KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF MATHEMATICS

APPLICATION OF DYNAMIC PROGRAMMING TO THE TRAVELING SALESMAN PROBLEM: CASE STUDY GHANA EDUCATION SERVICE INSPECTORATE DIVISION, GREATER ACCRA METROPOLIS

A Thesis Submitted to the Department of Mathematics, Kwame Nkrumah University of Science and Technology in Partial fulfilment of the requirement for the degree of

Master of Philosophy
(Industrial Mathematics)

By

GANDIS GARIBA KARMASUN

May, 2013

DECLARATION

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

Gandis Gariba Karmasun ....
(PG6320611)
(Name & ID of Student) (Signature) (Date)

Certified by:
Prof. S.K. Amponsah ....
....
(Supervisor's Name) (Signature) (Date)

Mr. K. F. Darkwah ....
....
(Head of Dept) (Signature) (Date)

Prof. I.K. Dontwi ....
....
(Dean, IDL) (Signature) (Date)

DEDICATION

To the King Eternal without whom I can do nothing

To the entire Karmasun family for the unflinching support they gave me.

ACKNOWLEDGEMENT

To God be the glory, for seeing me through this course.

I wish to express my heartfelt gratitude to Prof. S.K. Amponsah for encouraging me to embark on this programme and also for taking time out of his busy schedule to read through every line of my thesis.

I am also very grateful to all my lecturers in the Department of Mathematics (IDL), Kwame Nkrumah University of Science and Technology (KNUST) for being helpful and impacting knowledge in me in diverse ways.

I shall forever remain indebted to Mr. Maxwell Ofosu for his invaluable assistance throughout this programme.

Finally, my thanks go to all who in diverse ways helped me to successfully complete this thesis.

May God richly bless you all.

ABSTRACT

The Traveling Salesman Problem (TSP) arises in many different contexts. In this thesis, we model and solve the Inspectorate division problem of the Ghana Education Service, Accra Metropolis as a Traveling Salesman Problem. The problem is formulated as a network of distances and the solution is presented based on dynamic programming to identify the maximum number of schools an officer can visit with minimum distance. Data on distances were obtained from the Inspectorate Division-GES, Accra. In comparison, the maximum number of schools the officers normally visited was three (3) on a route from past experience whiles our proposed method increased the number of schools to be visited at the same time and condition to five(5).

TABLE OF CONTENTS

LIST OF TABLES

LISTS OF TABLES

CHAPTER ONE

1.0 INTRODUCTION

Route management is very important to make sure the user arrive at his/her destination much faster. In the transportation industry, the route that will be generated should consider the cost and time constraint, which is dependently on the distance of the route.

The Traveling Salesman Problem (TSP) being one of the most well-known optimization problems, has attracted the attention of many researchers over the last decade because of it's simple problem description but simultaneously its associated difficulty in obtaining an optimal solution efficiently. The traveling-salesman problem involves a salesman who must make a tour to a number of cities using the shortest path available. For each number of cities n, the number of paths, which must be explored is n!, causing this problem to grow exponentially rather than as a polynomial (Schmitt and Amini, 1998).

The TSP is a classic model for various production and scheduling problems. Many production and scheduling problems ultimately can be reduced to the simple concept that there is a salesperson that must travel from city to city (visiting each city exactly once) and wishes to minimize the total distance travelled during his tour of all the n cities.

1.1 BACKGROUND OF THE STUDY

The TSP is a problem whose solution has eluded many mathematicians for years. Currently there is no solution to the TSP that has satisfied mathematicians. The TSP was developed in the 1800's by Sir William Rowan Hamilton and Thomas Penyngton Kirkman, Irish and British mathematicians, respectively. Specifically, Hamilton was the creator of the Icosian Game in 1857. It was a pegboard with twenty holes that required each vertex to be visited only once, no edge to be visited more than once, and the ending point being the same as the starting point. This kind of path was eventually referred to as a Hamiltonian circuit. However, the general form of the TSP was first studied by Karl Menger in Vienna and Harvard in the late 1920's or early 1930's.

TSPs were first studied in the 1930s by mathematician and economist Karl Menger in Vienna and Harvard. It was later investigated by Hassler Whitney and Merrill Flood at Princeton. The TSP is about finding a Hamiltonian path with minimum cost. It is common in areas such as logistics, transportation and semiconductor industries. For instance, finding an optimized scan chains route in integrated chips testing, parcels collection and sending in logistics companies, are some of the potential applications of TSP.

The TSP is a classic model for various production and scheduling problems. Many production and scheduling problems ultimately can be reduced to the simple concept that there is a salesperson that must travel from city to city (visiting each city exactly once) and wishes to minimize the total distance travelled during his tour of all n cities.

The TSP deals with creating the ideal path that a salesman would take while travelling between cities. The solution to any given TSP would be the shortest way to visit a finite number of cities, visiting each city only once, and then returning to the starting point.

The aim of the TSP is to find the cheapest path reaching all elements in a given set of cities (nodes) where the cost of travel between each pair of them is given, including the return to the starting point.

The TSP has several applications even in its purest formulation, such as planning, logistics, and the manufacture of microchips. Direct application of the TSP is in the drilling problem of printed circuit boards (PCBs)(Grötschel et al., 1991). To connect a conductor on one layer with a conductor on another layer, or to position the pins of integrated circuits, holes have to be drilled through the board. The holes may be of different sizes. To drill two holes of different diameters consecutively, the head of the machine has to move to a tool box and change the drilling equipment. This is quite time consuming. Thus it is clear that one has to choose some diameter, drill all holes of the same diameter, change the drill, drill the holes of the next diameter, etc. Thus, this drilling problem can be viewed as a series of TSPs, one for each hole diameter, where the 'cities' are the initial position and the set of all holes that can be drilled with one and the same drill. The 'distance' between two cities is given by the time it takes to move the drilling head from one position to the other. The aim is to minimize the travel time for the machine head.

The problem of placing the vanes in the best possible way can be modeled as a TSP with a special objective function. Analysis of the structure of crystals (Bland and Shallcross, 1989) is an important application of the TSP. Here an X-ray diffractometer is used to obtain information about the structure of crystalline material. To this end a detector measures the intensity of X-ray reflections of the crystal from various positions. Whereas the measurement itself can be accomplished quite fast, there is a considerable overhead in positioning time since up to hundreds of thousands positions have to be realized for some experiments. In the two examples that we refer to, the positioning involves moving four motors. The time needed to move from one position to the other can be computed very accurately. The result of the experiment does not depend on the sequence in which the measurements at the various positions are taken. However, the total time needed for the experiment depends on the sequence. The problem consists of finding a sequence that minimizes the total positioning time. This leads to a traveling salesman problem.

Lenstra and Rinnooy (1974) reported a special case of connecting components on a computer board. Modules are located on a computer board and a given subset of pins has to be connected. In contrast to the usual case where a Steiner tree connection is desired, here the requirement is that no more than two wires are attached to each pin. Hence we have the problem of finding a shortest Hamiltonian path with unspecified starting and terminating points. A similar situation occurs for the so-called test bus wiring. To test the manufactured board one has to realize a connection which enters the board at some specified point, runs through all the modules, and terminates at some specified point. For each module we also have a specified entering and leaving point for this test wiring. This problem also amounts to solving a Hamiltonian path problem with the difference that the distances are not symmetric and that starting and terminating point are specified. Several applications such as; The order-picking problem in warehouses (Ratliff and Rosenthal, 1983), Vehicle routine (Lenstra and Rinnooy, 1974), Mask plotting in Printed Circuit Board(PCB) production (Grötschel et al., 1991), Printing press scheduling problem Gorenstein (1970, and Carter and Ragsdale (2002), School bus routing problem(Angel et al., 1972), Crew scheduling problem(Svestka and Huckfeldt, 1973), Interview scheduling problem (Gilbert and Hofstra, 1992),Hot rolling scheduling problem(Tang et al., 2000).

1.1.1 Definition

Given a set of cities and the cost of travel (or distance) between each possible pairs, the TSP, is to find the best possible way of visiting all the cities and returning to the starting point that minimize the travel cost (or travel distance).

1.1.2 Classification

Broadly, the TSP is classified as symmetric travelling salesman problem (sTSP), a symmetric travelling salesman problem (aTSP), and multi travelling salesman problem (mTSP).

sTSP: Let $V= \{v_1 ,......,v_n\}$ be a set of cities, $A=\{(r,s):r,s\in V\}$ be the edge set, and $d_{rs} \neq d_{sr}$ be a cost measure associated with edge $(r,s)\in A$.

The sTSP is the problem of finding a minimal length closed tour that visits each city once. In this case cities $v_i\in V$ are given by their coordinates (x_i, y_i) and d_{rs} is the Euclidean distance between r and s then we have an Euclidean TSP.

aTSP: If $d_{rs} \neq d_{sr}$for at least one (r,s) then the TSP becomes an aTSP.

mTSP: The mTSP is defined as: In a given set of nodes, let there be m salesmen located at a single depot node. The remaining nodes (cities) that are to be visited are intermediate nodes. Then, the mTSP consists of finding tours for all m salesmen, who all start and end at the depot, such that each intermediate node is visited exactly once and the total cost of visiting all nodes is minimized. The cost metric can be defined in terms of distance, time, etc. Possible variations of the problem are as follows: Single verses multiple depots: In the single depot, all salesmen finish their tours at a single point while in multiple depots the salesmen can either return to their initial depot or can return to any depot keeping the initial number of salesmen at each depot remains the same after the travel. Number of salesmen: The number of salesman in the problem can be fixed or a bounded variable. Cost: When the number of salesmen is not fixed, then each salesman usually has an associated fixed cost incurring whenever this salesman is used. In this case, the minimizing the requirements of salesman also becomes an objective. Timeframe: Here, some nodes need to be visited in a particular time periods that are called time windows which is an extension of the mTSP, and referred as multiple traveling salesman problem with specified timeframe (mTSPTW). The application of mTSPTW can be very well seen in the aircraft scheduling problems. Other constraints: Constraints can be on the number of nodes each salesman can visit, maximum or minimum distance a salesman travels or any other constraints. The mTSP is generally treated as a relaxed vehicle routing problems (VRP) where there is no restrictions on capacity. Hence, the formulations and solution methods for the VRP are also equally valid and true for the mTSP if a large capacity is assigned to the salesmen (or vehicles). However, when there is a single salesman, then the mTSP reduces to the TSP (Bektas, 2006).

1.2 PROBLEM STATEMENT

The TSP can easily be stated as follows: A salesman wants to visit n distinct cities and then returns home. He wants to determine the sequence of the travel so that the overall traveling distance is minimized while visiting each city not more than once. Although the TSP is conceptually simple, it is difficult to obtain an optimal solution. In an n-city situation, any permutation of n cities yields a possible solution. As a consequence, n! possible tours must be evaluated in the search space. Different formulations are put forward for TSP and its variants.

1.3 OBJECTIVE OF THE STUDY

The main objective of this study is to use TSP model and solve the Inspectorate division problem of the Ghana Education Service, Accra Metropolis as a Travelling Salesman problem.

1.4 METHODOLOGY

We shall propose the Dynamic Programming Approach in solving our problem. First, the algorithm will be presented. A real life computational study shall be performed and our data for the study will be captured using Euclidean distance formula between all pairs of destinations (Inspection centres), from the Inspectorate Division.

1.5 JUSTIFICATION

The traveling salesman problems are widely used in modeling most of the real-life industrial applications, and very interesting, from the perspective of computer science because of the time complexities in some of the well-known algorithms used in solving the problems. These have made the studies of traveling salesman problems and their algorithms an important area of research in the contribution to academic knowledge and the benefit of the economy as a whole, hence the reason for solving the traveling salesman problem.

1.6 LIMITATIONS OF THE STUDY

This study is limited to ten (10) selected schools in the Greater Accra Metropolis of Ghana. The findings of this study could however be adapted to other schools in the region.

1.7 ORGANIZATION OF THE STUDY

In chapter one, we presented the background, problem statement and objective of the study. The justification, methodology and limitation of the study were also put forward.

In chapter two, related work on the Traveling Salesman Problem will be discussed.

Chapter three presents the dynamic programming approach proposed to solve our problem.

Chapter four is devoted for the data collection and analysis of the study.

Chapter five, which is the last chapter of the study, presents the conclusions and recommendations of the study.

CHAPTER TWO

LITERATURE REVIEW

The Inspectorate division of the Ghana Education Service, Accra Metropolis has its administrative organs consisting of a central office and seven sub-metropolitan offices and are headed by the Director of Education.

Due to the large size of the Metropolis and the number of schools involved, it has not been easy to work with the stipulated number of officers. Apart from the seven sub-metropolitan offices, there are thirty-three Circuits each being managed by Supervisors who promote effective supervision of schools in their respective circuits.

The Inspectorate division has as part of its core responsibility to;

- Supervise private and public schools
- Ensure adherence to regulations and educational policies.
- Provide all necessary logistics

Applegate et al., (1994) solved TSP containing seven thousand, three hundred and ninety-seven (7397) cities. Later in 1998, the authors solved the same problem using thirteen thousand, five hundred and nine (13,509) cities in the United States. In 2001, the authors found the optimal tour of fifteen thousand, one hundred and twelve (15,112) cities in Germany. Later in 2004, TSP of visiting all twenty-four thousand, nine hundred seventy-eight (24,978) cities in Sweden was solved; a tour of length of approximately seventy-two thousand, five hundred (72,500) kilometers was found and it was proven that no shorter tour exists. This is currently the largest solved TSP.

The Travelling Salesman Problem (TSP) is a typical example of a very hard combinatorial optimization problem. The problem is to find the shortest tour that passes through each vertex in a given graph exactly once. The TSP has received considerable attention over the last two decades and various approaches are proposed to solve the problem. As early as in 1954, optimal solution to travelling salesman problem with forty-nine (49) number of cities has been obtained. In 1980's, Crowder and Padberg solved the problem with 318 cities using cutting-plane method.

The Traveling Salesman problem (TSP) is one of the benchmark and old problems in Computer Science and Operations Research. It can be stated as: A network with 'n' nodes (or cities), with 'node 1' as 'headquarters' and a travel cost (or distance, or travel time etc.,) matrix C= (C_{ij}) of order n associated with ordered node pairs (i, j) is given. In 1991 Grötschel and Holland proposed a solution for large scale TSP.

On the basis of the structure of the cost matrix, the TSPs are classified into two groups – symmetric and asymmetric. The TSP is symmetric if $C_{ij} = C_{ji}$, for all i, j and asymmetric otherwise. For an n-city asymmetric TSP, there are (n-1)! possible solutions, one or more of which gives the minimum cost. Applegate et al., (1998, 2001 and 2004) proposed solution for TSP using cuts that solved thirteen thousand, five hundred and nine (13509), fifteen thousand, one hundred and twelve (15112) and twenty-four thousand, nine hundred and seventy-eight (24978) cities respectively. The solutions worked well up to five thousand (5000) cities and can be used up to thirty-three thousand, eight hundred and ten (33,810) cities.

For an n-city symmetric TSP, there is $(n-1)! / 2$ possible solutions along with their reverse cyclic permutations having the same total cost. In either case, the number of solutions becomes extremely large for even moderately large n so that an exhaustive search is impracticable. There are mainly three reasons why TSP has attracted the attention of many researchers and remains an active research area. First, a large number of real-world problems can be modeled by TSP. Secondly, it was proved to be NP-Complete problem. Thirdly, NP-Complete problems are intractable in the sense that no one has found any really efficient way of solving them for large problem size. Also, NP-complete problems are known to be more or less equivalent to each other; if one knew how to solve one of them one could solve the lot. In the 1970's, Held and Karp used minimum spanning tree to solve the TSP with sixty-four (64) cities.

Broadly, the TSP is classified as symmetric travelling salesman problem (sTSP), a symmetric travelling salesman problem (aTSP), and multi travelling salesman problem (mTSP). With sTSP: Let $V = \{v_1, \ldots\ldots, v_n\}$ be a set of cities, $A = \{(r,s) : r,s \in V\}$ be the edge set, and $d_{rs} = d_{sr}$ be a cost measure associated with edge $(r,s) \in A$. The sTSP is the problem of finding a minimal length closed tour that visits each city once. In this case, cities $v_i \in V$ are given by their coordinates (x_i, y_i) and d_{rs} is the Euclidean distance between r and s resulting in Euclidean TSP. With aTSP**:** If $d_{rs} \neq d_{sr}$ for at least one (r, s) then the TSP becomes an aTSP with mTSP**:** The mTSP is defined as: In a given set of nodes, there are m salesmen located at a single depot node. The remaining nodes (cities) that are to be visited are intermediate nodes. Then, the mTSP consists of finding tours for all m salesmen, who all start and end at the depot, such that each intermediate node is visited exactly once and the total cost of visiting all nodes is minimized. The cost metric can be defined in terms of distance, time, etc. In 1971, Bellmore and Malone solved TSP using sub tour elimination.

One example of the usefulness of the TSP is a direct application of the TSP in the drilling problem of printed circuit boards (PCBs) (Grötschel et al., 1991). To connect a conductor on one layer with a conductor on another layer, or to position the pins of integrated circuits, holes have to be drilled through the board. The holes may be of different sizes. To drill two holes of different diameters consecutively, the head of the machine has to move to a tool box and change the drilling equipment. This is quite time consuming. Thus, it is clear that one has to choose some diameter, drill all holes of the same diameter, change the drill, drill the holes of the next diameter, etc. Thus, this drilling problem can be viewed as a series of TSPs, one for each hole diameter, where the 'cities' are the initial position and the set of all holes that can be drilled with one and the same drill. The 'distance' between two cities is given by the time it takes to move the drilling head from one position to the other. The aim is to minimize the travel time for the machine head.

Travel Salesman Problem (TSP) has been applied to solve a number of real-life problems, including overhauling gas turbine engines (Plante et al., 1987), X-Ray crystallography (Bland and Shallcross, 1989; Dreissig and Uebach, 1990), Computer wiring (Lenstra and Rinnooy, 1974). The order-picking problem in warehouses (Ratliff and Rosenthal, 1983), and Mask plotting in Printed Circuit Board (PCB) production (Gottschalk et al., 1991). Thus, TSP has played an important role in supporting managerial decisions in the areas of printing press scheduling, school bus routing, crew scheduling, interview scheduling, hot rolling scheduling, mission planning, and design of global navigation satellite system surveying networks.

Order picking in conventional warehouse environments involves determining a sequence in which to visit unique locations where each part in the order is stored, and therefore can often be modeled as s TSP. With computer tracking of inventories, parts may now be stored in multiple locations, simplifying the replenishment of inventory and eliminating the need to reserve space for each item. In such an environment, order picking requires choosing a subset of the locations that store an item to collect the required quantity. Thus, both the assignment of inventory to an order and the associated sequence in which the selected locations are visited affect the cost of satisfying an order.

Fagerholt and Christiansen (2000) studied a TSP with allocation, time window, and precedence constraints (TSP-ATWPC). The TSP-ATWPC occurs as a sub problem involving optimally sequencing a given set of port visits in a real bulk ship scheduling problem, which is a combined multi-ship pickup and delivery problem with time windows and multi-allocation problem. Each ship in the fleet is equipped with a flexible cargo hold that can be partitioned into several smaller holds in a given number of ways, thus allowing multiple products to be carried simultaneously by the same ship. The allocation constraints of the TSP-ATWPC ensure that the partition of the ship's flexible cargo hold and the allocation of cargoes to the smaller holds are feasible throughout the visiting sequence.

Ruiz et al., (2004) proposed a two-stage exact approach for solving a real-life problem of the TSP. In the first stage, all the feasible routes are generated by means of an implicit enumeration algorithm; thereafter, an integer programming model is designed to select in the second stage the optimum routes from the set of feasible routes. The integer model uses a number of 0-1 variables ranging from 2,000 to 15,000 and arrives at optimum solutions in an average time of sixty seconds (for instances up to 60 clients). The developed system was tested with a set of real instances and, in a worst-case scenario (up to 60 clients), the routes obtained ranged from a 7% to 12% reduction in the distance traveled and from a 9% to 11% reduction in operational costs.

Analysis of the structure of crystals is an important application of the TSP. Here an X-ray diffractometer is used to obtain information about the structure of crystalline material. To this end, a detector measures the intensity of X-ray reflections of the crystal from various positions. Whereas the measurement itself can be accomplished quite fast, there is a considerable overhead in positioning time since up to hundreds of thousands positions have to be realized for some experiments. In the two examples referred to, the positioning involves moving four motors. The time needed to move from one position to the other can be computed very accurately. The result of the experiment does not depend on the sequence in which the measurements at the various positions are taken. However, the total time needed for the experiment depends on the sequence. Therefore, the problem consists of finding a sequence that minimizes the total positioning time. This led to a traveling salesman problem, studied by Bland and Shallcross (1989) and Dreissig and Uebach (1990).

Lenstra and Rinnooy (1974) presented a special case of connecting components on a computer board. Modules are located on a computer board and a given subset of pins has to be connected. In contrast to the usual case where a Steiner tree connection is desired, here the requirement is no more than two wires are attached to each pin. Hence there is the problem of finding a shortest Hamiltonian path with unspecified starting and terminating points. A similar situation occurs for the so-called test bus wiring. To test the manufactured board one has to realize a connection which enters the board at some specified point, runs through all the modules, and terminates at some specified point. For each module there is also a specified entering and leaving point for this test wiring. This problem also amounts to solving a Hamiltonian path problem with the difference that the distances are not symmetric and that starting and terminating point are specified.

Ratliff and Rosenthall (1983) studied a problem of order-picking associated with material handling in a warehouse. Assume that at a warehouse an order arrives for a certain subsets of the items stored in the warehouse. Some vehicle has to collect all items of this order to ship them to the customer. The relation to the TSP is immediately seen. The storage locations of the items correspond to the nodes of the graph. The distance between two nodes is given by the time needed to move the vehicle from one location to the other. The problem of finding a shortest route for the vehicle with minimum pickup time can now be solved as a TSP.

One of the major and primary applications of the multiple traveling salesperson problems arises in scheduling a printing press for a periodical with multi-editions. Here, there exist five pairs of cylinders between which the paper rolls and both sides of a page are printed simultaneously. There exist three kinds of forms, namely 4-, 6-, and 8-page forms, which are used to print the editions. The scheduling problem consists of deciding which form will be on which run and the length of each run. In the multiple salesperson problem vocabulary, the plate change costs are the inter-city costs. Gorenstein (1970) and Carter and Ragsdale (2002) presented a real-life application of the above problem.

Angel et al., (1972) investigated the problem of scheduling buses as a variation of the multiple traveling salesperson problems with some side constraints. The objective of the scheduling is to obtain a bus loading pattern such that the number of routes is minimized, the total distance travelled by all buses is kept at minimum, no bus is overloaded and the time required to traverse any route does not exceed a maximum allowed policy.

Ryan et al., (1998) presented a model of the routing problems arising in the planning of unmanned aerial vehicle applications as a multiple traveling salesman problem, and proposed a tabu search approach for solving the problem.

A very recent and an interesting application of the multiple traveling salesperson problems, presented by Saleh and Chelouah (2004) arise in the design of Global Navigation Satellite System (GNSS) surveying networks. A Global Navigation Satellite System (GNSS) is a space-based satellite system which provides coverage for all locations worldwide and is quite crucial in real-life applications such as early warning and management for disasters, environment and agriculture monitoring, etc. The goal of surveying is to determine the geographical positions of unknown points on and above the earth using satellite equipment. These points, on which receivers are placed, are co-ordinated by a series of observation sessions. When there are multiple receivers or multiple working periods, the problem of finding the best order of sessions for the receivers can be formulated as a multiple traveling salesman problem.

Tobias and Osten (2007) introduced an optical method based on white light interferometer in order to solve the well-known NP–complete traveling salesman problem. According to the authors it was the first time that a method for the reduction of non–polynomial time to quadratic time has been proposed. The authors showed that this achievement is limited by the number of available photons for solving the problem. It was turned out that this number of photons is proportional to N^N for a traveling salesman problem with N cities and that for large numbers of cities the method in practice therefore is limited by the signal–to–noise ratio.

Kaur and Murugapan (2008) presented a novel hybrid genetic algorithm for solving Traveling Salesman Problem (TSP) based on the Nearest Neighbour heuristics and pure Genetic Algorithm (GA). The hybrid genetic algorithm exponentially derives higher quality solutions in relatively shorter time for hard combinatorial real world optimization problems such as Traveling Salesman Problem (TSP) than the pure GA. The hybrid algorithm outperformed the NN algorithm and the pure Genetic Algorithm taken separately. The hybrid genetic algorithm is designed and experimented against the pure GA and the convergence rate improved by more than 200% and the tour distance improved by 17.4% for 90 cities. These results indicate that the hybrid approach is promising and it can be used for various other optimization problems.

Traveling Salesman Problems with profits (TSPs with profits) are a generalization of the Traveling Salesman Problem, where it is not necessary to visit all vertices. A profit is associated with each vertex. The overall goal is the simultaneous optimization of the collected profit and the travel costs. These two optimization criteria appear either in the objective function or as a constraint.

Dominique et al., (2003) studied a classification of TSPs with profits is proposed, and the existing literature is surveyed. Different classes of applications, modeling approaches, and exact or heuristic solution techniques are identified and compared. Conclusions emphasize the interest of this class of problems, with respect to applications as well as theoretical results.

Cerny (1985) presented a Monte Carlo algorithm to find approximate solutions of the traveling salesman problem. The algorithm generates randomly the permutations of the stations of the traveling salesman trip, with probability depending on the length of the corresponding route. Reasoning by analogy with statistical thermodynamics, we use the probability given by the Boltzmann-Gibbs distribution. Surprisingly enough, using this simple algorithm, one can get very close to the optimal solution of the problem or even find the true optimum. The author demonstrates this on several examples. The author conjectures that the analogy with thermodynamics can offer a new insight into optimization problems and can suggest efficient algorithms for solving them.

Viera et al., (2002) put forward an approach to the well-known traveling salesman problem (TSP) via competitive neural networks. The neural network model adopted in this work is the Kohonen network or self-organizing maps (SOM), which has important topological information about its neurons configuration. The author was concerned with the initialization aspects, parameters adaptation, and the complexity analysis of the proposed algorithm. The modified SOM algorithm proposed by the author has shown better results when compared with others neural network based approaches to the TSP.

The traveling salesman problem with precedence constraints (TSPPC) is one of the most difficult combinatorial optimization problems. Chiung (2002) presented an efficient genetic algorithm (GA) to solve the TSPPC is presented. The key concept of the proposed GA is a topological sort (TS), which is defined as an ordering of vertices in a directed graph. Also, a new crossover operation is developed for the proposed GA. The results of numerical experiments showed that the proposed GA produces an optimal solution and shows superior performance compared to the traditional algorithms.

The classical traveling-salesman problem involves the establishment of a tour around a set of points in a plane such that each point is intersected only once and the circuit is of minimal total length. When the length of a salesman's tour cannot exceed a specified constant, the problem becomes that of finding the fewest number of salesmen such that every city is visited by a salesman and the length of each salesman's tour does not exceed a specified constant. This is the chromatic traveling-salesmen problem. An algorithm for this problem was presented by Milton et al., (2010) which was used to create periodic markets in parts of Sierra Leone. Fifteen rural areas were examined from Sierra Leone, and weekly market places were identified in each area. Salesmen were to be assigned to an area so that each market place was visited and each tour (or periodic ring) did not exceed forty hours. The chromatic traveling-salesmen algorithm was used to minimize the number of periodic rings needed for each area and provide the specific tour for each ring.

The Symmetric Circulant Travelling Salesman Problem asks for the minimum cost tour in a symmetric circulant matrix. The computational complexity of this problem is not known – only upper and lower bounds have been determined. Ivan and Federico (2008) presented a characterization of the two-stripe case. Instances where the minimum cost of a tour is equal to either the upper or lower bound are recognized. A new construction providing a tour is proposed for the remaining instances, and this leads to a new upper bound that is closer than the previous one.

The Traveling Salesman Problem is known to be a combinatorial optimization problem which belongs to NP-hard (a class of problems which does not allow polynomial time solution). Recently, various types of TSP are studied on the Web and the best solutions up to date are open to the public. The initial solution for a given TSP can be easily obtained by the well-known methods such as greedy, nearest neighbor, and saving method.

Bernd and Peter (1996) presented an approach which incorporates problem specific knowledge into a genetic algorithm which is used to compute near-optimum solutions to traveling salesman problems (TSP). The approach is based on using a tour construction heuristic for generating the initial population, a tour improvement heuristic for finding local optimal in a given TSP search space, and new genetic operators for effectively searching the space of local optima in order to find the global optimum. The quality and efficiency of solutions obtained for a set of TSP instances containing between three hundred and eighteen (318) and fourteen thousand (1400) cities are presented.

Gunter (1992) considered the special case of the Euclidean Traveling Salesman Problem where the given points lie on a small number (N) of parallel lines. Such problems arise for example in the fabrication of printed circuit boards, where the distance travelled by a laser which drills holes in certain places of the board should be minimized. By a dynamic programming algorithm, we can solve the N-line traveling salesman problem for *n* points in time n^N, for fixed N, i. e., in polynomial time. This extends a result of Cutler (1980) for 3 lines. The parallelity condition can be relaxed to point sets which lie on N "almost parallel" line segments. The author gave a characterization of the allowed segment configurations by a set of forbidden sub configurations.

The Traveling Salesman Problem is a well-studied combinatorial optimization problem with a wide spectrum of applications and theoretical value. Hains (2010) designed a new recombination operator known as Generalized Partition Crossover (GPX) for the TSP. GPX is unique among other recombination operators for the TSP in that recombining two local optima produces new local optima with a high probability. Thus the operator can 'tunnel' between local optima without the need for intermediary solutions. The operator is respectful, meaning that any edges common between the two parent solutions are present in the offspring, and transmits alleles, meaning that offspring are comprised only of edges found in the parent solutions. The author designed a hybrid genetic algorithm, which uses local search in addition to recombination and selection, specifically for GPX. The author showed that this algorithm outperforms Chained Lin-Kernighan, a state-of-the-art approximation algorithm for the TSP. The author next analyzed these algorithms to determine why the algorithms are not capable of consistently finding a globally optimal solution. The results revealed a search space structure which the author called 'funnels' because they are analogous to the funnels found in continuous optimization. Funnels are clusters of tours in the search space that are separated from one another by a non-trivial distance. The author found that funnels can trap Chained Lin-Kernighan, preventing the search from finding an optimal solution. The data used indicated that, under certain conditions, GPX can tunnel between funnels, explaining the higher frequency of optimal solutions produced by the author's hybrid genetic algorithm using GPX.

Traveling salesman problem is a classical complete nondeterministic polynomial problem. It is significant to solve Multiple Traveling Salesman Problems (MTSP). Previous researches on multiple traveling salesman problems are mostly limited to the kind that employed total-path-shortest as the evaluating rule, but little notice is made on the kind that employed longest-path-shortest as the evaluating rule.

Hai-Long et al., (2009) studied this problem and employed genetic algorithm to optimize it and decoding method with matrix was proposed. The method could solve symmetric and asymmetric MTSP. Symmetric and asymmetric multiple traveling salesman problems were simulated and different crossover operators were compared.

Logistics Management sometimes involves pickup as well as delivery along the route. Courier service is a typical example. The imposition of precedence constraints among the places to be visited constitutes a variant of the classical Travelling Salesman Problem (TSP). This well-known NP-hard problem is often solved by heuristics. The Precedence-Constrained TSP that incorporates Delivery and Pickup (PCTDP) is a much harder problem to solve. Ganesh and Narendran (2005) studied the PCTDP and presented a three-stage heuristic using clustering and shrink-wrap algorithms for finding an initial path as well as genetic algorithms for the final search to obtain the best solution. The proposed heuristic is tested over a range of trial datasets and is found to give encouraging results. With its ability to provide solutions of good quality at low computing times, the proposed heuristic has ample scope for application as an automated scheduler when implemented at the site of a logistics-planning cell.

Most researches in evolutionary computation focus on optimization of static and no-change problems. Many real world optimization problems however are actually dynamic, and optimization methods capable of continuously adapting the solution to a changing environment are needed. Yan et al., (2004) presented an approach to solving dynamic TSP. A dynamic TSP is harder than a general TSP, which is a NP-hard problem, because the city number and the cost matrix of a dynamic TSP are time varying. The authors proposed an algorithm to solve the dynamic TSP problem, which is the hybrid of EN and Inver-Over algorithm. From the results of the experiment, the authors concluded their algorithm was effective

Vardges (2009) proposed an LP relaxation for ATSP. The author introduced concepts of high-value and high-flow cycles in LP basic solutions and show that the existence of this kind of cycles would lead to constant-factor approximation algorithms for ATSP. The existence of high-flow cycles is motivated by computational results and theoretical observations.

The Multiple Traveling Salesmen Problem (MTSP) is an extension of the travelling salesman problem with many production and scheduling applications. The TSP has been well studied including methods of solving the problem with genetic algorithms. The MTSP has also been studied and solved with GAs in the form of the vehicle-scheduling problem. Carter (2003) presented a new modeling methodology for setting up the MTSP to be solved using a GA. The advantages of the new model are compared to existing models both mathematically and experimentally. The model is also used to model and solve a multi line production problem in a spreadsheet environment. The new model proves itself to be an effective method to model the MTSP for solving with GAs. The concept of the MTSP is then used to model and solve with a GA the use of one salesman make many tours to visit all the cities instead of using one continuous trip to visit all the cities. While this problem uses only one salesman, it can be modeled as a MTSP and has many applications for people who must visit many cities on a number of short trips. The method used effectively creates a schedule while considering all required constraints.

The traveling salesperson problem (TSP) is a classic model for various production and scheduling problems. Many production and scheduling problems ultimately can be reduced to the simple concept that there is a salesperson that must travel from city to city (visiting each city exactly once) and wishes to minimize the total distance traveled during his tour of all n cities. Obtaining a solution to the problem of a salesperson visiting n cities while minimizing the total distance traveled is one of the most studied combinatorial optimization problems. While there are variations of the TSP, the Euclidean TSP is NP-hard. Schmitt and Amini(1998) and Falkenauer (1998) studied a model with the interest in this particular type of problem being how common the problem is and how difficult the problem is to solve when n becomes sufficiently large.

The traveling salesman problem (TSP) has been an early proving ground for many approaches to combinatorial optimization, including classical local optimization techniques as well as many of the more recent variants on local optimization, such as simulated annealing, tabu search, neural networks, and genetic algorithms. David and Lyle (1995) studied how these various approaches have been adapted to the TSP and evaluates their relative success in this perhaps a typical domain from both a theoretical and an experimental point of view.

The traveling salesman problem with precedence constraints (TSPPC) is one of the most difficult combinatorial optimization problems. Chiung et al., (2000) presented an efficient Genetic Algorithm (GA) to solve the TSPPC. The key concept of the proposed GA is a topological sort (TS), which is defined as an ordering of vertices in a directed graph. Also, a new crossover operation is developed for the proposed GA. The results of numerical experiments show that the proposed GA produces an optimal solution and shows superior performance compared to the traditional algorithms.

Many real-life industrial applications involve finding a Hamiltonian path with minimum cost. Some instances that belong to this category are transportation routing problem, scan chain optimization and drilling problem in integrated circuit testing and production. Li-Pei et al., (2001) presented a Bee Colony Optimization (BCO) algorithm for Traveling

Salesman Problem (TSP). The BCO model is constructed algorithmically based on the collective intelligence shown in bee foraging behavior. The model is integrated with 2-opt heuristic to further improve prior solutions generated by the BCO model. Experimental results comparing the proposed BCO model with existing approaches on a set of benchmark problems were also presented.

Zakir (2010) presented a new crossover operator, Sequential Constructive crossover (SCX), for a genetic algorithm that generates high quality solutions to the traveling salesman Problem (TSP). The sequential constructive crossover operator constructs an offspring from a pair of parents using better edges on the basis of their values that may be present in the parents' structure maintaining the sequence of nodes in the parent chromosomes. The efficiency of the SCX is compared as against some existing crossover operators; namely, edge recombination crossover (ERX) and generalized N-point crossover (GNX) for some benchmark TSPLIB instances. Experimental results show that the new crossover operator is better than the ERX and GNX.

The aim of the Travelling Salesman Problem (TSP) is to find the cheapest way of visiting all elements in a given set of cities (nodes) exactly once and returning to the starting point. In solutions presented in the literature costs of travel between nodes are based on Euclidean distances, the problem is symmetric and the costs are constant and crisp values. Practical application in road transportation and supply chains are often uncertain or fuzzy. The risk attitude depends on the features of the given operation. Foldesi et al., (2010) presented a model that handles the fuzzy, time dependent nature of the TSP and also gives a solution for the asymmetric loss aversion by embedding the risk attitude into the fitness function of the eugenic bacterial memetic algorithm. Computational results are presented for different cases.

The classical TSP is investigated along with a modified instance where some costs between the cities are described with fuzzy numbers. Two different techniques are proposed to evaluate the uncertainties in the fuzzy cost values. The time dependent version of the fuzzy TSP is also investigated and simulation experiences are presented.

Iridia (1996) presented an artificial ant colony model capable of solving the traveling salesman problem (TSP). Ants of the artificial colony are able to generate successively shorter feasible tours by using information accumulated in the form of a pheromone trail deposited on the edges of the TSP graph. Computer simulations demonstrate that the artificial ant colony is capable of generating good solutions to both symmetric and asymmetric instances of the TSP. The method is an example, like simulated annealing, neural networks, and evolutionary computation, of the successful use of a natural metaphor to design an optimization algorithm.

The traveling salesman problem and the quadratic assignment problem are the two of the most commonly studied optimization problems in Operations Research because of their wide applicability. Due to their NP -hard nature, the individual problems are already complex and difficult to solve. Ping and William (2005) studied a model which integrated the two hard problems together, that is called the integrated problem of which the complexity is absolutely much higher than that of the individual ones. Not only a complete mathematical model which integrates both the traveling salesman and the quadratic assignment problems together is built, but also a genetic algorithm hybridized with several improved heuristics is developed to tackle the problem.

The Traveling Salesman Problem (TSP) is one of the most intensively studied problems in computational mathematics. To solve this problem a number of algorithms have been developed using genetic algorithms. But these algorithms are not so suitable for solving large-scale TSP. Kalyan et al., (2010) proposed a new solution for TSP using hierarchical clustering and genetic algorithm.

Time-constrained deliveries are one of the fastest growing segments of the delivery business, and yet there is surprisingly little literature that addresses time constraints in the context of stochastic customer presence. Ann and Barrett (2007) studied the probabilistic traveling salesman problem with deadlines (PTSPD). The PTSPD is an extension of the well-known Probabilistic Traveling Salesman Problem (PTSP) in which, in addition to stochastic presence, customers must also be visited before a known deadline. The authors presented two recourse models and a chance constrained model for the PTSPD. Special cases are discussed for each model, and computational experiments are used to illustrate under what conditions stochastic and deterministic models lead to different solutions.

Kenneth and Ruth (2007) proposed a new multi-period variation of the M-traveling salesman problem is introduced. The problem arises in efficient scheduling of optimal interviews among tour brokers and vendors at conventions of the tourism and travel industry. In classical traveling salesman problem vocabulary, a salesman is a tour broker at the convention and a city is a vendor's booth. In this problem, more than one salesman may be required to visit a city, but at most one salesman per time period can visit each city. The heuristic solution method presented is polynomial and is guaranteed to produce a non-conflicting set of salesmen's tours. The results of an implementation of the method for a recent convention are also reported.

Valentina et al., (2010) proposed an Equality Generalized Traveling Salesman Problem (EGTSP), which is a variant of the well-known traveling salesman problem. We are given an undirected graph G = (V, E), with set of vertices V and set of edges E, each with an associated cost. The set of vertices is partitioned into clusters. E-GTSP is to find an elementary cycle visiting exactly one vertex for each cluster and minimizing the sum of the costs of the traveled edges. The authors proposed a multi-start heuristic, which iteratively starts with a randomly chosen set of vertices and applies a decomposition approach combined with improvement procedures. The decomposition approach considers a first phase to determine the visiting order of the clusters and a second phase to find the corresponding minimum cost cycle. We show the effectiveness of the proposed approach on benchmark instances from the literature. On small instances, the heuristic always identifies the optimal solution rapidly and outperforms all known heuristics; on larger instances, the heuristic always improves, in comparable computing times, the best known solution values obtained by the genetic algorithm.

June and Sethian (2006) put forward a problem in which given a domain, a cost function which depends on position at each point in the domain, and a subset of points ("cities") in the domain. The goal is to determine the cheapest closed path that visits each city in the domain once. This can be thought of as a version of the traveling salesman problem, in which an underlying known metric determines the cost of moving through each point of the domain, but in which the actual shortest path between cities is unknown at the outset. The authors proposed algorithms for both a heuristic and an optimal solution to this problem. The complexity of the heuristic algorithm is at worst case M·N log N, where M is the number of cities, and N the size of the computational mesh used to approximate the solutions to the shortest paths problems. The average runtime of the heuristic algorithm is linear in the number of cities and O (N log N) in the size N of the mesh.

Many companies have travelling salesmen that market and sell their products. This results in much travelling by car due to the daily customer visits. This causes costs for the company, in form of travel expenses compensation, and environmental effects, in form of carbon dioxide pollution. As many companies are certified according to environmental management systems, such as ISO 14001, the environmental work becomes more and more important as the environmental consciousness increases every day for companies, authorities and public. Torstensson (2008) presented a model which computes reasonable limits on the mileage of the salesmen; these limits are based on specific conditions for each salesman's district. The objective is to implement a heuristic algorithm that optimizes the customer tours for an arbitrary chosen month, which will represent a "standard" month. The output of the algorithm, the computed distances, will constitute a mileage limit for the salesman. The algorithm consists of a constructive heuristic that builds an initial solution, which is modified if infeasible. This solution is then improved by a local search algorithm preceding a genetic algorithm, which task is to improve the tours separately. This method for computing mileage limits for travelling salesmen generates good solutions in form of realistic tours. The mileage limits could be improved if the input data were more accurate and adjusted to each district, but the suggested method does what it is supposed to do.

Davoian and Gorlatch (2005) presented a new modification of the Genetic Algorithm (GA) for solving the classical Traveling Salesman Problem (TSP), with the objective of achieving its efficient implementation on multiprocessor machines. The authors described the new features of our GA as compared to existing algorithms, and developed a new parallelization scheme, applicable to arbitrary GAs. In addition to parallel processes and iterative data exchanges between the involved populations, our parallel implementation also contains a generation of new possible solutions (strangers), which eliminates typical drawbacks of GA and extends the search area. The proposed algorithm allows accelerating the solution process and generates solutions of better quality as compared with previously developed GA versions.

Marco and Luca (1997) presented an artificial ant colony capable of solving the traveling salesman problem (TSP). Ants of the artificial colony are able to generate successively shorter feasible tours by using information accumulated in the form of a pheromone trail deposited on the edges of the TSP graph. Computer simulations demonstrate that the artificial ant colony is capable of generating good solutions to both symmetric and asymmetric instances of the TSP. The method is an example, like simulated annealing, neural networks, and evolutionary computation, of the successful use of a natural metaphor to design an optimization algorithm.

An analogy with the way ant colonies function has suggested the definition of a new computational paradigm, which we call Ant System. Marco et al., (1996) proposed it as a viable new approach to stochastic combinatorial optimization. The main characteristics of this model are positive feedback, distributed computation, and the use of a constructive greedy heuristic. Positive feedback accounts for rapid discovery of good solutions, distributed computation avoids premature convergence, and the greedy heuristic helps find acceptable solutions in the early stages of the search process. The authors applied the proposed methodology to the classical Traveling Salesman Problem (TSP), and report simulation results. The authors also discussed parameter selection and the early setups of the model, and compare it with tabu search and simulated annealing using TSP. To demonstrate the robustness of the approach, the authors showed how the Ant System (AS) can be applied to other optimization problems like the asymmetric traveling salesman.

Durbin and Willshaw (1987) studied the ant colony system (ACS), a distributed algorithm that is applied to the traveling salesman problem (TSP). In the ACS, a set of cooperating agents called ants cooperate to find good solutions to TSP's. Ants cooperate using an indirect form of communication mediated by a pheromone they deposit on the edges of the TSP graph while building solutions. The authors studied the ACS by running experiments to understand its operation. The results showed that the ACS outperforms other nature-inspired algorithms such as simulated annealing and evolutionary computation, and we conclude comparing ACS-3-opt, a version of the ACS augmented with a local search procedure, to some of the best performing algorithms for symmetric and asymmetric TSP's.

Kenneth and Ruth (1992) presented a new multi period variation of the M-traveling salesman problem. The problem arises in efficient scheduling of optimal interviews among tour brokers and vendors at conventions of the tourism and travel industry. In classical traveling salesman problem vocabulary, a salesman is a tour broker at the convention and a city is a vendor's booth. In this problem, more than one salesman may be required to visit a city, but at most one salesman per time period can visit each city. The heuristic solution method presented is polynomial and is guaranteed to produce a non conflicting set of salesmen's tours. The results of an implementation of the method for a recent convention are also reported.

June and Sethian (2006) considered a problem in which given a domain, a cost function which depends on position at each point in the domain, and a subset of points (“cities”) in the domain, the goal is to determine the cheapest closed path that visits each city in the domain once. This can be thought of as a version of the traveling salesman problem, in which an underlying known metric determines the cost of moving through each point of the domain, but in which the actual shortest path between cities is unknown at the outset.

CHAPTER THREE

METHODOLOGY

3.0 INTRODUCTION

In this chapter, we shall put forward the research methodology of the studies.

3.1 VARIOUS FORMULATIONS OF TRAVELING SALESMAN PROBLEM

The TSP is the most well known combinatorial optimization problem. The TSP can be easily stated as follows. A salesman wants to visit n distinct cities and then returns home. He wants to determine the sequence of the travel so that the overall traveling distance is minimized while visiting each city not more than once. Although the TSP is conceptually simple, it is difficult to obtain an optimal solution. In an n-city situation, any permutation of n cities yields a possible solution. As a consequence, n! possible tours must be evaluated in the search space.

By introducing variables x_{ij} to represent the tour of the salesman travels from city i to city j, one of the common integer programming formulations for the TSP can be written as:

$$\text{Minimize } z = \sum_{\substack{j=1 \\ j \neq i}}^{m} d_{ij} x_{ij}$$

Subject to

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad j = 1, 2, \ldots, m;\ i \neq j.$$

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad i = 1, 2, \ldots, m;\ i \neq j.$$

$$u_i - u_j + m x_{ij} \leq m - 1 \qquad i, j = 2, 3, \ldots, m;\ i^{1} j.$$

All $x_{ij} = 0$ or 1, All $u_i \geq 0$ and is a set of integers

The distance between city *i* and city *j* is denoted as d_{ij}. The objective function Z is simply to minimize the total distance travelled in a tour.

The first constraint set ensures that the salesman arrives once at each city. The second constraint set ensures that the salesman leaves each city once. The third constraint set is to avoid the presence of sub-tour. Generally, the TSP formulated is known as the Euclidean TSP, in which the distance matrix *d* is expected to be symmetric, that is $d_{ij}= d_{ji}$ for all i, j, and to satisfy the triangle inequality, that is $d_{ik} \leq d_{ij}+ d_{jk}$ for all distinct *i, j, k*.

Mathematical formulations of TSP and mTSP

The TSP can be defined on a complete undirected graph $G= (V,E)$ if it is symmetric or on a directed graph $G= (V,A)$ if it is asymmetric. The set V ={1, . . . , n} is the vertex set, $E=\{(i,j):i,j \in V,i<j\}$ is an edge set and $A= \{(i,j):i,j \in V,i \neq j\}$ is an arc set. A cost matrix $C= (c_{ij})$ is defined on E or on A. The cost matrix satisfies the triangle inequality whenever $c_{ij} \leq c_{ik}+c_{kj}$, for all *i, j, k*. In particular, this is the case of planar problems for which the vertices are points $P_i= (X_i, Y_i)$ in the plane, and

$c_{ij}= \sqrt{(Xi - Xj)2 + (Yi - Yj)2}$ is the Euclidean distance. The triangle inequality is also satisfied if c_{ij} is the length of a shortest path from *i* to *j* on *G*.

Integer programming formulation of sTSP

Many TSP formulations are available in literature. Recent surveys by (Orman and Williams, 2006; O¨ncan et al., 2009) can be referred to for detailed analysis. Among these, the (Dantzig et al., 1954) formulation is one of the most cited mathematical formulation for TSP. Incidentally, an early description of Concorde, which is recognized as the most performing exact algorithm currently available, was published under the title 'Implementing the Dantzig–Fulkerson–Johnson algorithm for large traveling salesman problems' (Applegate et al., 2003). This formulation associates a binary variable x_{ij} with each edge (i, j), equal to 1 if and only if the edge appears in the optimal tour. The formulation of TSP is as follows.

$$\text{Minimize} \sum_{i<j} c_{ij} x_{ij}$$

Subject to

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2 \quad (k \in V)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \qquad (S \subset V,\ 3 \leq |S| \leq |\ n- 3$$

$$x_{ij} = 0 \text{ or } 1 \qquad (i, j) \in E$$

In this formulation, the constraints are referred to as degree constraints, sub tour elimination constraints and integrality constraints, respectively. In the presence of the first constraint, the second constraints are algebraically equivalent to the connectivity constraints

$$\sum_{i \in S, j \in V \backslash S, j \in S} x_{ij} \geqslant 2 \qquad (S \subset V,\ 3 \leq |S| \leq |\ n- 3)$$

Integer programming formulation of aTSP

The (Dantzig et al., 1954) formulation extends easily to the asymmetric case. Here x_{ij} is a binary variable, associated with arc (i,j) and equal to 1 if and only if the arc appears in the optimal tour. The formulation is as follows.

$$\text{Minimize} \sum_{i \neq j} c_{ij}\, x_{ij}$$

Subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad (i \in V,\ i \neq j)$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad (j \in V,\ j \neq i)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \qquad (S \subset V,\ 2 \leq |S| \leq |\ n- 2)$$

$$x_{ij} = 0 \text{ or } 1 \qquad (i, j) \in A$$

Integer programming formulations of mTSP

Different types of integer programming formulations are proposed for the mTSP. Before presenting them, some technical definitions are as follows. The mTSP is defined on a graph *G*= (*V, A*), where V is the set of n nodes (vertices) and A is the set of arcs (edges).Let $C= (c_{ij})$ be a cost (distance) matrix associated with A. The matrix C is said to be symmetric when $c_{ij}=c_{ji}, \forall (i, j) \in A$ and asymmetric otherwise. If $c_{ij}+c_{jk} \geq c_{ik}, \forall i, j, k \in V$, C is said to satisfy the triangle inequality. Various integer programming formulations for them TSP have been proposed earlier in the literature, among which there exist assignment based formulations, a tree-based formulation and a three-index flow-based formulation. Assignment based formulations are presented below. For tree based formulation and three-index based formulations refer (Christofides et al., 1981).

Assignment-based integer programming formulations

The mTSP is usually formulated using an assignment based double-index integer linear programming formulation. A general scheme of the assignment-based directed integer linear programming formulation of the mTSP can be given as follows:

$$\text{Minimize} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, x_{ij}$$

Subject to

$$\sum_{j=2}^{n} x_{1j} = m$$

$$\sum_{j=2}^{n} x_{j1} = m$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 2, \ldots, n$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 2, \ldots, n$$

$$x_{ij} = \begin{cases} 1 & if\ arc\ (i, j) is\ used\ in\ the\ tour \\ 0 & otherwise \end{cases}, \quad x_{ij} \in \{0, 1\}, \forall (i, j) \in A$$

In this chapter of the study, we shall provide an explanation of the dynamic programming algorithm which we proposed to solve our problem, but prior to this, since TSP problems are Network related problem, a brief introduction of Graph theory as well as Euclidean distance formula which we shall use to capture the distances between all pairs of cities are introduced.

3.2 GRAPH THEORY.

DEFINITION: A graph is an ordered pair G = (V, E) consisting of a finite set and a subset E of elements of the form(x, y) where x and y are in V. The set V are called the vertices of the graph and the set E are called the edges

DEFINITION: A graph G is said to be a bipartite (or bicoloured) graph if the vertices can be partitioned into two mutually disjoint sets X and Y so that there is no edge of the form (x, x′) with x and x′ in X or of the form (y, y′)with y and y′ in Y. A bipartite graph will be denoted by G = ({X, Y}, E).

NOTATION: The cardinality of a set X will be denoted by |X|.

Bipartite graphs G = ({X, Y}, E) are represented by matrices. The X vertices are for example used for row indices and the Y vertices are used as column indices. Generally, the existence of an edge (x, y) is indicated by a 1 in the x, y cell of the |X| × |Y| matrix; no edge is indicated by 0. For the assignment problem, we are representing an edge by 0and no edge by a nonzero number.

DEFINITION: A matching for a bipartite graph G = ({X, Y}, E) is a subset M of E such that no two elements of M have a common vertex.

DEFINITION: If G = ({X, Y}, E) is a bipartite graph, set

ρ (G) = max {|M| | M is a matching of G}.

A matching M such that |M| = ρ (G) will be called a maximal matching.

DEFINITION: A set of vertices V′ is said to be a cover of a set of edges E′ if every edge in E′ is incident on one or more of the vertices of V′. A set of vertices S will be called a cover of the bipartite graph G = ({X, Y}, E) if every edge of G is incident on one or more of the vertices of S.

DEFINITION: If G = ({X, Y}, E) is a bipartite graph, set

c (G) = min {|S| | S is a cover of G}.

A cover S such that |S| = c (G) will be called a minimal cover of G.

THEOREM: If G = ({X, Y}, E) is a bipartite graph, then ρ (G) ≤ c (G).

PROOF: Let S be a cover with |S| = c (G). Let M be a matching. Then each e in M has at least one of its vertices in S. If |M| > S, then by the pigeonhole principle, two edges e_1 and e_2 meet the same vertex v in S. This contradicts the definition of a matching. So we have that |M| ≤ |S| = c (G).

3.3 Optimizing Problems

The efficient way of solving numerous programming problems implies their optimal breaking down into sub-problems. In this chapter we are dealing with such optimizing problems where the following conditions are true:

- There is a target function which has to be optimized.

- The optimizing of the target function implies to break down the problem into sub-problems.
- This involves a sequence of decisions.
- Concerning the division into sub-problems, with each decision (cut) the problem is reduced to one (I. type optimizing problems) similar, but smaller size sub-problem, or breaks into two or more (II. type optimizing problems) similar, but smaller size sub-problems.
- The target function is defined on the set of the problem's sub-problems.
- The principle of optimality is valid for the problem, according to which the optimal solution of the problem can be built from the optimal solutions of its sub-problems (the optimal value of the target function referring to the problem can be determined from the optimal values referring to the sub-problems).
- Out of the different possibilities of breaking down the problem, we consider optimal that one (or that sequence of decisions) which - in accordance with the basic principle of optimality - involves the optimal construction of the solution of the problem.
- We call a sub-problem trivial when the value of the target function referring to it is given by the input data of the problem in a trivial way.

Such an optimizing problem is solved efficiently with the so called dynamic programming technique. In this study we are going to introduce a special graph, which we have called d-graph (from division-graph),in order to provide a special tool for such an optimum problem's analysis which breaks down into two or more sub problems by every decision.

3.4. D-GRAPHS

Definition: We call the connected weighted digraph $G_d(V, E, C)$ a d-graph if the following conditions are fulfilled:

(1) $V = V_p \cup V_d$ and $E = E_p \cup E_d$

(2) V_p - the set of the p type nodes of the graph (p-nodes).

$V_p = \{p_1, p_2, \ldots, p_{nr\text{-}p}\}$, nr-p is number of p-nodes.

(3) Exactly one element of the set V_p is a source node (f).

(4) We assign the set of p type sink nodes of G_d with S (G_d) (nr-s marks the number of sink nodes).

(5) V_d - the set of the graph's d type nodes (d-nodes); nr- d is number of d-nodes.

(6) All the neighbours of the d-nodes are p type and inversely, all the neighbours of the p-nodes are of d type. Each d-node has exactly one in-neighbour of p type, which we call p-father. The out-neighbours of the p-nodes are called d-sons. Each d-node has at least one p type out-neighbour and we are going to refer to these as p-sons.

(7) The d-nodes are identified with two indexes: For example the notation d_{ik} refers to the d-son identified as the k^{th} d-sons of the p-node p_i.

(8) E_p - the set of p type arcs of the graph (p-arcs).

$E_p = \{(p_i, d_{ik}) / p_i \in V_p;, d_{ik} \in V_d\}$.

(9) E_d - the set of d type arcs of the graph (d-arcs).

$E_d = \{(d_{ik}, p_j) / d_{ik} \in V_d, p_j \in V_p; i < j\}$. We should notice that the p- type descendent of any p-node has bigger indexes. So in case of any d-graph the source is the 1 node.

(10) The C: Ep$\rightarrow$ R function associates a cost to every p-arc. We consider the d-arcs of zero cost.

Theorem: Every d-graph is a cyclic.

Proof. Let's assume that an oriented cycle exists in one of the d-graphs. According to the sixth item of the definition the p and d type nodes alternate on the cycle. Should the cycle consist of one p-node and one d-node, then the p-node is the p-father and also the p-son of node d in the same time. But this contradicts the ninth item of the definition according to which the p-sons of a d-node have always bigger indexes than its p-father. In case there are at least two nodes of both types, then let's consider p_i and p_j two consecutive p-nodes of the cycle. As p_i is the ancestor and in the same time the descendent of p_j- also according to the ninth item of the definition - *i* should be smaller and also bigger than *j*, which is obviously impossible. So every d-graph is acyclic.

Conclusion: The p-nodes of any d-graph can be arranged in topological order.

The following picture presents such a d-graph where each d-node has exactly two p-sons.

Definition: We call the d-graph $g_d(v, e, c)$ the d-sub graph of the d-graph $G_d(V, E, C)$,if

- $v_p \subseteq V_p$, $v_d \subseteq V_d$, $e_p \subseteq E_p$, $e_d \subseteq E_d$ and $S(g_d) \subseteq S(G_d)$
- $c{:}e_p \rightarrow R$ and $c(x) = C(x)$ for any $x \in e_p$
- The set of the d, respectively p type sons of any p, respectively d type node of g_d are similar in the g_d and G_d d-graphs.

It results from the above definition that every p-node of a d-graph unequivocally identifies the d-sub graph for which the respective node is its source.

Definition: We call d-tree the d-graph where every p-node (except the sinks) has exactly one son. The source of a d-tree is called d-root and its sinks are called d-leaves. The set of leaves of the T_d d-tree are marked with $L(T_d)$.

Definition: We call the d-tree t_d (vt, et, c) the d-subtree of the d-tree T_d(V t, Et, C)

if

- $vt_p \subseteq V\, t_p$, $vt_d \subseteq V\, t_d$, $et_p \subseteq Et_p$, $et_d \subseteq Et_d$ and $L(t_d) \subseteq L(T_d)$
- $c{:}et_p \rightarrow R$ and $c(x) = C(x)$ for any $x \in et_p$
- the set of p-sons of any d-node of t_d corresponds in the td and T_d d-trees.

Definition We call a d-tree T_d(V t, Et, c) the d-sub tree of the d-graph G_d(V, E, C)

if

- $V\, t_p \subseteq V_p$, $V\, t_d \subseteq V_d$, $Et_p \subseteq E_p$, $Et_d \subseteq E_d$ and $L(T_d) \subseteq S(G_d)$
- $c{:}Et_p \rightarrow R$ and $c(x) = C(x)$ for any $x \in Et_p$
- the set of p-sons of any d-node of T_d corresponds in the d-tree T_d and the d-graph G_d.

If the root of T_d corresponds to the source of G_d, then we can speak about a spanning d-subtree.

Definition: By the costs of a d-tree we mean the total costs of its p-arcs.

Definition: We call the spanning d-subtree of a d-graph with the lowest costs minimal cost spanning d-subtree.

Definition (the basic principle of optimality): We say that a d-graph has an optimal structure if every d-subtree of its optimal (having minimal costs) spanning d-subtree is itself an optimal spanning d-subtree of the d-subgraph determined by its root.

3.5 Optimal Structure d-graphs

Let G_d(V, E, C) be a d-graph. In the followings we are going to define a function C of p-arc-costs where every d-graph will be of optimal structure. Before doing that we are defining the node-weighing functions w_p and w_d. We mark the set of d-sons of the p-node p_i with d-son set (p_i) and the set of p-sons of the d-node d_{ik} with p-son set (d_{ik}).

The weight-function w_p:

w_p: $V_p \rightarrow R$

For every p_i, i= 1 *: : :* nr_p p-node corresponds $w_p(p_i)$ = optimum $\{w_d(d_{ik})\}$, if $p_i \notin S(G_d)$

$d_{ik} \in$ d-son set(p_i)

$w_p(p_i) = h_r$, if p_i is the r^{th} sink of the d-graph

where $\{h_1, h2, ..., h_{nr-s}\} \subset R$ is an input set which characterizes the G_d d-graph

The w_p weight of every p-node (except the sinks) is equal to the w_d weight of its "optimal d-son".

The weight function w_d:

w_d:$V_d \rightarrow$ R

For every d_{ik} d-node corresponds $w_d(d_{ik}) = \varphi(\{w_p(p_j) / p_j \in$ p-son set $(d_{ik})\})$

The function $'$ describes mathematically how the w_d weight of a d-node can be calculated from the w_p weights of its p-sons. The function φalso characterizes the G_d d-graph

After having introduced the above weight functions, we define the cost function

C^*in the following way:

C^*:$E_p \rightarrow$ R, $C^*((p_i, d_{ik})) = |w_p(p_i) - w_d(d_{ik})|$

Theorem: Every d-graph $G_d(V, E, C^*)$ has optimal structure.

Proof. As we have chosen the weight of the optimal d-sons as the weight of the p-nodes, every p-node is adjacent to at least one zero cost p-arc. It derives from this that the minimal cost spanning d-subtree and its every d-subtree will have zero costs. As C^*, by its definition, assigns positive costs to the p-arcs, it is natural that every d-subtree of the minimal cost spanning d-subtree will be a minimal cost spanning d-subtree of the d-subgraph which has a corresponding source of its root.

3.6 Determination of the optimal spanning d-subtree with the implementation of the basic principle of optimality

Let G_d (V, E, C^*) be an optimal structure d-graph. According to the basic principle of optimality, the optimal spanning d-subtree of any g_d d-subgraph of G_d can be determined from the optimal spanning d-subtrees of the son-d-subgraphs of g_d. Consequently we are going to determine the optimal spanning d-subtrees belonging to the nodes $p_i \in V_p$ (i = 1, ..., nr-p) in a reversed topological order. This order can be ensured if at the depth-traversing, we deal with the certain nodes at the moment we are leaving them.

We use the arrays WP[1 : : : nr-p] and WD[1 : : : nr-d] in order to store the weights of the p, respectively d type nodes of the d-graph G_d. At the beginning we fill up the elements of array WP corresponding to the sinks with their h_i ($i = 1 : : : nr\text{-}s$) weights, the other elements with the value NIL. For the storage of the optimal spanning d-subtree we take array ODS[1 : : : nr p],which stores the optimal d-sons of the p-nodes. We initialize this array with the value NIL. The initialization procedure, depending on the nature of the optimum to be calculated, gives a suitable starting value to the array-element WP [p_i] received as a parameter. The function is better analysed whether the first parameter is better than the second one, according to the nature of the optimum.

optimal division(p_i)

initialization (WP[p_i])

for all $d_{ik} \in$ d-son set (p_i) do

for all $p_j \in p$-son set (d_{ik}) do

if$WP[p_j] = NIL$ then optimal division(p_j)

endif

endfor

$WD[d_{ik}] = \varphi(\{WP[p_j] / p_j \in p\text{-son set } (d_{ik})\})$

if is better($WD[d_{ik}]$, $WP[p_i]$) then

$WP[p_i] = WD[d_{ik}]$

$ODS[p_i] = d_{ik}$

endif

endfor

end optimal division

Of course we call the optimal division procedure for the source node, presuming that it is not a sink in the same time. The OSD values of the sinks remain NIL. The following recursive procedure, based on the ODS array prints the p-arcs of the optimal spanning d-subtree in a pre-order order.

optimal tree (p_i)

write (p_i, $ODS[p_i]$)

for all $p_j \in p$-son set ($ODS[p_i]$) do

if$ODS[p_j]$ φ NIL then optimal tree (p_j)

endif

endfor

end optimal tree

3.7. d-graphs

A d-graph can be associated to any optimizing problem described in the introduction.

- The p-nodes represent the different sub-problems given by the breaking down of the problem. The source represents the original problem, the sinks the trivial ones.
- The numbering of the p-nodes and the a cyclicity given by this go hand in hand with the fact that, in the course of the breaking down, we reduce the problem to simpler and simpler sub-problems.
- A p-node will have as many d-sons as the number of possibilities in which the sub-problem represented by it can be broken down to its sub-problems, by the respective decision. These decision possibilities are represented by the p-arcs.
- The d-nodes represent the way the respective sub-problem breaks down into its sub-problems with the choices given by the different decisions.
- A d-node will have as many p-sons, as the number of sub-problems resulted after the disintegration - with the occasion of the decision represented by it - of the sub-problem described by its p-father. This breaking down into sub-problems is described by the d-arcs.
- If different sequences of decisions taken at the breaking down of a problem lead to the same sub-problem, then the respective p-node will have identical p-descendents on different descent branches.
- The d-subgraphs of a d-graph express the way in which the sub-problems represented by its sources can be broken down onto further, smaller sub-problems.
- A certain subtree of a d-graph describes one of the breaking downs onto sub-problems of the sub-problem represented by its root. The spanning sub-trees of a

d-graph represent the possibilities of breaking down the original problem onto its sub-problems.

- [2]The optimal structure of the d-graphs expresses the fact that the optimal solution of the problem is built from the optimal solutions of the sub-problems. In other words, the corresponding sub-sequences of the optimal sequence of the decisions are also optimal.
- The optimal spanning d-subtree represents the optimal breaking down of the problem into sub-problems (its every p-arc represents one of the decisions of the optimal sequence of decisions.).
- The w_p function is nothing else but the returning of the target function to be optimized to the G_d d-graph.
- h_1, h2..., h_{nr-s} real values are the optimal values referring to the trivial sub-problems of the target function, represented by the sinks.
- The nature of the optimum function is directly given by the target function of the problem and is often one of the minimum or maximum functions.
- The function $\varphi$is determined by the structure of the problem, the general rule according to which the solution of a sub-problem is built from the solutions of its sub-problems.

Hereby, an optimizing problem can be regarded as the determination of the weight of the source of a d-graph (the optimal value of the target function concerning the original problem) and of its optimal spanning d-subtree (optimal sequence of decisions, respectively optimal breaking down into sub-problems).We call the procedure optimal division, which implements the basic principle of optimality, dynamic programming.

3.8 DYNAMIC PROGRAMMING

Dynamic programming is an optimization approach that transforms a complex problem into a sequence of simpler problems; its essential characteristic is the multistage nature of the optimization procedure. More so than other optimization techniques, dynamic programming provides a general framework for analyzing many problem types. Within this framework a variety of optimization techniques can be employed to solve particular aspects of a more general formulation. Usually creativity is required before we can recognize that a particular problem can be cast effectively as a dynamic program; and often subtle insights are necessary to restructure the formulation so that it can be solved effectively.

3.8.1 CHARACTERISTICS OF DYNAMIC PROGRAMMING PROBLEMS

One way to recognize a situation that can be formulated as a dynamic programming problem is to notice it's basic features.

These basic features that characterize dynamic programming problems are presented and discussed below:

i. The problem can be divided into stages, with a policy decision required at each stage. Dynamic programming problems require making a sequence of interrelated decisions, where each decision corresponds to one stage of the problem.

ii. Each stage has a number of states associated with the beginning of that stage.

In general, the states are the various possible conditions in which the system might be at that stage of the problem. The number of states may be either finite or infinite.

iii. The effect of the policy decision at each stage is to transform the current state to *a* state associated with the beginning of the next stage (possibly according to a probability distribution).

This procedure suggests that dynamic programming problems can be interpreted in terms of the networks. Each node would correspond to a state. The network would consist of columns of nodes, with each column corresponding to a stage, so that the flow from a node can go only to a node in the next column to the right. The links from a node to nodes in the next column correspond to the possible policy decisions on which state to go to next. The value assigned to each link usually can be interpreted as the immediate contribution to the objective function from making that policy decision. In most cases, the objective corresponds to finding either the shortest or the longest path through the network.

iv. The solution procedure is designed to find an optimal policy for the overall problem, i.e., a prescription of the optimal policy decision at each stage for each of the possible states.

For any problem, dynamic programming provides this kind of policy prescription of what to do under every possible circumstance (which is why the actual decision made upon reaching a particular state at a given stage is referred to as a policy decision). Providing this additional information beyond simply specifying an optimal solution (optimal sequence of decisions) can be helpful in a variety of ways, including sensitivity analysis.

v. Given the current state, an optimal policy for the remaining stages is independent of the policy decisions adopted in previous stages. Therefore, the optimal immediate decision depends on only the current state and not on how you got there. This is the principle of optimality for dynamic programming.

For dynamic programming problems in general, knowledge of the current state of the system conveys all the information about its previous behavior necessary for determining the optimal policy henceforth. Any problem lacking this property cannot be formulated as a dynamic programming problem.

vi. The solution procedure begins by finding the optimal policy for the last stage.

The optimal policy for the last stage prescribes the optimal policy decision for each of the possible states at that stage. The solution of this one-stage problem is usually trivial, as it was for the stagecoach problem.

vii. A recursive relationship that identifies the optimal policy for stage n, given the optimal policy for stage (n+ 1), is available.

Therefore, finding the optimal policy decision when you start in state *s* at stage *n* requires finding the minimizing value of x_n.

This property is emphasized in the next (and final) characteristic of dynamic programming.

viii. When we use this recursive relationship, the solution procedure starts at the end and moves backward stage by stage - each time finding the optimal policy for that stage - until it finds the optimal policy starting at the initial stage. This optimal policy immediately yields an optimal solution for the entire problem.

3.8.2 The Algorithm

• Identify the decision variables and specify objective function to be optimized under certain limitations, if any.

• Decompose or divide the given problem into a number of smaller sub-problems or stages. Identify the state variables at each stage and write down the transformation function as a function of the state variable and decision variables at the next stage.

• Write down the general recursive relationship for computing the optimal policy. Decide whether forward or backward method is to follow to solve the problem.

• Construct appropriate stage to show the required values of the return function at each stage.

• Determine the overall optimal policy or decisions and its value at each stage. There may be more than one such optimal policy.

The basic features, which characterize the dynamic programming problem, are as follows:

(*i*) Problem can be sub-divided into stages with a policy decision required at each stage. A stage is a device to sequence the decisions. That is, it decomposes a problem into sub-problems such that an optimal solution to the problem can be obtained from the optimal solution to the sub-problem.

(*ii*) Every stage consists of a number of states associated with it. The states are the different possible conditions in which the system may find itself at that stage of the problem.

(*iii*) Decision at each stage converts the current stage into state associated with the next stage.

(*iv*) The state of the system at a stage is described by a set of variables, called state variables.

(*v*) When the current state is known, an optimal policy for the remaining stages is independent of the policy of the previous ones.

(*vi*) To identify the optimum policy for each state of the system, a recursive equation is formulated with 'n' stages remaining, given the optimal policy for each stage with $(n - 1)$ stages left.

(*vii*) Using recursive equation approach each time the solution procedure moves backward, stage by stage for obtaining the optimum policy of each stage for that particular stage, still it attains the optimum policy beginning at the initial stage.

CHAPTER FOUR

DATA COLLECTION AND ANALYSIS

4.0 INTRODUCTION

In this chapter, we shall consider a computational study of the Travelling Salesman Problem (TSP). Emphasis will be placed on TSP, which was worked out as a network problem. Data from the Inspectorate Division-GES, Accra shall be examined.

4.1 Data Collection and Analysis

The Inspectorate Division of the Ghana Education Service (GES), Accra usually sends inspection officers to inspect the overall academic work of various schools in the region to ensure that teachers and circuit supervisors perform their tasks as desired.

An officer moves from the first school, Kinbu Sec/ Technical School and is expected to visit as many schools as possible on each route within each journey in a day.

Table 4.1 shows the distance matrix table, taken from Transport Department of GES-Accra, showing the various links of connecting selected schools assigned to an officer to inspect in kilometers (km).

Table 4.1 Distance matrix table connecting schools in km

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	∞	16	10	32	∞	∞	∞	∞	∞	∞
2	16	∞	∞	∞	12	∞	∞	∞	∞	∞
3	10	∞	∞	∞	15	24	∞	∞	∞	∞
4	32	∞	∞	∞	∞	∞	22	∞	∞	∞
5	∞	12	15	∞	∞	28	∞	21	∞	∞
6	∞	∞	24	∞	28	∞	40	24	56	∞
7	∞	∞	∞	22	∞	40	∞	111	76	∞
8	∞	∞	∞	∞	21	24	111	∞	127	42
9	∞	∞	∞	∞	∞	56	76	127	∞	97
10	∞	∞	∞	∞	∞	∞	∞	42	97	∞

The infinity symbol signifies no direct link between the two schools.

Table 4.2 shows the names of the schools assigned to an officer.

1	Kinbu Sec/ Technical School
2	Accra Academy Senior High School
3	Labone Senior High School
4	Tema Senior High School
5	Odorgonno Senior High School
6	West Africa Senior High School
7	Prampram Senior High School
8	Amasaman Senior High/Technical School
9	Ada Senior High School
10	Ghanata Senior High School

The problem at hand is to find the minimum distance that an officer could cover and visit a maximum number of schools in a day.

Modeling the above problem as a Network problem, we obtain Figure 4.1, which shows the route map of the various ways of reaching the schools, with each node representing a school. The numbers on the lines indicate the distances in kilometres (km).

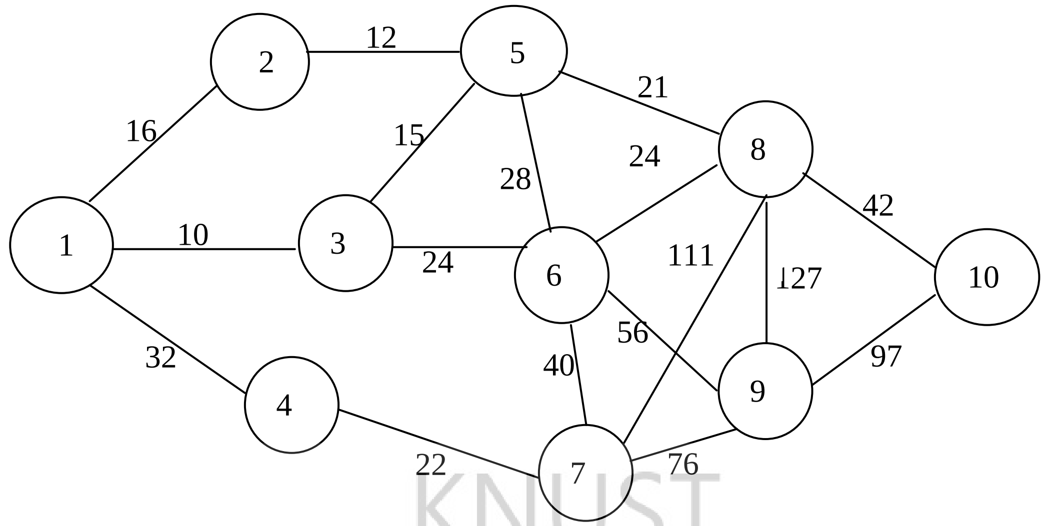

Figure 4.1: Route map of the various ways of reaching the schools

By applying dynamic programming, the problem may be considered as 4-stage problem. This is shown in Figure 4.2.

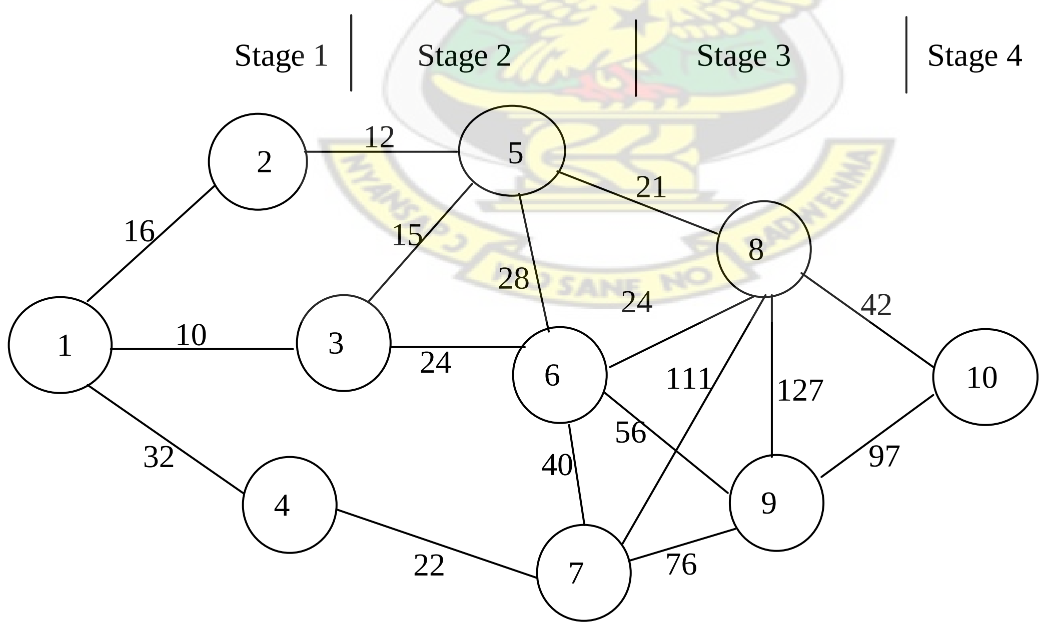

Figure 4.2: Route maps of the various ways of reaching the schools in Stages

Let x_i be the state variable in the i^{th} stage, and $d_i(x_i)$ be the distance covered in the i^{th} stage. Our model then becomes;

$d_{(i)} = \min \{\sum_{k=i-1}^{i} d_k(x_k) \}$

In the first stage, i = 1, and the officer moves from school 1(node number 1) and can reach schools 2, 3, and 4 directly.

We shall therefore have $d_{(1)} = \min \{\sum_{k=0}^{1} d_k(x_k)\}$.

Considering the distances: 1 to 2, which is 16 km, 1 to 3, which is 10 km, and 1 to 4, which is 32 km. Since this is a minimization problem and the goal of an officer is to visit more number of schools and travel less distance, we can show the routes, which give the minimum distance in bolded line and the rest of the lines we can neglect or we can show in normal lines. In this problem, lines 1 – 3 will be in bolded line and the rest in normal line as shown in Figure 4.3.

$d_{(1)} = \min \{16, 10, 32\}$, which is 10 km. The distance covered up to that stage is written just above the node.

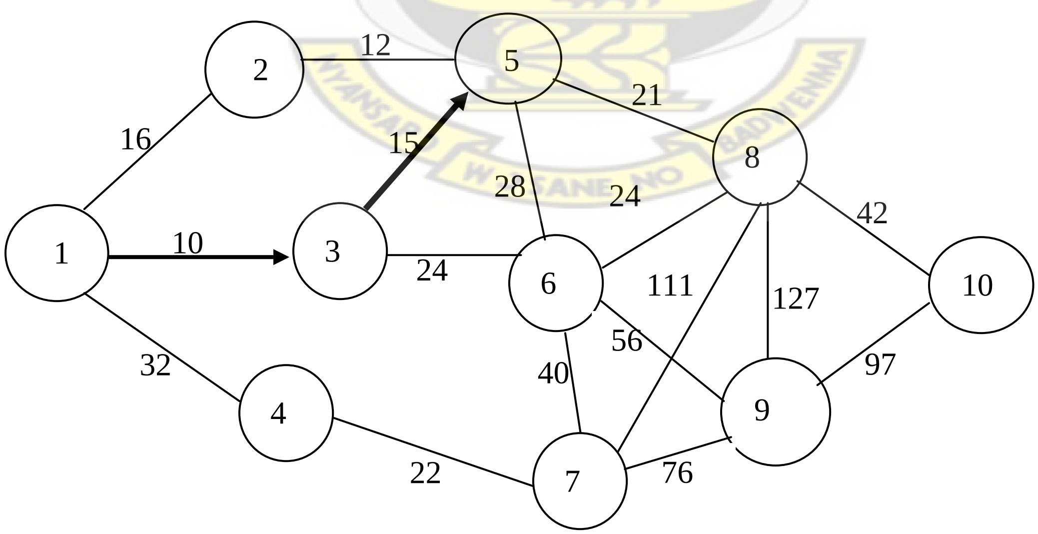

Figure 4.3: Distance of reaching the various schools

In the second stage, i = 2 and the inspection officer can reach school 6 directly from schools 2 and 3, school 6 directly from schools 2 and 4 and school 7 from 4.

We have $d_{(2)}$= min $\{\sum_{k=1}^{2} d_k(x_k)\}$.

Considering the distance from school 2 to 5, we have 16 + 12 = 28,

Similarly, from school 3 to 5 is 10 + 15 = 25 km.

The distance from school 3 to 6 = 10 + 24 = 34 km.

The distance from school 4 to 7 = 32 + 22 = 54 km.

The distance from school 5 to 6 is 25 + 28 = 53 km.

The distance from school 6 to 7 is 34 + 40 = 74 km.

$d_{(2)}$ = min{28, 25, 34, 54, 53, 74}.

The minimum distance is 25 km. Hence, the inspection officer will travel from school 1 to 3 and from school 3 to 5 covering 25 Km on the route 1-3-5.

We then move on to the next stage, which is stage 3, with i = 3.

We have $d_{(3)}$= min $\{\sum_{k=2}^{3} d_k(x_k)\}$.

In the third stage, the inspection officer may be at school 5 or at school 6 or at school 7. From there, the officer can directly go to school 8 or school 9.

Working out the minimum distance from school 5, 6 and 7 to 8 and 9, we have:

The distance from school 5 to 8 = 25 + 21 = 46 km.

The distance from school 6 to 8 = 34 + 24 = 58 km.

The distance from school 7 to 8 = 54 + 111 = 165 km.

The distance from school 6 to 9 = 34 + 56 = 90 km.

The distance from school 7 to 9 = 54 + 76 = 130 km.

The distance from school 8 to 9 = 46 + 127 = 173 km.

$d_{(3)}$ = min {46, 58, 165, 90, 130, 173}.

The minimum of all these is 46 km. Hence, the inspection officer can go from school 5 to 8 at the distance of 46 km only on the routes 1-3-5-8.

Next, we consider our final stage which is stage 4, thus i = 4.

Thus, we have $d_{(4)}$= min $\{\sum_{k=3}^{4} d_k(x_k)\}$.

In the 4th stage the inspection officer can reach school 10 from schools 8 or 9. Calculating the minimum distances from schools 8 and 9 to school 10 we have:

The distance from school 8 to 10 =46 + 42 = 88 km.

The distance from school 9 to 10 = 90 + 97 = 187 km.

$d_{(4)}$ = min {88, 187}.

The minimum of these is 88 km. This is shown in Figure 4.4 with bold lines and the distances written on top of the nodes.

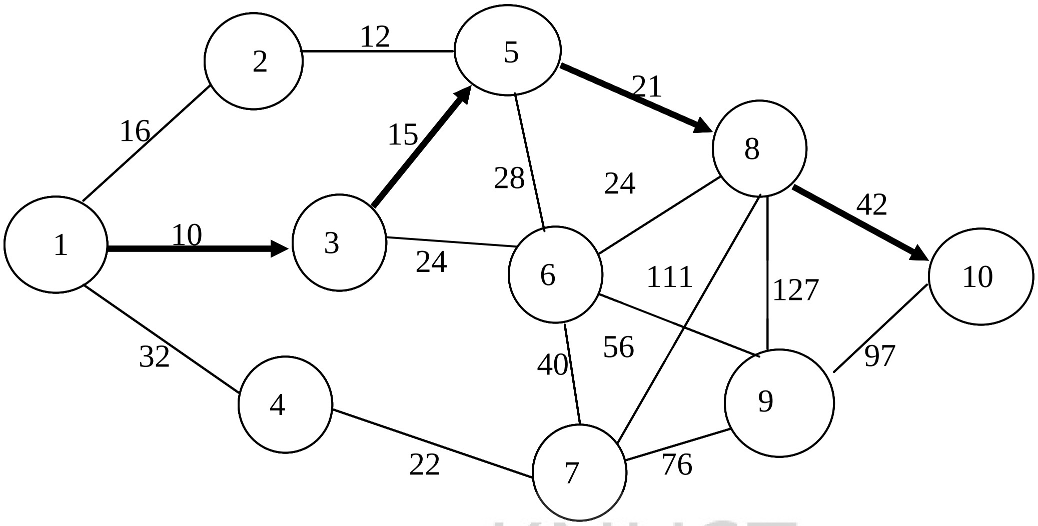

Figure 4.4: Distance of reaching schools

Hence the minimum distance from school 1 to 10 on the path is 88 km, and the routes are 1 – 3 – 5 – 8 – 10**.**

This implies that the inspection officer can use any of the above routes and visit as many as five schools on the route.

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

5.0 INTRODUCTION

The Traveling Salesman Problem is a traditional problem that has to do with making the most efficient use of resources while at the same time spending the least amount of energy in that utilization. The designation for this type of problem hails back to the days of the travelling salesman, who often wished to arrange travel distances in a manner that allowed for visiting the most towns without having to double back and cross into any given town more than once.

In a wider sense, the travelling salesman problem is considered to be a classic example of what is known as a tour problem. Essentially, any type of tour problem involves making a series of stops along a designated route and making a return journey without ever making a second visit to any previous stop. Generally, a tour problem is present when there is concern on making the most of available resources such as time and mode of travel to accomplish the most in results. Finding a solution to a tour problem is sometimes referred to as discovering the least-cost path, implying that the strategic planning of the route will ensure maximum benefit with minimum expenditure incurred.

The concept of the travelling salesman problem can be translated into a number of different disciplines. For example, the idea of combinatorial optimization has a direct relationship to the travelling salesman model. As a form of optimization that is useful in both mathematical and computer science disciplines, combinatorial optimization seeks to team relevant factors and apply them in a manner that will yield the best results with repeated usage.

In a similar manner, discrete optimization attempts to accomplish the same goal, although the term is sometimes employed to refer to tasks or operations that occur on a one-time basis rather than recurring. Discrete optimization also is helpful in computer science and mathematical disciplines. In addition, discrete optimization has a direct relationship to computational complexity theory and is understood to be of use in the development of artificial intelligence.

While the imagery associated with a travelling salesman problem may seem an oversimplification of these types of detailed options for optimization, the idea behind the imagery helps to explain a basic fundamental to any type of optimization that strives for efficiency. The travelling salesman problem that is solved will yield huge benefits in the way of maximum return for minimum investment of resources.

TSP is a very attractive problem for the research community because it arises as a natural sub-problem in many applications concerning the everyday life. Indeed, each application, in which an optimal ordering of a number of items has to be chosen in a way that the total cost of a solution is determined by adding up the costs arising from two successively items, can be modelled as a TSP instance. Thus, studying TSP can never be considered as an abstract research with no real importance.

5.1 CONCLUSIONS

This thesis seeks to model a real-life problem of Inspectorate Division of Ghana Education Service (GES), Accra as a network problem and apply dynamic programming approach in solving the problem. It was observed that the route that gave minimum achievable inspection plan was:

1 – 3 – 5 – 8 – 10 at the minimum distance of 88 km, by visiting as many as five schools on the route.

It is important to state that, past records from the Inspectorate Division show that there is no set down procedure for determining which routes to be used by inspection officers. The routes are chosen using guess work and by the discretion of the driver who is sending the officer. The maximum number of schools they normally visit were three on a route from past experience.

5.2 RECOMMENDATIONS

The use of mathematical models has proved to be efficient in the computation of optimum results and gives a systematic and transparent solution as compared with an arbitrary method. Operation has become one of the key competitive advantages with optimization-based approaches being expected to play an important role. Using optimization-based approaches to model industrial problem gives a better result. Management will benefit from the proposed approach for officers who would be assigned to inspect various schools in order to visit more school on a route at a minimized distance. We therefore recommend that our TSP model should be adopted by the Inspectorate Division of Ghana Education Service (GES), Accra for its routine inspection planning.

REFERENCE

1. Alexander I Barvinok (2010). Two Algorithmic Results for the Traveling Salesman Problem. Journal of the institute for operations research and management science, Vol. 51, pp.141-202.

2. Angel R D, Caudle W L, Noonan R and Whinston A (1972). Computer assisted school bus scheduling. Management Science, Vol. 18, pp.279–88.

3. Ann M Campbell and Barrett W Thomas (2007). Probabilistic Traveling Salesman Problem with Deadlines. Journal of the institute for operations research and management science.

4. Applegate D L, Bixby R E, Chv´atal V, and Cook W J (2003). Implementing the Dantzig–Fulkerson–Johnson algorithm for large scale traveling salesman problems. Math Program Ser B Vol. 97, pp. 91–153.

5. Applegate D L, Bixby R E, Chvátal V and Cook W J (2006). The Traveling Salesman Problem: A Computational Study, Princeton University Press, ISBN 978-0-691-12993-8.

6. Applegate D L, Bixby R E, Chv´atal V, Cook W J, Espinoza D G, Goycoolea M and Helsgaun K (2009). Certification of an optimal TSP tour through 85900 Cities. Operations Research Letter., Vol. 37, No. 1, pp. 11–15.

7. Applegate D, Cook W, and Rohe A (2000). Chained Lin-Kernighan for large traveling salesman problems. Journal of Operations Reearch, *Vol*. 37, No. 1, pp. 111–115.

8. Bernd Freisleben and Peter Merz (1996). New Genetic Local Search Operators for the Traveling Salesman Problem, Vol. 16, pp.179–88.

9. BektasTolga (2006). The multiple traveling salesman problems: an overview of formulations and solution procedures. www.elsevier.com/locate/omega

10. Bland et al.,(1989). An effective heuristic for the m-tour Traveling Salesman Problem with some side conditions. Operational Research, Vol. 25, issue 3, pp. 517-24

11. Černý V (1985).Thermo-dynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal of Optimization Theory and Applications Volume: 45, Issue: 1, Springer Netherlands, Pages: 41-51

12. Christos H Papadimitriou and Mihalis Yannakakis (2010). The Traveling Salesman Problem with Distances, One and Two. Journal of the institute for operations research and management science, Vol. 18, pp.280–99

13. Cliff Stein and David P. Wagner (2000). Approximation Algorithms for the Minimum Bends Traveling Salesman Problem. Dartmouth College Computer Science Technical Report TR2000-367, Vol. 14, pp.209–22

14. Chiung Moon, Jongsoo Kim, Gyunghyun Choi and YoonhoSeo (2000).An efficient genetic algorithm for the traveling salesman problem with precedence constraints. European Journal of Operational Research 140 (2002) 606–617.

15. Chiung Moon (2002). An efficient genetic algorithm for the traveling salesman problem with precedence constraints. European Journal of Operational Research. Volume: 140, Issue: 3, Elsevier, Pages: 606-617

16. Croes G A (2005). Method for Solving Traveling-Salesman Problems. Operations Research 2005 53:982-995. INFORMS Journal on Computing 17:111-122

17. Davoian K and Gorlatch S (2005). Modified Genetic Algorithm for the Traveling Salesman Problem and Its Parallelization. Proceeding (453) Artificial Intelligence and Applications. Maria de Press, ISBN 988-0-691-12393-9.

18. Dominique Feillet, Pierre Dejax and Michel Gendreau (2001).Traveling Salesman Problems with Profits. Journal of the institute for operations research and management science. Volume: 138, Issue: 2, Elsevier, Pages: 506-517

19. Dominique Feillet, Pierre Dejax and Michel Gendreau (2003). Traveling Salesman Problems with Profits. Journal of the institute for operations research and management science. Volume: 182, Issue: 3, Elsevier, Pages: 606-617

20. Donald L Miller and Joseph F Pekny (1991). Exact Solution of Large Asymmetric Travelling Salesman Problems. European Journal of Operational Research 140 (1991) 606–617.

21. Dorigo, M. and Gambardella, L.M. (1996). Ant Colonies for the Traveling Salesman Problem. University Libre de Bruxelles, Belgium. : Springer; 1996. p. 291–92.

22. Dreissig W and Uebaeh W (1990). Personal communication. Linear programming with pattern constraints. PhD thesis, Department of Economics, Harvard University, Cambridge, MA. Vol. 16, No. 6, pp.B373–83

23. Fagerholt and Christiansen(2000). ATraveling Salesman Problem with allocation, time window and precedence constraints-an application to ship scheduling. International Transactions in Operational Research(2000). Vol. 7, Issue: 3. Pp 231-244

24. Flip Phillips and Oliver Layton (2009). The traveling salesman problem in the natural environment. Journal of the institute for operations research and management science. *Vol*. 40, No. 1, pp. 111–125.

25. Foldesi Peter, Janos Botzheim and Laszlo T. Koczy (2010). Eugenic Bacterial Memetic Algorithm for Fuzzy Road Transport Traveling Salesman Problem. International Journal of Innovative Computing, Information and Control, Volume 7. pp 284-99

26. Glover Fand Punnen A P (1996). The travelling salesman problem: new solvable cases and linkages with the development of approximation algorithms. Journal of the Operational Research Society. Vol. 21, No. 5, pp. 99-105

27. Gorenstein, S. (1970). Printing press scheduling for multi-edition periodicals.Management Science, Vol. 16, No. 6, pp.B373–83.

28. Grötschel M and Padberg M W (1985). Polyhedral theory. The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization. Wiley: Chichester, pp 251–305.

29. Gromicho, J.; Paixão, J. & Branco, I. (1992). Exact solution of multiple traveling salesman problems. In: Mustafa Akgül, et al., editors. Combinatorial optimization. NATO ASI Series, Vol. F82. Berlin: Springer; 1992. p. 291–92.

30. Grötschel M and Pulleyblank W R (1986). Clique tree inequalities and the symmetric Travelling Salesman Problem. Mathematics of Operations Research, Vol. 11, No. 4, (November, 1986), pp. 537–569.

31. Grötschel M and Holland O (1991). Solution of Large-scale Symmetric Traveling Salesman Problems. Mathematical Programming, Vol. 51, pp.141-202.

32. Grötschel M, Jünger M and Reinelt G (1991). Optimal Control of Plotting and Drilling Machines: A Case Study. Mathematical Methods of Operations Research, Vol. 35, No. 1, (January, 1991), pp.61-84.

33. Hains Douglas R M S (2010). Generalized partition crossover for the Traveling Salesman Problem. Masters Degree Thesis Colorado State University, pages; 1492391

34. Harlan Crowder and Manfred W. Padberg (1980). Solving Large-Scale Symmetric Travelling Salesman Problems to Optimality. Journal of the institute for operations research and management science. European Journal of Operational Research 136 (1980) 602–617.

35. Ivan Gerace and Federico Greco (2008). The Travelling Salesman Problem in symmetric circulant matrices with two stripes. Mathematical Structures in Computer Science, 18 pp 165-175

36. Johan Torstensson (2008). Computation of Mileage Limits for Traveling Salesmen by Means of Optimization Techniques. Master Degree Thesis, University of Linköpingsuniversitet/Matematiska institutionen. Pp 66-85

37. John P Norback and Robert F. Love (2010). Geometric Approaches to Solving the Traveling Salesman Problem. Journal of the institute for operations research and management science. European Journal of Operational Research Vol.200 (2010) pp 156-77.

38. June Andrews and Sethian J A (2006). Fast marching methods for the continuous traveling salesman problem. http://www.pnas.org/content/104/4, 24/01/2013; 9:34 am

39. Kalyan S N S Bharadwaj. B, Krishna Kishore. G, Srinivasa RaoV (2010). Solving Traveling Salesman Problem Using Hierarchical Clustering and Genetic Algorithm. International Journal of Computer Science and Information Technologies, Vol. 2 (3), 1096-1098

40. Kaur D and Murugappan M M (2008). Performance enhancement in solving Traveling Salesman Problem using hybrid genetic algorithm. Fuzzy Information Processing Society Annual Meeting of the North American. Whales Packs, No. 3, pp.405-55

41. Kenneth C. Gilbert and Ruth B. Hofstra (2007). A New Multi-period Multiple Traveling Salesman Problem with Heuristic and Application to a Scheduling Problem. Journal of the institute for operations research and management science. Vol. 155, No. 4, pp.607-24.

42. Lenstra J K, Lawler E L, RinnooyKan A H G and Shmoys D B (1974). The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization. Wiley: Chichester, pp 251–305.

43. Lenstra, J K and Rinnooy A H G Kan (1974). Some Simple Applications of the Travelling Salesman Problem. BW 38/74, Stichting Mathematisch Centrum, Amsterdam. Roxenberg Press, ISBN 188-0-291-12393-4

44. Lenstra J K and RinnooyKanA H G (1975). Some simple applications of the Traveling salesman problem. Operational Research Quarterly, Vol. 26, pp. 717–33.

45. Lin S and Kernighan B (1973). An effective heuristic algorithm for the traveling salesman problem. Operations Research, Vol. 21, pp. 498–516.

46. Li-Pei Wong, Malcolm Yoke Hean Low and Chin Soon Chong (2001). Bee Colony Optimization with Local Search for Traveling Salesman Problem. Master's thesis, School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798.

47. Marco Dorigo and Luca Maria Gambardella (1997). Ant colonies for the traveling salesman problem.

http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/tsplib95. , 11/2/2012; 4:12pm

48. Martikainen J and S.J. Ovaska (2005). Using Fuzzy Evolutionary Programming to Solve Traveling Salesman Problems. Proceeding (481) Artificial Intelligence and Soft Computing. Hiley: Chichester, pp 331–376

49. Milton E. Harvey, Ralph T. Hocking and J. Randall Brown (2010). The Chromatic Traveling Salesmen Problem and Its Application to Planning and Structuring Geographic Space
http://onlinelibrary.wiley.com/doi/10.1111/j.1538-4632.1974.tb01015, 26/02/2013; 12:13 pm

50. Muhammad Umair Khan (2010). Use Multilevel Graph Partitioning Scheme to solve traveling salesman problem. Thesis, University of HögskolanDalarna / AkademinIndustriochsamhälle. Pp 125-89

51. Murano Takenori and Matsumoto Naoki (2003). Comparison of Approximate Methods for Traveling Salesman Problem. Technical Report, Institute of Electronics, Information and Communication Engineers Vol**.**102 N**o.** 625 93-106), pages1-6

52. PerMattsson (2010). The Asymmetric Traveling Salesman Problem. Master Degree Thesis, University of Uppsala universitet/Matematiskainstitutionen

53. Plante et al., 1987. Traveling Salesman Problem.Theory and Applications. Janeza Trdne 9, 51000 Rijeka, Craefa, pp.156-204

54. Ratliff H D and Rosenthal A S (1983). Order-Picking in a Rectangular Warehouse: A Solvable Case for the Travelling Salesman Problem. Operations Research, Vol. 31, pp. 507-521.

55. Russell R A (1997). An effective heuristic for the *m*-tour traveling salesman problem with some side conditions. Operations Research, Vol. 25, No. 3, pp.517-24.

56. Ruiz et al., (2004). Solving the vehicle routing problem by hybrid meta-heuristic algorithm. Journal of Industrial Engineering International. Vol:13, Issue:3, pp 88-104

57. Ryan J L, Bailey T G, Moore J T and Carlton W B (1998). Reactive Tabu search in unmanned aerial reconnaissance simulations. Proceedings of the 1998 winter simulation conference, Vol. 1, pp .873–9.

58. Saleh and Chelouah(2004). The Multiple traveling Salesman Problem: an overview of formulations and solution procedures. International Journal of management science, Vol. 85, issue. 5,pp 209-219

59. Schmitt, L.J. and Amini, M.N. (1998). Performance Characteristics of alternative genetic algorithm approaches to the Traveling Salesman problem using path representation: An empirical study. In European Journal of Operational Research 108:551-570

60. Susan N Twohig and Samuel O Aletan (1990). The travelling Salesman problem. Proceedings of the ACM annual conference. Vol. 2, pp .173–9

61. Svestka J A and Huckfeldt V E (1973). Computational experience with an m-salesman traveling salesman algorithm. Management Science, Vol. 19, No. 7, pp. 790–9.

62. Tobias and Ostein(2007). An Optical solution for Traveling Salesman Problem. Optics Express(2007). Vol. 15, issue:16, 10473-10482.

63. Toth P and Vigo D (2002). Branch-and-bound algorithms for the capacitated VRP.SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, pp.29–51.

64. Valentina Cacchiani, Albert Einstein Fernandes Muritiba, Marcos Negreiros, and Paolo Toth (2010). A multi start heuristic for the equality generalized traveling salesman problem. Wiley Periodicals, Inc. NETWORKS. Vol**.**11 N**o.** 625 223-106, pages12-19

65. Vieira F, Neto A and Costa J (2002).An efficient approach of the SOM algorithm to the traveling salesman problem. Proceedings 7th Brazilian Symposium on Neural Networks, IEEE Computer Society, Pages: 152-173.

66. YindaFeng (2010). Ant colony for TSP. Master Degree Thesis, University of HögskolanDalarna / AkademinIndustriochsamhälle

67. Yu Z, Jinhai L, Guochang G, Rubo Z and Haiyan Y (2002).An implementation of evolutionary computation for path planning of cooperative mobile robots.
Proceedings of the fourth world congress on intelligent control and automation, Vol. 3, pp. 1798–802.

68. Zakir H. Ahmed (2010). Genetic Algorithm for the Traveling Salesman Problem using Sequential Constructive Crossover Operator.International Journal of Biometrics and Bioinformatics Volume (3): Issue (6), pp.165-77