van de Klundert, J., and Spieksma, F. C.R. (2002). .Production Planning problems in Printed Circuit Board Assembly.. Discrete Applied Mathematics. 123:339-361.

16. Croes, G.A. 1958. A Method for Solving Travelling-Salesman Problems. Operations research. 6:791-812.

17. Dacey, M.F. 1960. Selection of an Initial Solution for the Traveling-Salesman Problem. operations Research. 8:133-134. Darwin.C, (1859).On the origin of spacies,1st edition (facsimile-1964), Harvard University Press,MA

18. Datta, S,(2000).Application of operational Research to the Transportation Problems in Developing Countries:A review, Global Business Review, Volume 18,No.1 pages 113-132

19. Dantzig, G.B., Fulkerson, D.R., and Johnson, S.M. (1954). Solution of a Large-Scale Traveling-Salesman Problem. Operations Research. 2: 393-410.

.

20. Eastman, W.L. (1958). Linear Programming with Pattern Constraints. Ph.D. dissertation. Harvard University: Cambridge, MA.

21. Ferreir, J.V. (1995). .A Travelling Salesman Model for the Sequencing of Duties in Bus crew Rotas. Journal of Operational Research Society (UK). 46(4): 415-426.

22. Fischetti, M., Lodi, A., Toth, P., (2002). Exact Methods for the Asymmetric Traveling Salesman Problem. In: Gutin, G., Punnen, A.P. (Eds.), the Traveling Salesman Problem and its Variations. Kluwer, Dordrecht, pp. 169–194 (Chapter 9).

23. Foulds, L.R. and Hamacher, H.W. (1993). .Optimal Bin Location and Sequencing in printed Circuit Board Assembly.. European Journal of Operational Research (Netherlands). 66(3):279- 290.

24. Garey M.R. and Johnson D.S. (1979).Computers and Intractability: A Guide to the theory of NP Completeness, W.H .Freeman, San Francisco. Hitchcock F.L.,The Distribution of Product from Several Sources to Numerous Localities, J. Math. Phys., 20, No. 2, 1941, 217–224.

25 Genetic Algorithms. In Ibrahim H. Osman and James P. Kelly, editors, Proceedings of the meta-heuristics Conference, pages 53–62, Norwell, USA,. Kluwer Academic Publishers

26. Gerard Reinelt, (1994). The Traveling Salesman: Computational Solutions for TSP applications.

27. Goldberg, D.E, (1989). Genetic Algorithm for Search, Optimization and Machine learning, Addison-Wesley.

28. Golden B.L., Wasil E.A., Kelly J.P, and. Chao I-M, (1998). Metaheuristics in Vehicle routing. In Fleet Management and Logictics, T.G. Crainic and G. Laporte (eds), Kluwer, Boston, 33-56.

29. GoldenB.L.and Assad A.A.,(1988). Vehicle Routing: Method sand Studies, Elsevier Science, Amsterdam

30. Gomory, R.E. (1960). Solving linear programming problems in integers. In Bellman, R., Hall Jr., M. (eds) Combinatorial Analysis: Proceedings of Symposia in Applied Mathematics X. American Mathematical Society, Providence, Rhode Island, pp. 211–215.

31. Gonzales, R.H. (1962). Solution to the Traveling Salesman Problem by Dynamic programming on the Hypercube. Technical Report Number 18, Operations Research Center, Massachusetts Institute of Technology: Boston, MA.

32.Gr¨otschel, M., J¨unger, M., and Reinelt, G. (1984). A cutting plane algorithm for the linear ordering problem in Operations Research 32, 1195

33.Gr¨otschel, M., Martin, A., and Weismantel, R. (1996): Packing Steiner trees: a cutting plane algorithm and computational results‘, Mathematical Programming 72 ,125–145.

34.Grötschel, M. (1980). On the Symmetric Travelling Salesman Problem: Solution of a 120-City Problem. Mathematical Programming Study. 12: 61-77.

35. Grötschel M and Padberg .M, (1993). Ulysses 2000: In Search of Optimal Solutions to hard Combinatorial Problems, Technical Report, New York University Stern School of business

36. G¨unther, H.O., Gronalt, M., and Zeller, R. (1998). .Job Sequencing and Component Set-up on a Surface Mount Placement Machine. Production Planning & Control. 9(2):201–211.

37. Gutin, G. and Punnen, J. (2002). The TSP and its Variations. Kluwer Academic Publishers.

38 .Haggarty, L.; Shirley M. M. and Wallsten, S. (2002). Telecommunication Reform in Ghana World Bank Policy Research Working Paper No. 2983, World Bank, Washington DC.

39. Hatfield, D. J. AND J. F. Pierce, (1966).Production sequencing by Combinatorial Programming, IBM Cambridge Scientific Center, Cambridge, Mass.

40. Helbig, H.K. and Krarup, J. (1974). Improvements of the Held-Karp Algorithm for the Symmetric Traveling-Salesman Problem. Mathematical Programming. 7:87-96.

41. Held, M. and Karp, R.M. (1970). The Traveling-Salesman Problem and Minimum Spanning Trees. Operations Research. 18:1138-1162.

42. Held, M. and Karp, R.M. (1971). The Traveling-Salesman Problem and Minimum Spanning Trees: Part II. Mathematical Programming. 1:6-25.

43. Heller, I. (1955). On the Travelling Salesman's Problem. Proceedings of the Second Symposium in Linear Programming: Washington, D.C. Vol. 1.

44. Hillier F.S and Lieberman G.J (2005): Introduction to operations Research. McGraw-hill companies inc,1221 Avenue of the Americas, New York ,NY 10020.

45. Hong, S. (1972). A Linear Programming Approach for the Traveling Salesman Problem. Ph.D. Thesis. The Johns Hopkins University: Baltimore, MD.

46. Johnson D.S and Mcgeoch L.A (2002): Experimental Analysis of Heuristics for STSP, In The Traveling Salesman Problem and its Variations (G. Gutin and A. P. Punnen, eds.), Kluwer.

47. Johnson, D.S., Gutin, G. McGeoch, L.A., Yeo, A., Zhang, W., and Zverovitch, A. (2002). Experimental Analysis of Heuristics for the Asymmetric Traveling Salesman

48. Kahng, A.B. and Reda, S. (2004). 'Match Twice and Stitch: A new TSP Tour Construct heuristic.. Operations Research Letters. 32(6): 499-ion 509.

49. Karg, R.L. and. Thompson, G.L. (1964). A Heuristic Approach to Solving Travelling Salesman Problems. Management Science. 10:225-248.

50. Karp, R.M., Steele, J.M., (1990). Probabilistic Analysis of Heuristics. In The Traveling Salesman Problem, Wiley, New York, pp. 181–205 .

51. Keuthen, R. (2003). .Heuristic Approaches for Routing Optimization.. PhD thesis at the University of Nottingham: UK.

52. Kirkpatrick, S., Gelatt, C.D. Jr., and Vecchi, M.P. (1983). Optimizations by simulated annealing. Science, 220, 671-681.

53. Kirkpatrick, S., C. D. Gelatt Jr., M. P. Vecchi, (1983). Optimization by Simulated annealing, Science, 220, 4598, 671-680.

54. Kolohan, F. and Liang, M. (2000). .Optimization of Hole Making: A Tabusearch approach.. International Journal of Machine Tools & Manufacture. 50:1735-1753.

55. Kruskal, J.B. (1956). On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. Proceedings of the American Mathematical Society. 2:48-50.

56. Kwon, S., Kim, H., and Kang, M. (2005). Determination of the Candidate Arc Set for the asymmetric Traveling Salesman Problem. Computers and Operations Research. 32(5): 1045-1057.

57. Lambert, F. (1960): The Traveling-Salesman Problem. Cahiers du Centre de Recherche opérationelle. 2:180-191.

58. Lawler, E.L. and Wood, D.E. (1966): Branch-and-Bound Methods: A Survey. Operations Research. 14:699-719.

59. Lawler, E.J., Lenstra, J.K., Rinnoy Kan, A.H.G., and Shmoys, D.B. (1985). .The Travelling Salesman Problem: A Guided Tour of Combinatorial Optimization.. John Wiley & Sons: New York, NY.

60. Lawler and. Wood D. E, (1966) Branch-and-Bound Methods: A Survey, Opns. Res. 14, 69-719

61. Lin, S. and Kernighan, B.W. (1973). An Effective Heuristic Algorithm for the Traveling-Salesman Problem. Operations Research. 21:498-516.

62. Little, J. D. C., K. G.Murty , D. W. Sweeney, AND C. Karel , (1963). An Algorithm for the Traveling Salesman Problem, Opns. Res. 11, 979-989 .

63. Little, J.D.C., Murty, K.G., Sweeney, D.W., and Karel, C. (1963). An Algorithm for the Traveling Salesman Problem. Operations Research. 11: 972-989.

64. M. L. Fisher (1981). The Lagrangian relaxation method for solving integer programming problems. Management Science, 27:1–18

65. Miller, D. and Pekny, J. (1991),Exact Solution of Large Asymmetric Traveling Salesman Problems, Science, 251:754–761

66. Mitchell, J. E., and Borchers, B. (1997): Solving linear ordering problems with a Combined interior point/simplex cutting plane algorithm, Tech. rep. Mathematical Sciences,

67. Mitchell, J. E., and Borchers, B. (1996): Solving real world linear ordering problems using a primal-dual interior point cutting plane method, Annals of Operations Research 62, 253–276.

68. Mitrovic-Minic, S. and Krishnamurti, R. (2006). The Multiple TSP with Time Windows: Vehicle Bounds Based on Precedence Graphs. Operations Research Letters. 34(1): 111-120.

69. Morton, G. and Land, A.H. (1955). A Contribution to the Travelling-Salesman' Problem. Journal of the Royal Statistical Society, Series B. 17:185-194.

70. Nemhauser, G. L., and Sigismondi, G. (1992): A strong cutting plane/branch-and-bound algorithm for node packing, Journal of the Operational Research Society 43 443–457.

71. Padberg, M.W. and Rinaldi, G. (1987). Optimization of a 532-city Symmetric Traveling Salesman Problem by Branch and Cut. Operations Research Letters. 6:17.

72. Padberg M. and Rinaldi G. (1991) A branch-and-cut algorithm for the resolution of large-scale traveling salesman problem, SIAM Review,33, 60–100.

73. Padberg, M., and Rinaldi, G. (1991): A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems, SIAM Review 33(1)), 60–100.

74. Papadimitriou C.H. and Stieglitz K.(1982). Combinatorial Optimization: algorithms and Complexity, Prentice-Hall Inc., Englewood Cliffs, New Jersey

75. Potvin, J.Y. (1996):The Traveling Salesman Problem: A Neural Network Perspective.. ORSA Journal on Computing. 5:328-347.

76. Problem.. In: Gutin G and Punnen H (eds). The Traveling Salesman Problem and it Variations. Kluwer Academic Publishers.

77. Rachev and Ruschendorf (1993), constrained transportation Problems, Decision and Control, Proceedings of the 32nd IEEE conference ,Volume 3,Pages 2896-2900 .

78. Radin, L.R. (1998):Optimization in Operations Research. Prentice Hall Inc. New Jersey.

79. Rajkumar, K. and Narendran, T.T. (1996): A Heuristic for Sequencing PCB Assembly to Minimize Set-up Times, Production Planning & Control. 9(5): 465–476.

80. Raymond, T.C. (1969). Heuristic Algorithm for the Traveling-Salesman Problem. IBM Journal of Research and Development. 13:400-407.

81. Rego C and Glover, F (2002) Local Search and Metaheuristic, in G. Gutin and A.P. punnen (eds.): The Traveling Salesman Problem and its Variations, Kluwer, Dordrecht.

82. Riera-Ledesma, J. and Salazar-González, J.J. (2005). .A Heuristic Approach for The Travelling Purchaser Problem. European Journal of Operations Research. 162(1):142-152.

83. Robacker, J.T. (1955). Some Experiments on the Traveling-Salesman Problem. RAND research Memorandum.

84. Roberts, S.M. and Flores, B. (1966). An Engineering Approach to the Traveling Salesman Problem. Management Science. 13:269-288.

85. Robinson, B. (1949). On the Hamiltonian Game (A Traveling-Salesman Problem). research Memorandum.

86. Rossman, M.J and Twery, R.J. (1958). A Solution to the Travelling Salesman Problem. operations Research. 6:687.

87. Salia, E. (1995). Telecommunications in Ghana: A Changing Horizon, Africa Communications

88. Shapiro, D. (1966). Algorithms for the Solution of the Optimal Cost Traveling Salesman problem. Sc.D. Thesis, Washington University: St. Louis, MO.

89. Shapiro, D., (1966) Algorithms for the Solution of the Optimal Cost Traveling Salesman Problem, Sc.D. Thesis, Washington University, St. Louis,

90. Schrijver, A. (1986): Theory of Linear and Integer Programming, John Wiley, Chichester.

91. Tian, P. and Yang, S. (1993). An Improved Simulated Annealing Algorithm with Generic characteristics and Travelling Salesman Problem. Journal of Information and Optimization Science. 14(3):241-254.

92. Volgenant, T. and Jonker, R. (1982). A Branch and Bound Algorithm for the Symmetric Traveling Salesman Problem Based on the 1-Tree Relaxation. European Journal of operational Research. 9:83-89.

93. Walshaw, C.A. (2002). Multilevel Approach to the Travelling Salesman Problem. operations Research. 50(5):862-877.

APPENDIX A

Matlab Program

```
%function simanneal()

% **********Read distance (cost) matrix from Table 3.2 ******
clc
d = xlsread('dist.xls');
dorig = d;
start_time = cputime;
summ=0;
dim1 = size(d,1);
dim12 = size(d);
for i=1:dim1
d(i,i)=10e+06;
end
for i=1:dim1-1
for j=i+1:dim1
d(j,i)=d(i,j);
end
end
%d
% ****************Initialize all parameters**********************
d1=d;
tour = zeros(dim12);
cost = 0;
min_dist=();
short_path=();
%*****************************************************************
```

```
%************Initialize Simulated Annealing parameters************
%T0 Initial temperature is set equal to the initial solution value
Lmax = 400; %Maximum transitions at each temperature
ATmax = 200; %Maximum accepted transitions at each temperature
alfa = 0.99; %Temperature decrementing factor
Rf = 0.0001; %Final acceptance ratio
Iter_max = 1000000; %Maximum iterations 13
start_time = cputime;
diary output.txt
% *******generate Initial solution - find shortest path from each node****
% if node pair 1-2 is selected, make distance from 2 to each of earlier
%visited nodes very high to avoid a subtour
k = 1;
for i=1:dim1-1
min_dist(i) = min(d1(k,:));
short_path(i) = find((d1(k,:)==min_dist(i)),1);
cost = cost+min_dist(i);
k = short_path(i);
% prohibit all paths from current visited node to all earlier visited nodes
d1(k,1)=10e+06;
for visited_node = 1:length(short_path);
d1(k,short_path(visited_node))=10e+06;
end
end
tour(1,short_path(1))=1;
for i=2:dim1-1
tour(short_path(i-1),short_path(i))=1;
end
%Last visited node is k;
%shortest path from last visited node is always 1, where the tour
%originally started from
```

```
last_indx = length(short_path)+1;
short_path(last_indx)=1;
tour(k,short_path(last_indx))=1;
cost = cost+d(k,1);
% A tour is represented as a sequence of nodes starting from second node (as
% node 1 is always fixed to be 1
crnt_tour = short_path;
best_tour = short_path;
best_obj =cost;
crnt_tour_cost = cost;

obj_prev = crnt_tour_cost;
fprintf('\nInitial solution\n');
crnt_tour
fprintf('\nInitial tour cost = %d\t', crnt_tour_cost);
nbr = crnt_tour;
T0 = 1.5*crnt_tour_cost;
T=T0;
iter = 0;
iter_snc_last_chng = 0;
accpt_ratio =1;
%*********perform the iteration until one of the criteria is met***********
%1. Max number of iterations reached***************************************
%2. Acceptance Ratio is less than the threshold
%3. No improvement in last fixed number of iterations
While (iter < Iter_max && accpt_ratio > Rf)
iter = iter+1;
trans_tried = 0;
trans_accpt = 0;
while(trans_tried < Lmax && trans_accpt < ATmax)
```

```
trans_tried = trans_tried + 1;
city1 = round(random('uniform', 1, dim1-1));
city2 = round(random('uniform', 1, dim1-1));
While (city2 == city1)
city2 = round (random ('uniform', 1, dim1-1));
end
if (city2>city1)
i=city1;
j=city2;
else
i=city2;
j=city1;
end
nbr(i)=crnt_tour(j);
nbr(j)=crnt_tour(i);

if i==1
if j-i==1
nbr_cost=crnt_tour_cost-d(1,crnt_tour(i))+d(1,crnt_tour(j))-
d(crnt_tour(j),crnt_tour(j+1))+d(crnt_tour(i),crnt_tour(j+1));
else
nbr_cost=crnt_tour_cost-d(1,crnt_tour(i))+d(1,crnt_tour(j))-
d(crnt_tour(j),crnt_tour(j+1))+d(crnt_tour(i),crnt_tour(j+1))-
d(crnt_tour(i),crnt_tour(i+1))+d(crnt_tour(j),crnt_tour(i+1))-d(crnt_tour(j-
1),crnt_tour(j))+d(crnt_tour(j-1),crnt_tour(i));
end
else
if j-i==1
nbr_cost=crnt_tour_cost-d(crnt_tour(i-1),crnt_tour(i))+d(crnt_tour(i-1),crnt_tour(j))-
d(crnt_tour(j),crnt_tour(j+1))+d(crnt_tour(i),crnt_tour(j+1));
else
```

```
nbr_cost=crnt_tour_cost-d(crnt_tour(i-1),crnt_tour(i))+d(crnt_tour(i-1),crnt_tour(j))-
d(crnt_tour(j),crnt_tour(j+1))+d(crnt_tour(i),crnt_tour(j+1))-
d(crnt_tour(i),crnt_tour(i+1))+d(crnt_tour(j),crnt_tour(i+1))-d(crnt_tour(j-
1),crnt_tour(j))+d(crnt_tour(j-1),crnt_tour(i));
end
end
delta = nbr_cost - crnt_tour_cost;
prob1 = exp(-delta/T);
prob2 = random('uniform',0,1);
if(delta < 0 || prob2 < prob1)
summ = summ+delta;
crnt_tour = nbr;
crnt_tour_cost = nbr_cost;
trans_accpt = trans_accpt + 1;
if crnt_tour_cost < best_obj
best_obj = crnt_tour_cost;
best_tour = crnt_tour;
end
else
nbr = crnt_tour;

nbr_cost = crnt_tour_cost;
end


end
accpt_ratio = trans_accpt/trans_tried;
fprintf('\niter#=%d\t, T=%2.2f\t,obj= %d\t, accpt ratio=%2.2f', iter,T,crnt_tour_cost,accpt_ratio);
if crnt_tour_cost == obj_prev
iter_snc_last_chng = iter_snc_last_chng + 1;
else
iter_snc_last_chng = 0;
```

```
end
if iter_snc_last_chng == 10
fprintf('\n No change since last 10 iterations');
break;
end
obj_prev = crnt_tour_cost;
T = alfa*T;
iter = iter + 1;
end
fprintf('\nbest obj = %d', best_obj);
fprintf('\n best tour\n');
best_tour
end_time = cputime;
exec_time = end_time - start_time;
fprintf('\ntime taken = %f\t\n', exec_time);
diary off
```