KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY, KUMASI

INSTITUTE OF DISTANCE LEARNING (IDL)

OPTIMAL ROUTE FOR SELECTED TOURIST SITES IN THE KWAHU ZONE OF EASTERN REGION OF GHANA

BY

TAMIMU HALIDU

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 M.SC INDUSTRIAL MATHEMATICS

MARCH, 2014

DECLARATION

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

Tamimu Halidu	....	....
Student	Signature	Date

Certified by:

Prof. S. K. Amponsah	....	....
Supervisor	Signature	Date

Certified by:

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

Certified by:

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

DEDICATION

To The Glory of God

This work is dedicated to my wife, Janet Ampadu and my daughter Pharidha Tamimu.

ABSTRACT

Tourism is becoming more than an economic venture with increasing travels for leisure. Various interventionist approaches have been adopted to improve the tourism sector in Ghana including constructing roads and leisure centres. In this thesis, a mathematical model for finding an optimal route for selected tourist sites in the Kwahu zone of Ghana. The study found out the possible implications of the optimal route on the cost of embarking on leisure travels in the Kwahu zone of Ghana. The distances between the tourist sites were thus determined. The grid points of the sites were also taken and recorded. Genetic Algorithm was applied to the data set and the study revealed that the total minimum distance for the selected tourist sites is One Hundred and Forty-Seven and two – tenth (147.2) kilometres. The study also showed that with this optimal route, the total cost of touring the sites would be significantly reduced. This consequently will make resources available to the tourists.

ACKNOWLEDGMENT

I am most grateful to the Almighty God for guiding and sustaining me throughout this course.

His everlasting love and Grace has made me see the light of day and subsequent completion of this work. My profound gratitude to my supervisor, Professor S. K. Amponsah, whose guidance has enabled me to produce this work.

Special thanks to my parents and my siblings, my friends Bakira Tahiru and Dennis Kwame Agyapong, for their prayers and support throughout my education. I owe this masterpiece to Bamfo Ernest Frempong, Justice Kwame Appati, Daniel Gyamfi, Baffoe Eric and Gabriel Obed Fosu.

While I share the credit of this Masters' thesis with all the above mentioned people, responsibility for any errors, shortcomings or omissions in this thesis is solely mine.

Contents

LIST OF ABBREVIATION

BBs	Building Blocks
DC	Direct Current
EA	Evolutionary Algorithms
EFM	Event – Freeing Mutation
EOL	End – Of – Life
FMSs	Flexible Manufacturing Systems
GA	Genetic Algorithm
GDP	Gross Domestic Product
GP	Genetic Programming
GPS	Global Positioning System
IID	Integrated Inventory Distribution Problem
JSSP	Job Shop Scheduling Problems
LISP	Locator/Identifier Separation Protocol
LTL	Less – than – Truckload
MATLAB	Mathematics Laboratory
MEXCLP	Non-Linear Maximum Expected Covering Location Problem
MLUFLP	Mult-Level Incapacitated Facility Location Problem
MOBRGRA	Muti-Objective Bisexual Reproduction Genetic Algorithms

NASA .... National Aeronautics and Space Administration

NEH .... Nawaz, Enscore and Ham

NFL .... No-Free Lunch

NW .... FSSP – No – Wait Flowshop Problem

P .... Crossover Probability

PDPTW .... Pick Up and Delivery Problem With Time Windows

Pm .... Mutation Probability

PMX .... Partiality – Matched Crossover

PPX .... Precedence Preservative Crossover

RCMPSP .... Resource Constrained Mult-Project Vehicle Routing Problem With Time Windows

SA .... Simulated Annealing

SDVRP .... Split Delivery Vehicle Routing Problem

SEFM .... Stochastic Violation – Directed Mutation

SRC .... Simple Random Crossover

TSP .... Traveling Salesman Problem

UNWTO .... United Nations World Tourism

UTM .... Universal Transverse

URLs .... Uniform Resource Locator

VDM .... Violation – Directed Mutation

VRPTW .... Vehicle Routing Problem With Time Windows

WHO ....World Health Organization

WGS .... World Grid System

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

An overall picture of developing countries shows that they are predominantly agricultural. This is particularly certain if the measurement of the significance of a sector of an economy is based upon the proportion of the population employed in or dependent upon it for their livelihood. The dependence of approximately seventy percent (70%) of the people of West Africa on agriculture for their livelihoods indicates its importance to the countries' economies. Unfortunately, agricultural productivity tends to be low in Africa countries, for a number of reasons, though various measures can be introduced to increase productivity.

The few countries which are well endowed with natural resources, and have considerable economic potential, require tremendous capital investment to exploit the resources owing to their inaccessibility. Admittedly, tourism is developed because it promises to generate revenue, employment, enhance community infrastructure and assist in revitalizing the flagging economies of both urban and rural areas. The industry itself could set up training institutes, and engage in lobbying the government if the interests of the industry were threatened by foreign competition or a tax on imported raw materials for aesthetic reasons.

It is necessary to distinguish between the situations when a particular tourist centre is located in one area, and when one area attracts a lot of tourists from a wide variety of enterprises than others. Industrial concentration causes a different set of problems to tourism localization. An important constraint on tourism industry in most developing countries especially Ghana is due to a lack of capital and transportation to the various tourist centers. Tourism is one of the fastest growing and attractive sectors of the economy.

This has brought about the need to have good modes of transport to the various tourism centres in the country. Tourism centres could well be established in order to use the natural resources of a country. This work seeks to provide the optimal routes to the major tourism centres in the Kwahu Zone of Ghana.

1.2 BACKGROUND TO THE STUDY

1.2.1 Tourism

Tourism is travel for recreational, leisure, or business purposes. The World Tourism Organization defines tourists as people "traveling to and staying in places outside their usual environment for not more than one consecutive year for leisure, business and other purposes".

Tourism has become a popular global leisure activity. After slowly recovering from the contraction resulting from the late-2000s recession, where tourism suffered a strong slowdown from the second half of 2008 through the end of 2009, and the outbreak of the H1N1 influenza virus, international tourist arrivals surpassed the milestone 1 billion tourists globally for first time in history in 2012. International tourism receipts (the travel item of the balance of payments) grew to US$1.03 trillion (€740 billion) in 2011, corresponding to an increase in real terms of 3.8% from 2010. In 2012, China became the largest spender in international tourism globally with US$102 billion, surpassing Germany and United States. China and emerging markets significantly increased their spending over the past decade, with Russia and Brazil as noteworthy examples (www.wikipedia.org).

Tourism is important, and in some cases, vital for many countries. It was recognized in the Manila Declaration on World Tourism of 1980 as "an activity essential to the life of nations because of its direct effects on the social, cultural, educational, and economic sectors of national societies and on their international relations". Tourism brings in large amounts of income in payment for goods and services available, accounting for 30% of the world's exports of services, and 6% of overall exports of goods and services. It also creates opportunities for employment in the service sector of the economy, associated with tourism. These service industries include transportation services, such as airlines, cruise ships, and taxicabs; hospitality services, such as accommodations, including hotels and resorts; and entertainment venues, such as amusement parks, casinos, shopping malls, music venues, and theatres (www.wikipedia.org).

In 1994, the United Nations classified three forms of tourism in its Recommendations on Tourism Statistics: Domestic tourism, involving residents of the given country traveling only within this country, inbound tourism, involving non-residents traveling in the given country and outbound tourism, involving residents traveling in another country.

1.2.2 History

Wealthy people have always travelled to distant parts of the world, to see great buildings, works of art, learn new languages, experience new cultures, and to taste different cuisines. Long ago, at the time of the Roman Republic, places such as Baiae were popular coastal resorts for the rich. The word tourist was used by 1772 and tourism by 1811(www.wikipedia.org).

1.2.3 Recent developments

There has been an up-trend in tourism over the last few decades, especially in Europe, where international travel for short breaks is common. Tourists have a wide range of budgets and tastes, and a wide variety of resorts and hotels have developed to cater for them. For example, some people prefer simple beach vacations, while others want more specialized holidays, quieter resorts, family-oriented holidays or niche market-targeted destination hotels. The developments in technology and transport infrastructure, such as jumbo jets, low-cost airlines and more accessible airports have made many types of tourism more affordable.

On 28 April 2009 The Guardian noted that “the WHO estimates that up to 500,000 people are on planes at any time”. There have also been changes in lifestyle, for example some retirement-age people sustain year round tourism. This is facilitated by internet sales of tourist services. Some sites have now started to offer dynamic packaging, in which an inclusive price is quoted for a tailor-made package requested by the customer upon impulse (www.wikipedia.org).

There have been a few setbacks in tourism, such as the September 11 attacks and terrorist threats to tourist destinations, such as in Bali and several European cities. Also, on 26 December 2004, a tsunami, caused by the 2004 Indian Ocean earthquake, hit the Asian countries on the Indian Ocean, including the Maldives. Thousands of lives were lost including many tourists. This, together with the vast clean-up operations, stopped or severely hampered tourism in the area for a time (www.wikipedia.org).

People partake in various forms of tourism including but not limited to ecotourism, medical tourism, educational tourism, creative tourism, dark tourism, doom tourism and sports tourism.

1.2.4 World Tourism Statistics

International tourist arrivals reached 1.035 billion in 2012, up from over 983 million in 2011, and 940 million in 2010. In 2011 and 2012, international travel demand continued to recover from the losses resulting from the late-2000s recession, where tourism suffered a strong slowdown from the second half of 2008 through the end of 2009. After a 5% increase in the first half of 2008, growth in international tourist arrivals moved into negative territory in the second half of 2008, and ended up only 2% for the year, compared to a 7% increase in 2007. The negative trend intensified during 2009, exacerbated in some countries due to the outbreak of the H1N1 influenza virus, resulting in a worldwide decline of 4.2% in 2009 to 880 million international tourists' arrivals, and a 5.7% decline in international tourism receipts. International tourism receipts grew to US$1.03 trillion (€740 billion) in 2011, corresponding to an increase in real terms of 3.8% from 2010 (www.wikipedia.org).

1.2.5 Growth

The World Tourism Organization (UNWTO) forecasts that international tourism will continue growing at the average annual rate of 4%. With the advent of e-commerce, tourism products have become one of the most traded items on the internet. Tourism products and services have been made available through intermediaries, although tourism providers (hotels, airlines, etc.), including small-scale operators, can sell their services directly. This has put pressure on intermediaries from both on-line and traditional shops (www.wikipedia.org).

It has been suggested there is a strong correlation between tourism expenditure per capita and the degree to which countries play in the global context. Not only as a result of the important economic contribution of the tourism industry, but also as an indicator of the degree of confidence with which global citizens leverage the resources of the globe for the benefit of their local economies. There has been a limited amount of orbital space tourism, with only the Russian Space Agency providing transport to date. A 2010 report into space tourism anticipated that it could become a billion dollar market by 2030 (www.wikipedia.org).

1.2.6 Latest trends

As a result of the late-2000s recession, international arrivals suffered a strong slowdown beginning in June 2008. Growth from 2007 to 2008 was only 3.7% during the first eight months of 2008. This slowdown on international tourism demand was also reflected in the air transport industry, with a negative growth in September 2008 and a 3.3% growth in passenger traffic through September. The hotel industry also reported a slowdown, with room occupancy declining. In 2009 worldwide tourism arrivals decreased by 3.8%. By the first quarter of 2009, real travel demand in the United States had fallen 6% over six quarters. While this is considerably milder than what occurred after the 9/11 attacks, the decline was at twice the rate as real GDP has fallen (www.wikipedia.org).

However, evidence suggests that tourism as global phenomena shows no signs of substantially abating in the long term. It has been suggested that travel is necessary in order to maintain relationships, as social life is increasingly networked and conducted at a distance. For many vacations and travel are increasingly being viewed as a necessity rather than a luxury, and this is reflected in tourist numbers recovering some 6.6% globally over 2009, with growth up to 8% in emerging economies (www.wikipedia.org).

1.2.7 Some Tourist Sites in Ghana

Paga Crocodile Pond, Whistling Rocks, Sivigu Craft Village, Navrongo Basilica (largest Mud-build Basilica in the world) all in the upper East Region. Kumasi Zoological Gardens, Bobiri Forest Reserve and Butterfly Sanctuary. Bonwire Kente and Craft Village, Lake Bosomtwe, Okomfo Anokye Sword Site, Manhyia Palace, Military Museum all in the Ashanti Region. Bia national park, Ankasa National Park, Nzulezu (Village on stilts), Egyambra Crocodile Sanctuary, Fort Metal Cross all in the Western Region.

Kakum National Park, Elimina Castle, Cape Coast Castle, Ajumako Craft Village, Assin Manso Reverential Gardens all in the central Region.

Tafi Tome monkey Sanctuary, Wli waterfalls Tagbo waterfalls, Mount Gemini and Afadjato all in the Volta Region. Christian burg Castle, Kwame Nkrumah Mausoleum, Ostrich Farm all in the Greater Accra Region.

Boabeng Fiema Monkey Sanctuary, Nkyeraa Waterfalls, Tano Sacred Groove, Duasidan Wildlife Sanctuary, Forikrom Boten Shrine, Dr. K. A. Busia Mausoleum and the Hani Archeological Site. Those in the Eastern Region are Boti falls, Aburi Botanical Gardens, Adomi Bridge and Akosombo Hydro Dam. The highest habitable point in Ghana, Afram river, Aligator Stone, Oboa da bo So, Stone Chair, Oku Abena waterfalls, Odweanoma paragliding center, Buruku Shrine, The ladyship etc.

1.3 STATEMENT OF THE PROBLEM

There are several modes of transport in Ghana, roads, water, air and rail. The Commonest among the transportation System in Ghana today is road transport. Road transport shows a high degree of patronage by which the tourists use to the various tourist centers. Although, there are a number of routes that the tourist use anytime they visit a particular tourist centers, the question is, which of these routes is more economical in terms of cost and time consumption without comprising access? The enormous obstacles to attaining a satisfactory rate of sustained tourism have encouraged the study to introduce some form of economic planning. When tourists visit, they may have tourists' maps, but may not be in position to see where to start and end an itinerary in order to reduce cost. While the tourists identify nature of movement at a glance on the map, they may not be able to identify the most reliable route. It is against this background that the research is being undertaken to determine the optimal route for selected tourist centres using the Kwahu Zone of the Eastern Region of Ghana as a case study.

1.4 OBJECTIVES OF THE STUDY

The objectives of the study include the under listed.

i. To determine, using Genetic Algorithm, an optimal route for selected tourist centres in the Kwahu Zone of the Eastern Region of Ghana

ii. To determine the implications for tourism of the resulting optimal route.

1.5 METHODOLOGY

The data for the study was primarily collected through visits to the selected tourist sites. The grid points (longitudes and latitudes) and the distances connecting the various sites were taken, with aid of the Global Positioning System (GPS) with high sensitivity.

The data involved information about the selected tourist centers in the Kwahu Zone of the Eastern Region. The Global Positioning System (GPS) with high sensitivity was used to determine both the Units: World Grid System (WGS) & Universal Transverse Mercater (UTM) at each tourist centers. The use of tourist map, topological maps, road map and also MATLAB and Grid point Machine

Supported computations and programming in other to set high level of accuracy in the Global Coordinates, distance calculations and the determination of the optimal routes of the tourist centers by means of an implementation of Genetic Algorithm

1.6 SIGNIFICANCE OF THE STUDY

In an attempt to identify routes that will assist minimize difficulties in travelling, high cost of transporting goods and services from one place to another, repair waste time spent in travelling, we found it imperative to undertake this study to assist motorists and tourists to travel with ease by using the shortest paths between any two or more given tourist sites.

The study will reveal to the tourists and tour operators the most appropriate routes to maximize profit by using the identified routes during their visits. In addition, transporting food stuffs and other pertinent commodities to and from the sites will be relatively economical, preventing time wasting. A fuller realization of tourist sites' potentials and the incentives for tourists will be improved. At the same time, economic activities will generally improve in the area.

1.7 LIMITATIONS OF THE STUDY

Although the study endeavored to cover the entire tourist sites in the Kwahu Zone of the Eastern Region of Ghana, not every tourist site is captured in the study due to several reasons. Some of the sites could not be reached by any means of transportation except possibly by walking long distances. Again insufficient resources and funds to capture every tourist site in the Kwahu Zone was a major setback.

1.8 ORGANIZATION OF THE THESIS

The thesis is divided into five (5) chapters. The Chapter 1 is made up of the introduction of the study; this will be made up of the background of the study with emphasis on what tourism is and what an optimal route is, statement of the problem and the objectives of the study, methodology to be used in the study and the significance of the study. Also include in this chapter, is limitations of the study and then the organization of the study. Chapter 2 is zeroed towards review of related literature; and work done by other researchers. Chapter 3 contains methodology, the mathematical model for solving optimal routes problem using Genetic Algorithm. Chapter 4 is discussion of result; emphasis will be on results obtained. Chapter 5 contains the summary of findings, conclusions and recommendations.

CHAPTER 2

LITERATURE REVIEW

Amponsah et al., (2007) considered the function optimization over integral domain: the comparative performance of elite genetic algorithm for small iterations and small generation size. The authors presented a version of the genetic algorithm (GA) which they call XSGA and found integral suboptimal global solution of one variable multimodal functions. The XSGA made use of local search and restarts. The authors' proposed XSGA worked better than the elite genetic algorithm for small generation size and small number of iterations. Both versions of the GA worked well for larger generation size and larger number of iterations.

Amponsah et al., (2010) studied the location of ambulance emergency medical service in the Kumasi metropolis of Ghana. The authors' posited that several approaches have been employed in solving this very sensitive location problem. They used the Non-Linear Maximum Expected Covering Location Problem (MEXCLP) implemented by Saydam and Aytug (2003). The author' implemented the Genetic Algorithm (GA) that uses random key coding to solve the problem. A formula for renormalization was introduced. They used real route distances for computation and statistical deviation was introduced in the selection of their optimal route.

Aickelin (1999) delved into the Genetic Algorithms for Multiple-Choice Optimization Problems. The author investigated the use of problem-specific knowledge to enhance a genetic algorithm approach to multiple-choice optimization problems. Two multiple-choice problems were considered. The first was constructing a feasible nurse roster that considered as many requests as possible and in the second problem; shops were allocated to locations in a mall subject to constraints and maximizing the overall income. Genetic Algorithms were chosen for their well-known robustness and ability to solve large and complex discrete optimization problems. The author's main theme for the study was to balance feasibility and cost of solutions. In particular, co-operative co-evolution with hierarchical sub-populations, problem structure exploiting repair schemes and indirect genetic algorithms with self-adjusting decoder functions were identified as promising approaches. The research started by applying standard genetic algorithms to the problems and explaining the failure of such approaches due to epistasis. To overcome this, problem-specific information is added in a variety of ways, some of which are designed to increase the number of feasible solutions found whilst others are intended to improve the quality of such solutions. The indirect approach was found to rely less on problem structure and hence was easier to implement and superior in solution quality. The most successful variable of the algorithm has a more than 99% chance of finding a feasible solution which is either optimal or within a few percent of optimality.

Parvez (2013) presented a review to the path planning optimization problem using genetic algorithm as a tool. The author defined path planning as term used in robotics for process of detailing a task into discrete motions. It was aimed at enabling robots with capabilities of automatically deciding and executing a sequence motion in order to achieve a task without collision with other objects in given environment. Genetic algorithms are considered as a search process used in computing to exact or approximate solutions for optimization and search problems. GAs is also termed as global search heuristics. These techniques are inspired evolutionary biology such as inheritance mutation, selection and cross over.

BjarnardÓttir (2004) used genetic algorithms to solve capacitated vehicle routing problem. The problem involved optimizing a fleet of vehicles that are to serve a number of customers from a central depot. GAs maintain a population of solutions by means of a cross over and mutation operators. Two operators were adopted from a program developed by the author: Simple Random Crossover and Simple Random Mutation. Three additional crossover operators were adopted. They were named Biggest Overlap Crossover, Horizontal line Crossover and Uniform Crossover. Three local search algorithms were designed: Simple Random Algorithm, Non-repeating Algorithm and steepest Improvement Algorithm and then two supporting operators Repairing Operator and Geographical Merge were made. Steepest Improvement Algorithm is the most effective of the Local Search Algorithms. The algorithms are called SRC –GA and VC –GA. A comparison was made of SRC-GA, VC-GA, three Tabu Search Heuristics and a new hybrid genetic algorithm, using a number of small and large problems. SRC-GA and VA-GA are on average 10.25$\pm$ 5.48% from optimum or best known values and all the other heuristics are within 1%. Thus, the algorithms are not effective enough. However, they have some good qualities, such as speed and simplicity. With that taken into account, they could make a good contribution to further work in field.

Noor (2004) said Genetic Algorithm is one of the Artificial Intelligence's approaches that adapt the idea of simulating the nature's process of evolution and natural selection through computer implementation. The study was found on the applications that used algorithms such as University Timetabling, Job scheduling, prediction and classification. It also addressed the identification of best parameters used in genetic algorithms and identifying the strengths and weaknesses of genetic algorithm.

Marian et al., (2011) focused on the second stage of a three stage, integrated methodology for modeling and optimization of distribution networks based on Hybrid Genetic Algorithms. The authors' analyzed and compared the variation of overall costs when the number of facilities varied and indicated how to minimize them. The distribution network directly and ethically affected costs, efficiency and service level of the essential performance operation indicators for supply chains. The authors concentrated on Capacitated Location Allocation of distribution centers, a large scale, highly constrained, NP-hard combinational problem. The Hybrid Genetic Algorithm used has a classical structure, but incorporates a special encoding of solutions as chromosomes and the integration of Linear Programming/Mixed Inter Programming module embedded in the generation, crossover and pseudo-mutation operators. A complex and extensive case study- 25 production facilities, 5 to 10 distribution centers and 25 retailers (up to 520 variables intricately connected with a significant number of constraints) was described, demonstrating the robustness of the Hybrid Genetic Algorithm and the optimization approach.

Rakkiannan (2012) integrated the Genetic Algorithm with the parallel version of Simulated Annealing Algorithm (SA) and applied to the job shop scheduling problem. The proposed algorithm was implemented in a distributed environment using Remote Method Invocation Concept. The implementation was done successfully to examine the convergence and effectiveness of the proposed hybrid algorithm. The Job Shop Scheduling Problems (JSSP) tested with very well-known benchmark problems, which were considered to measure the quality of proposed system. The empirical results showed that the proposed Genetic Algorithm with simulated annealing was quite successful to achieve better solution than the individual genetic or simulated annealing algorithm.

CantÚ-Paz (1998) said that Parallel GAs were particularly easy to implement and promised substantial gains in performance. As such the author's survey attempted to collect, organize and present in unified way some of the most representative publications on parallel genetic algorithms. The author presented a categorization of the techniques used to parallize GAs and showed examples. The paper described some of the most significant problems in modeling and designing multi-population parallel GAs and presented some recent advancement(s).

Gallotta (2007) studied Color Image Segmentation in reference to Genetic Algorithms. The author said that Image segmentation is a crucial problem in image processing and can determine the final outcome of many image processing tasks. Genetic Algorithms have been shown to be a viable method to segment images, according to the author. The author posited that the island model has being shown to be the most effective model for parallel Genetic Algorithms. The authors presented some of the key segmentation techniques that have been developed. The study then went into genetic algorithms and how they have been parallelized and finally looked at the little research that has been done on Grid-Based GAs.

Bhasin et al., (2003) posited that the maximum clique problem is one of the most important NP-hard problems which finds its application in numerous fields ranging from networking to the determination of the structure of a protein molecule. The authors' work examined the effect of variation of parameters for achieving optimization. They considered that the GA was more encouraging since it has exceptional searching capabilities and could be extended to many unsolvable problems and could be used in many applications from loop determination and circuit solving.

Abdelmaguid et al., (2006) introduced a new genetic algorithm approach for the integrated inventory distribution problem (IIDP). The authors presented the developed genetic representation and used a randomized version of a previously developed construction heuristic to generate the initial random population. They designed suitable crossover and mutation operators for the GA improvement phase. The authors compared the results and it showed the significance of the designed GA over the construction heuristic and demonstrated the capability of reading solutions within 20% of the optimum on sets of randomly generated test problems.

Marié (2008) presented a new evolutionary approach for solving the mult-level incapacitated facility location problem (MLUFLP). Binary encoding scheme was used with appropriate objective function containing dynamic programming approach for finding sequence of located facilities on each level to satisfy client's demands. Genetic algorithms reach all known optimal solutions for smaller dimension instances, obtained by total enumeration and CPLEX solver. Moreover, all optimal/best known solutions were reached by genetic algorithm for a single-level variant of the problem.

Ramirez (2010) considered timetabling problems as optimization problems and could be thought of as subsets of scheduling problems. Computationally, timetabling problems are NP – Complete problem. NP complete problems are problems that cannot be solved in polynomial time. Genetic algorithms are among those that can be used to find approximate solutions to NP complete problems and more specifically can be used to solve timetabling problems. Smart operators were employed during the mutation process in order to solve two high school course timetabling problems, one with a fixed master schedule and the second when the master schedule was not defined. Those smart operators were Violation-Directed Mutation (VDM), Event-Freeing Mutation (EFM), Stochastic Violation-Directed Mutation (SVDM) and stochastic Event-Freeing Mutation (SEFM).

Carvalier et al., (2012) posited that the Split Delivery Vehicle Routing Problem (SDVRP) allows customers to be assigned to multiple routes. The authors developed two hybrid genetic algorithms for the SDVRP the computational results were given for thirty-two data sets from previous literature. With respect to the total travel distance and computer time, the genetic algorithm compared favorably versus a column generation method and a two-phase method.

Fang (1994) investigated the use of genetic algorithm for solving a range of timetabling and scheduling problems. A framework was presented for GAs to solve modular timetabling problems in educational institutions. The approach involved three components: declaring problem specific constraints, constructing a problem – specific evaluation function and using a problem independent GA to attempt to solve problem. The approach relies for its success on the use of specially designed mutation operators which greatly improve upon the performance of a GA with standard operators. A framework for GAs in job- shop and open-shop scheduling was also presented. One of the key aspects of this approach is the use of specially designed representations for such scheduling problems. The representation implicitly encoded a schedule by encoding instructions for a schedule builder. When compared against a variety of common heuristic search approaches, the GA approach is clearly the most successful method over all. The general approaches also shown to be readily extendable to rescheduling and dynamic scheduling.

BermÚdez et al., (2002) presented a genetic algorithm for assigning doors in less-than-truckload (LTL) breakbulk terminals. Typically, strip (incoming freight) doors and stack (outgoing freight) doors were assigned to city areas. Incoming trailer loads were broken up and moved to appropriate outgoing trailers. The objective was to maximize the total weighted travel distance, a surrogate for labour cost and cycle time .The underlying problem was a Quadratic Assignment Problem with the experiments based on real world data.

Sodsee et al., (2012) concerned themselves with a GA focusing on the Pareto based approach for solving multi-objective optimitation problems. Multi-objective Bisexual Reproduction Genetic algorithms (MOBRGA) was proposed. MOBRGA uses a concept of sexual selection with different types and mutation rates in reproducing offspring. The results for MOBRGA and other algorithms showed that the proposed algorithm performed well in some benchmark multi-objective optimization problems. In Shaffer's and Murata's problems, MOBRGA could find the Pareto front which is the line of the non-dominated solutions to the problems. MOBRGA could successfully find network design solutions that meet user's requirements.

Sadrsadat et al., (2012) considered the Bus Network Design Using Genetic Algorithms as an important problem in transportation planning. The problem concerned determining a network of bus lines which best achieves a predetermined objective. The authors devoted the study to solving the problem using GA. The fitness function was defined as the benefit to the users of the bus network less the cost of the operator of the network, which is to be maximized subject to constraints that properly distribute bus routes over the study area. Several good solutions were generated through a sensitivity analysis by changing the parameters of the problem affecting bus route geographical distribution. The authors solved a network assignment problem for each of the alternative bus networks and several measures of effectiveness were evaluated for them. A multi-objective analysis (concordance analysis) was performed based on 10 measures of effectiveness and 14 weighting systems and a bus network was proposed for the city of Mashad, Iran.

Chang et al., (2005) introduced the two-phase sub population genetic algorithm to solve the paralle machine-scheduling problem. In the first phase the population was decomposed into many sub-populations and each sub-population was designed for a scalar multi-objective. In the second phase, non-dominant solutions were combined after the first phase and all sub-populations were unified as one big population. The authors algorithm did not only merge the sub-populations but the external memory of pareto solution was also search for a specific weighted objective during the next evolution process. The two-phase sub-population genetic algorithm was applied to solve the parallel machine-scheduling problems in testing of the efficiency and efficacy.

Agarwal (2012) posited that in today's information age, information sharing and transfer has increased exponentially. Security, integrity, non-repudiation, confidentiality and authentication services are the most important factors in information security. Many different image encryption methods have been proposed to keep the security of these images. Images encryption technique tries to convert an image to another image that is hard to understand. Gas are used, through modeling, to a simplified version of genetic processes to solve many optimization problems. The author proposed a method based on GA which was used to produce a new encryption method by exploitation of the powerful features of the crossover and Mutation operations of GA.

Kumar et al., (2009) suggested that scheduling of production in flexible Manufacturing Systems (FMSs) has been extensively investigated over the past years and it continues to attract the interest of both academic researchers and practitioners. The generation of new and modified production schedules is becoming a necessity in today's complex manufacturing environment. GAs were used to obtain an initial schedule whiles uncertainties in the production environment and modeling limitations inevitably result in deviations from the generated schedules which made rescheduling or reactive scheduling essential. The authors considered unforeseen machine break-downs increased order priority, rush order arrival and order cancellations as some of the uncertainties that cause discrepancies between actual output and planned output. The algorithms revise only those operations that must be rescheduled.

Bhasin et al., (2012) suggested the problem of finding a minimum vertex cover as an NP hard optimitation problem. The author considered some of the approximation algorithms for the problem as neither optimal nor complete. The author proposed the use of the theory of natural selection via Genetic Algorithms (GAs) for solving the problem. The proposed work was tested for some constrained inputs and the results proved encouraging. The authors discussed the application of genetic algorithms to the solution and the requisite analysis. The approach presented a GAs based solution to problem.

Fu et al., (2003) suggestion that in dealing with very large data set, it might be impracticable to construct a decision tree using all of the points and even whiles it is practical, this might not be the best way to utilize the data. Asan alternative, subsets of the original data were extracted, a tree construction on each subset and parts of individual trees were combined in a smart way to produce an improved final set of feasible trees. The authors took trees generated by a commercial decision tree package, namely, C4.5, and allowed them to yield trees of better quality. The authors divided their data set into training, scoring, and test sets and their approach produced uniformly high quality decision trees. The authors investigated the impact of scaling and demonstrated that their approach could be used effectively on very large data sets

Súer (2012) posited that minimizing total tardiness in single machine scheduling is NP-hard. The authors extended the problem to include non-zero ready times and the preemption of jobs was not allowed. The author first developed a mathematical model. Due to computational complexities with the mathematical model, a GA approach was also proposed and its performance was compared with optimal solutions. The results also showed that GA can find optimal solutions for small problems and near optimal solutions for large problem. The results also showed that among Delay-only, Non-delay-only and Random strategies, Non-Delay strategy produced more robust solution whereas random strategy found the optimal solution in smaller problem categories.

Al-Dulaimi (2008) coded in genetic form the well known NP-complete of the Traveling Salesman Problem (TSP). The author proposed the software system to determine the optimum route for a TSP using GA technique. The system started from a matrix of the calculated Euclidean distances between the cities to be visited by the traveling salesman and a randomly chosen city order as the initial population. New generations were then created repeatedly until the proper path was reached upon reaching a stopping criterion. This search was guided by a solution evaluation function.

Blanco (1999) considered selecting the optimal topology of a neural network for a particular application to be a difficult task. In the case of recurrent neural networks, most methods only induced topologies in which their neurons were fully connected. The authors presented a genetic algorithm capable of obtaining not only the optimal topology of a recurrent neural network but also the least number of connections necessary. Finally, their GA was applied to a problem of grammatical inference using neural networks, with very good results.

Wu et al., (2009) suggested that Geometry modeling has become an increasingly powerful approach for architecture and building design. It is an effective approach especially when a geometry model is constructed by making use of the associative parameters, so- called algorithm-based parametric design, which enables designers to easily change the desired geometry parameters and thus fine tune the design. An integrated design tool was implemented for the optimization of parametric geometry design. Base on evolutionary search algorithm and geometry modeling tools such as Generative component, a design was represented by encoding designs variable onto a binary string or genotype, design alternatives were evolved by mimicking crossover, mutation and natural selection principle of Darwin's survival-of-the-fittest. Solutions are optimized generations after generation via emulating natural evolution.

Slavov et al., (2012) presented a feed forward feedback (PID) controller designed for control of glucose concentration during the E. Coli fed-batch cultivation process. The controller was used to control the feed rate and to maintain glucose concentration at a desired set point. The authors suggested an equation for correcting the measured glucose based on Kalman filter estimates of biomass concentration and bacteria growth rate. The authors used GA turning of the

PID controller to achieve good closed-loop system performance. For a short time the controller sets the control variable and maintained it at the desired set point during the process tuning of the controller on the basis of a genetic algorithm led to higher level of accuracy and efficiency of the system performance.

Fu et al., (2005) suggested that there are a number of transportation applications that require the use of a heuristic shortest path algorithm rather than one of the standard optimal algorithms. This is due to the requirements of some transportation applications where shortest paths used need to be recalculated repeatedly. The authors presented a survey review of various heuristic shortest path algorithms that have been developed in the past.

Garkaz et al., (2010) predicted the bankruptcy using genetic algorithm. The authors found a complete list of financial proportions that showed high capabilities of predicting the bankruptcy. These proportions included the ratio of operational income to sale, ratio of total debts to total assets, current assets to current debts, sale to current assets and interest cost to grass profit. The authors investigated the possibility of the use of the genetic algorithm to predict the bankruptcy of the accepted companies in Tehran Stock Exchange. The independence t-test showed that there was a meaningful difference between the average of these ratios of bankrupted group with that of non-bankrupt one.

Barrero et al., (2010) suggested that despite the excellent results of GAs, their use generated new problems. One of such problems was how to provide a good fitting in the usually large number of parameters that must be turned to allow a good performance. The authors described a new platform that was able to extract the Regular Expression that matched a set of examples, using a suggested learning and agent-based framework. In order to do that, GA based agents decompose the GA execution in a distributed sequence of operations performed by them. The plat formed was applied to language induction problem, for that reason the experiments were focused on the extraction of the regular expression that matched a set of (in terms of fitness value) applied to three case studies: e-mails, phone numbers and URLs. Moreover it described how the codification of the alphabet affected the performance of the platform.

Elsayed et al., (2010) posited that End-Of-Life (EOL) processing options include revise, remanufacturing, recycling and proper disposal. In almost all the cases, a certain level of disassembly was required due to possible changes in the original product structure. Thus, finding an optimal or near optimal disassembly sequence was crucial to increasing the efficiency of the process. Disassembly operations are labour intensive, could be costly, have unique characteristics and could not be considered as reverse of assembly operarations. The authors presented a GA for disassembly sequencing of EOL products.

Tyagi et al., (2012) introduced the concepts of a GA and described a GA-based heuristic for solving the flow shop scheduling problems. The flow shop scheduling problem is a production problem where a set of n jobs have to be processed with identical flow patterns on m machines. The authors suggested that heuristics played a major role in solving the NP-hard combinational optimization problems. The authors described a GA-based heuristic to make-spam minimization on flowshop scheduling. The authors compared their heuristic with the NEH (Nawaz, Enscore, Ham) Algorithm which is the most popular heuristic in the literature. The author's computational experience showed that the GA approach provided competitive results for flowshop scheduling problems.

Shah et al., (2006) suggested that early and accurate detection of cancer was critical to the patient's well being. Analysis of gene expression data sets for ovarian, prostate and lung cancer was carried out. The authors proposed an integrated an integrated gene-search algorithm for genetic expression data analysis. The integrated algorithm involved a genetic algorithm and correlation-based heuristics for data preprocessing (on partitioned data sets and data mining) (decision tree and support vector machines algorithms) to make predictions.

The authors further applied bagging and stacking algorithms to enhance the classification accuracy. The mapping of genotype information to the phenotype parameters would ultimately reduce the cost and complexity of cancer detection and classification.

Khan (2012) explained minimization of makes pan or total completion time for n jobs, m-machines, no-wait flowshop problem (NW-FSSP). The authors proposed for the NWFSSP a spreadsheet based general purpose genetic algorithm. The example analysis showed that the proposed approach produced results comparable to the previous approaches. The authors demonstrated that the current application is a general purpose approach whereby the objective function could be tailored without any change in the logic of the GA routine.

Konanur(2005) suggested that the physics behind motor cycle riding are well understood and implemented by studying the laws of kinetics and kinematics behind the operation of single track motor vehicle. The author worked with an application which is currently using open GL and implemented an interactive motorcycle simulator which is based on the laws of physics. The application involved a multi-agent pilot capable of autonomously riding the motor cycle using some configurable equations. The author applied GAs to find suitable value for the parameters of the pilot by testing in anon graphical environment and visually verified the results of the GAs with the graphical interface application and comparisons made.

Dridi et al., (2011) suggested the PDPTW as an optimization vehicles routing problem which must meet requests for transport between suppliers and costumers in purpose to satisfy precedence, capacity and time constraints. The authors presented a GA for multi-objective optimization of a multi-pick up and delivery problem with the time windows (m-PDPTW), based on aggregation method and lower bounds. The authors proposed a brief literature review of the PDPTW and presented the approach to give a satisfying solution to the m-PDPTW minimizing the compromise between total travel cost and total tardiness time.

Resende et al., (2004) presented a GA for the Resource Constrained Multi-Project (RCMPSP). The chromosome representation of the problem was based on random key. The schedules were constructed using a heuristic that built parameterized active schedules based on priorities, delay times and release dates defined by the genetic algorithm. The approach was tested on a set of randomly generated problems. The computational results validated the effectiveness of the proposed algorithm.

Tuzkaya et al., (2012) investigated the vehicle routing problem with time windows (VRPTW) and formulated as a multi-objective model. The authors proposed a hybrid meta-heuristic algorithm as a solution approach. The proposed algorithm consisted of two meta-heuristics: GA and simulated annealing (SA). SA was used as an improvement operator in GA. Besides, a hypothetical application was presented to foster the better understanding of the proposed model and algorithm. The validity of the algorithm was tested via some well-known benchmark problems from the literature.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

In this chapter, we shall consider the appropriate methodologies that shall be useful in the study. We shall touch on the various mathematical theories and algorithms required to solve the problem. The profile of the study area would be put forward. The other areas to be considered shall include but not limited to the working principles of genetic algorithms, encoding, genetic algorithm operators, crossover, mutation, replacement, search termination, how genetic algorithms work, when to use genetic algorithm, building block hypothesis, schema theorem, no-free-lunch theorem, distinction between genetic algorithms and other optimization techniques and the Floyd-Warshall algorithm.

3.2 PROFILE OF THE STUDY AREA

The Kwahu zone is located on the Kwahu Mountain, with some of its villages beneath the mountain. The area has a vast green belt. The Kwahu zone has the river Asuboni, a source of fish and other resources. The land in the Kwahu zone is a little bit rocky in some areas. The Kwahu zone has the famous Odweanoma Mountain where the paragliding festival is held annually during the Easter festivities. The weather is relatively cool. The people of Kwahu mostly migrated from the Ashanti region of Ghana to the mountains to seek protection from persecutors. The people of Kwahu undertake numerous economic activities including trading, farming and making of pottery. Pottery making is a preserve of the women. The major towns in the zone include Nkawkaw,Afram Plains, Atibie, Asakraka, Bepong, Obomeng and Obo.

3.3 WORKING PRINCIPLES OF GENETIC ALGORITHMS

Genetic Algorithm is a search heuristic that mimics the process of evaluation. Genetic Algorithms can be applied to process controllers for their optimization using natural operators. Genetic Algorithms are applied to direct torque control of induction motor drive, speed control of gas turbine, speed control of DC servo motor for the optimization of control parameters. The simulations could be carried out in simulink package of MATLAB. The workability of genetic algorithms (GAs) is based on Darwinian's theory of survival of the fittest. Genetic algorithms (GAs) may contain a chromosome, a gene, set of population, fitness, fitness function, breeding, mutation and selection. Genetic algorithms (GAs) begin with a set of solutions represented by chromosomes, called population. Solutions from one population are taken and used to form a new population, which is motivated by the possibility that the new population will be better than the old one. Further, solutions are selected according to their fitness to form new solutions, that is, offspring. The above process is repeated until some condition is satisfied. Algorithmically, the basic genetic algorithm (GAs) is outlined as below:

Step 1 [Start] Generate random population of chromosomes, that is, suitable solutions for the problem.

Step 2 [Fitness] Evaluate the fitness of each chromosome in the population.

Step 3 [New population] Create a new population by repeating following steps until the new population is complete.

a. [Selection] Select two parent chromosomes from a population according to their fitness. Better the fitness, the bigger chance to be selected to be the parent.

b. [Crossover] With a crossover probability, cross over the parents to form new offspring, that is, children. If no crossover was performed, offspring is the exact copy of parents.

c. [Mutation] With a mutation probability, mutate new offspring at each locus.

d. [Accepting] Place new offspring in the new population.

Step 4 [Replace] Use new generated population for a further run of the algorithm.

Step 5 [Test] If the end condition is satisfied, stop, and return the best solution in current population.

Step 6 [Loop] Go to step 2.

Each iteration of this process is called a generation. A GA is typically iterated for anywhere from 50 to 500 or more generations. The entire set of generations is called a run. At the end of a run there are often one or more highly fit chromosomes in the population. Since randomness plays a large role in each run, two runs with different random-number seeds will generally produce different detailed behaviours. GA researchers often report statistics (such as the best fitness found in a run and the generation at which the individual with that best fitness was discovered) average over many different runs of the GA on the same problem. The simple procedure just described is the basis for most applications of GAs. There are a number of details to fill in, such as the size of the population and the probabilities of crossover and mutation, and the success of the algorithm often depends greatly on these details. As a more detailed example of a simple GA, suppose that l (string length) is 8, that f(x) is equal to the number of ones in bit string x (an extremely simple fitness function, used here only for illustrative purposes), that n (the population size) is 4, that pc =0 and that pm =0.001.

(Like the fitness function, these values of l and n were chosen simplicity. More typical values of l and n are in the range 50-1000. The values for and pm are fairly typical.) The initial (randomly generated) population might look like this:

Table 3.1: Initial (randomly generated) population

Chromosomes Label	Chromosomes String	Fitness
A	00000110	2
B	11101110	6
C	00100000	1
D	00110100	3

A common selection method in GAs fitness-proportionate selection, in which the number of times an individual is expected to reproduce is equal to its fitness divided by the average of fitness in the population. (This is equivalent to what biologists call "viability selection".) Genetic Algorithm is a random search optimization technique that has it roots in the principle of genetics. Before a GA can be run, a suitable coding (or representation) for the problem must be advised. We also require a fitness function, which assigns a figure of merit to each coded solution. During the run, parents must be selected for the reproduction, and recombined to generate offspring. Genetic diversity or variation is a necessity for the process of evolution. The genetic operators used in GAs maintain genetic diversity. Genetic operators are analogues to those which occur in real world. As already mentioned, GAs evolve a population of individuals according to the process of natural selection. During this process, genetic operators create new individuals from highly fit old individuals. These operators are used after the coding process and the genetic algorithm enters the reproduction stage. Holland's introduction of a population based algorithm with selection, crossover, inversion and mutation was a major innovation. These operators are used at different stages of the GA. In addition to these operators, there are some parameters of GAs. One important parameter is Population Size. The population size says how many chromosomes are in population, if there are only few chromosomes, then GA would have a few possibilities to perform crossover and only a small part of search space is explored, if there are many chromosomes, then GA slows down. Research shows that after some limit, it is not useful to increase population size, because it does not help in solving the problem faster. The population size depends on the type of encoding and the problem.

3.4 ENCODING

Encoding techniques in genetic algorithms (GAs) are problem specific, which transforms the problem solution into chromosomes. Various encoding techniques used in genetic algorithms (GAs) are binary encoding, permutation encoding, value encoding and tree encoding. Binary encoding is the most common form of encoding in which the data value is converted into binary strings. Binary encoding gives many possible chromosomes with a small number of alleles. A chromosome is represent

Chromosome 1	2.4351	3.8609	4.110	6.783
Chromosome 2	north	south	east	west

Permutation encoding is best suited for ordering or queuing problems. Travelling salesman is a challenging problem in optimization, where permutation encoding is used. In permutation encoding, every chromosome is a string of numbers in a sequence as shown below.

Chromosome 1	3	4	2	7	1	5	6	8
Chromosome 2	8	3	6	1	2	7	4	5

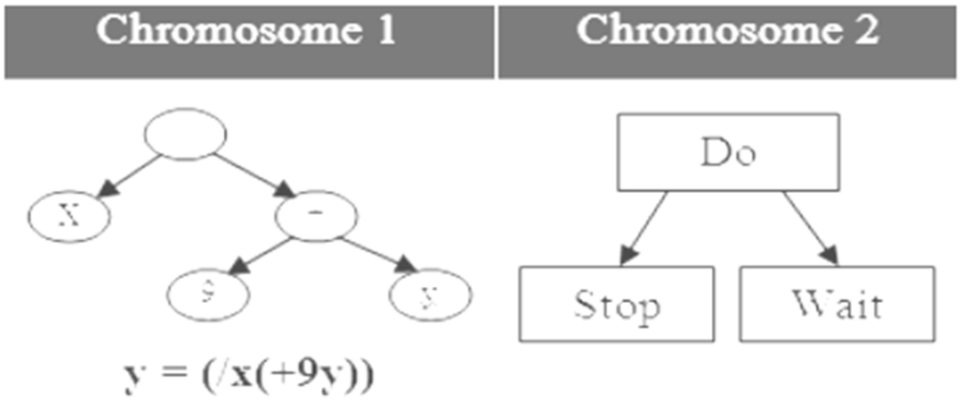


Figure 3.1: Chromosome representation in tree encoding

Value encoding Value encoding can be form number, real number on characters to some complicated objects. Value encoding is a technique in which every chromosome is a string of some values and is used where some more complicated values are required. It can be expressed as shown below;

Chromosome 1	2.4351	3.8609	4.110	6.783
Chromosome 2	north	south	east	west

Tree Encoding is best suited technique for evolving expressions or programs such as genetic programming. In tree encoding, every chromosome is a tree of some objects, functions or commands in programming languages. Locator/identifier separation protocol (LISP) programming language is used for this purpose. Locator/identifier separation protocol (LISP) programs can be represented in tree structure for crossover and mutation. In tree encoding, the chromosomes are represented as shown below;

There are no specific directions for using the type of encoding scheme in the specified problem rather, it depends upon the applicability and the requirements of the problem.

3.4.1 Scaling and Fitness function

All scaling functions can (theoretically) be divided into three categories: Linear, Sigma Truncation, and Power Law. Linear scaling methods (such as flinear below) usually have constants that are not problem dependent, but that may depend on the population characteristics (max, min, mean, etc). Sigma methods include problem depended data. Power scaling takes into account the raw fitness values themselves. For these trials, the four methods of scaling all fall under the category of population dependent, linear scaling. One of the most common scaling techniques is traditional linear scaling. This scaling remaps the fitness values of each individual using the following equation

$$f_{linear} = a + b.f_{raw}$$

where a and b are constants defined by the user. For these trials, the values of a and b were tied to specific characteristics of the population. Another scaling option is rank scaling. This is more of a two-step process. First, all individuals are sorted by their raw fitness scores (they are "ranked"). Then new fitness values are computed based solely on their "rank" using

$$f_{ranked} = p - 2.\frac{(r-1)(p-1)}{N-1}$$

where r is the "rank" of the individual, p is the desired selection pressure (best/median ratio), and N is the size of the population. Exponential scaling also begins with ranking all the individuals, but the new fitness values are instead computed with

$$f_{experiment} = m^{(r-1)}$$

where each individual's new fitness is m times greater than the previous individual. Low m can result in high selection pressure and all that that implies. Low pressure means premature convergence, possibly isolating the entire population in a local maximum and not the true maximum. Top scaling is probably the most simple scaling method. Using this approach, several of the top individuals have their fitness set to the same value (which is proportional to the population size), with all remaining individuals having their fitness values set to zero. This simple concept yields

$$f_{top} = \begin{cases} s.N, & \text{for } r \geq c \\ 0 & \text{for } r < c \end{cases}$$

where s is some proportionality constant, c is the number of individuals that will be scaled up, and N is the size of the population. Since this gives several individuals identical fitness levels, regardless of how different their raw scores might be, the diversity of the succeeding generations is increased. Of these four scaling options, all have arbitrary user inputs. In an attempt at fairness, the values used in the following tests were selected randomly, but only after they proved not to dramatically affect the outcome of the trials. This was done so that no one scaling method would have an advantage due to better tuning by the user. At the end of the paper, another, possibly better, alternative will be discussed.

3.4.2 Fitness functions

All four scaling methods will be measured against each other given a constant set of population parameters (population size, individual length, etc). Mutation and crossover rates will also be held constant throughout all the trials. Four fitness functions will be used to demonstrate the scaling methods' effects. Let

$$f_1(x_t) = \sum_{t=1}^{v} x_i^2$$

where v is the number of variables (in this and all following examples, v=2) or dimensions. For these trials, this function was slightly modified (see Fig. 3.2.): the dome was inverted (opening downwards now) and the peak was moved

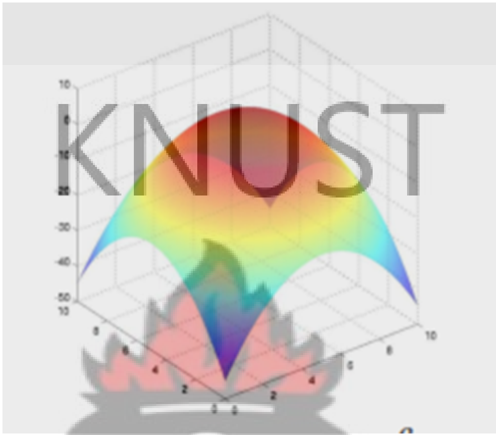


Figure 3.2: An example diagram on fitness function

off of the origin to (5,5,5). The x- and y-axis values ranged from 0 to 10. This function tests the algorithm's ability to focus on the true maximum. This is because there is little differentiation between two points very close to each other when they are both close to the peak (the function flattens out at the peak). In order to select the true maximum, the function must be able to discriminate between fitness levels that are very close. The second fitness function is

$$f_2(x_i) = \sum_{i=1}^{v} \text{integer}(x_i)$$

The function integer(x) effectively rounds the values of x to the nearest integer. The function looks like the side of a hill with hundreds of flat terraces (see Figure 3.3).

Here the x- and y-axis values ranged from −10 to 10. The third test function for these trials,

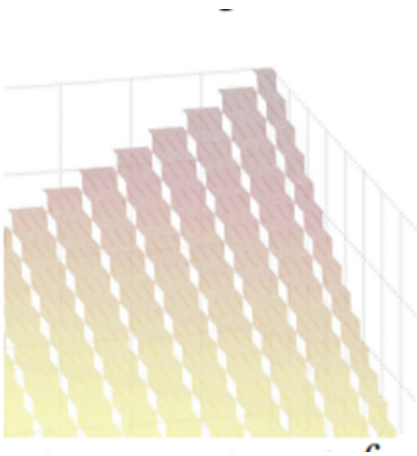


Figure 3.3: An example diagram on fitness function

was very similar to f2 above, the primary difference being that this one more resembles a staircase. This function offers a hidden difficulty. Since its maximum fitness is zero, the algorithm must be able to handle such a value without conflict. This means that there should be no point in the algorithm where anything is divided by the maximum fitness. This may seem to be an innocent quirk, but it has the potential to disrupt some algorithms. Note that this function is symmetric in the y-axis (see Figure 3.4).

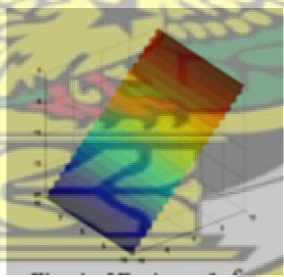


Figure 3.4: An example diagram on fitness function

Both f2 and f3 have multiple solutions that have equally maximum fitness values (the highest points form a plateau rather than a peak). This can cause problems when the population prematurely converges. If this happens, then there is nothing in a specific terrace to indicate that there are higher fitness levels elsewhere.

The final test function is not analytical, but rather is a plot of actual data. In this case the z-axis represents the Kolmogorov distance between two targets. The x-axis and y-axis in turn represent pairs of transmission / reception polarization angles. There are numerous hills and valleys, with steep walls and plateaus (see Figure 3.5).

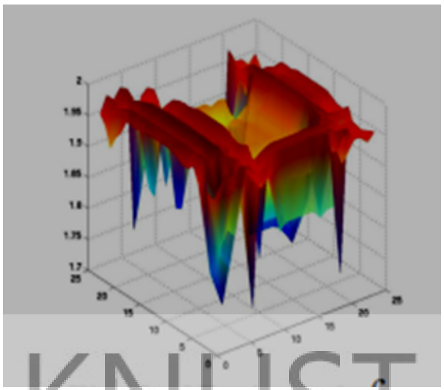


Figure 3.5: An example diagram on fitness function

The peak resides on the edge of a plateau, next to a steep drop off. Since this is a real data set, the function only has integer inputs (ranging from 1 to 24), which are the coordinates of the Kolmogorov distance in the data array. Therefore, since the algorithm naturally produces double precision output, the phenotype generated by the algorithm must be rounded down to the nearest integer (in this case, all numbers to the right of the decimal are dropped). This function represents a typical real-world application of a genetic algorithm (albeit, since this data set is quite small, the genetic algorithm is overkill). The Kolmogorov distance is calculated with

$$f_4 = \int |p(x|w_1)p_1 - p(x|w_2)p_2| \, dx$$

where p is the conditional probability density function, p1 and p2 are a priori probabilities of classes w_1 and w_2 and x is the received (and as yet unclassified) signature.

3.5 GENETIC ALGORITHM OPERATORS

3.5.1 Breeding and Selection

Breeding

- The breeding process is the heart of the genetic algorithm, where we create new and hopefully better and fitter individuals.
- The breeding cycle consists of three steps:

  a Selecting parents.

  b Crossing the parents to create new individuals (offspring or children).

  c Replacing old individuals in the population with the new ones.

Selection

- Selection is the process of choosing (two) parents from the population for crossing.
- After deciding on an encoding, the next step is to decide how to perform selection of individuals the population that will create offspring for the next generation and how many offspring will created.
- The purpose of selection is to emphasize fitter individuals in the population in hopes that the offspring have higher fitness.
- Selection is a method that randomly picks chromosomes out of the population according to the valuation function. The higher the fitness function, the more chance an individual has to be selected.
- Selection guides the GA to improve the population fitness over the successive generations.

- The convergence rate of GA is largely determined by the magnitude of the selection procedure.

- Genetic Algorithms should be able to identify optimal or nearly optimal solutions.

- If the selection pressure is too low, the convergence rate will be slow; the GA will take longer time to find the optimal solution.

- If the selection pressure is too high, there is an increased change of the GA prematurely converging to an incorrect or sub-optimal solution.

- Selection schemes should also preserve population diversity, as this helps to avoid premature convergence.

- Selection schemes can be probabilistic where chances of being selected are proportional to fitness, yet it is possible for less fit individuals to be selected. They can also be greedy, where only the fittest solutions are selected.

1- Roulette Wheel Selection

- Roulette selection is one of the traditional GA selection techniques.

- The principle of roulette selection is a linear search through a roulette wheel with the slots in the wheel weighted in proportion to the individual's fitness values.

- A target value is set, which is a random proportion of the sum of the fitnesses in the population. The population is scanned one by one until the target value is reached.

- Fit individuals are not guaranteed to be selected, but somewhat have a greater chance.

* It is essential that the population not be sorted by fitness, since this would dramatically bias the selection.

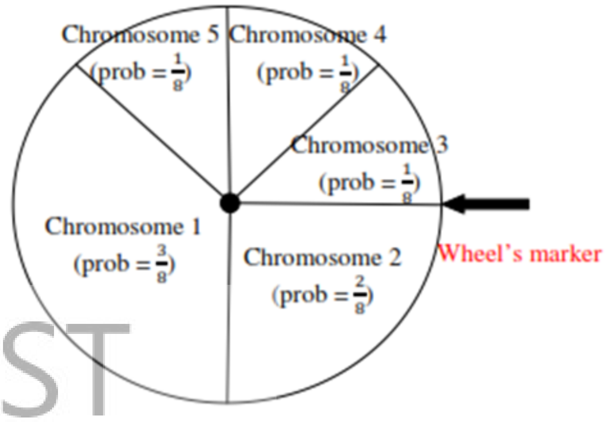


Figure 3.6: Roulette Wheels **A**

- The expected value of an individual is its fitness divided by the actual fitness of the population.
- Each individual is assigned a slice of the roulette wheel.
- The wheel is spun N times, where N is the number of individuals in the population.
- On each spin, the individual under the wheel's marker is selected to be in the pool of parents for the next generation.
- There are many implementations of the roulette methods.

1. Let T be the sum of the total expected value of the individuals in the population.
2. Repeat N times:
   i. Choose a random integer 'r' between 0 and T.
   ii. Loop through the individuals in the population, summing the expected values, until the sum is greater than or equal to 'r'. Then we pick up the individual we stop at, and repeat step (i) for the next individual.

Example 3.5.1 Objective function: Maximize $x^2 - 3x + 2$ Initial population = {4, 6, 10, 12}

X	Expected Value
4	6
6	20
10	72
12	110
Total	208

We select a random number between 0 and 208: 30. At 4, the sum of expected values = 6. At 6, the sum of expected values = 26. At 10 the sum of expected values is 92. So 10 is selected. We select again another number between 0 and 208: 115. At 12, the sum of expected values = 110. At 4, the sum of expected values = 116. So, 4 is selected. We select again another number between 0 and 208: 101. At 6, the sum of expected values = 20. At 10, the sum of expected values = 92. At 12, the sum of expected values = 202. So, 12 is selected.

2- Random Selection

- This technique randomly selects parents from the population.
- Random selection does not take in consideration ranking of parents.
- It does not follow the rule that good characteristics are passed to offspring.

3- Rank Selection

- The Roulette wheel will have a problem when the fitness values differ very much.

- In case an individual has 90% of fitness, then it occupies most the surface of the Roulette wheel, so other individuals have less chance to be selected.

- In Rank Selection ranks the population as follows: For N individuals, the worst individual has fitness = 1 and the best has fitness = N.

  * Rank selection is characterized by slow convergence compared to Roulette wheel.

  * Rank selection also prevents too quick convergence.

  * It keeps up selection pressure when the fitness variance is low.

  * In case of high variance, it preserves diversity which leads to a successful search.

  * There are many ways to implement Rank Selection.

- Technique 1:

  - r is a parameter to be chosen (0 ¡ r ¡ 1).

  - Select a pair of individuals at random (Ind1, Ind2).

  - Generate a random number R between 0 and 1.

  - If R ¡ r use the first individual as a parent (P1= Ind1).

  - If the R¿=r then use the second individual as the parent (P1 = Ind2).

  - Repeat generating R to select the second parent (P2)

- Technique 2:

  - Select two individuals at random (Ind1, Ind2).

  - The individual with the highest evaluation becomes the parent (P1).

  - Repeat to find a second parent (P2).

3- Tournament Selection

- An ideal selection strategy should be such that it is able to adjust its selective pressure and population diversity.
- Unlike, the Roulette wheel selection, the tournament selection strategy provides selective pressure by holding a tournament competition among k individuals.
- Selection pressure is directly proportional to the number k of participants.
- The best individual from the tournament is the one with the highest fitness, which is the winner of k.
- The winners of the tournaments are then inserted into the mating pool.
- The tournament competition is repeated until the mating pool for generating new offspring is filled.

  (a) Randomly select k individuals for a tournament.

  (b) Extract the best individual and insert it in the mating pool

  (c) Repeat step 1 until mating pool is filled

4- Stochastic Universal Sampling

- Stochastic universal sampling provides zero bias, yet individuals of better fitness have better chance to be selected.
- The individuals are mapped to contiguous segments of a line or circle, such that each individual's segment is equal in size to its fitness exactly as in roulette-wheel selection.
- Here equally spaced pointers are placed over the line, as many as there are individuals to be selected.

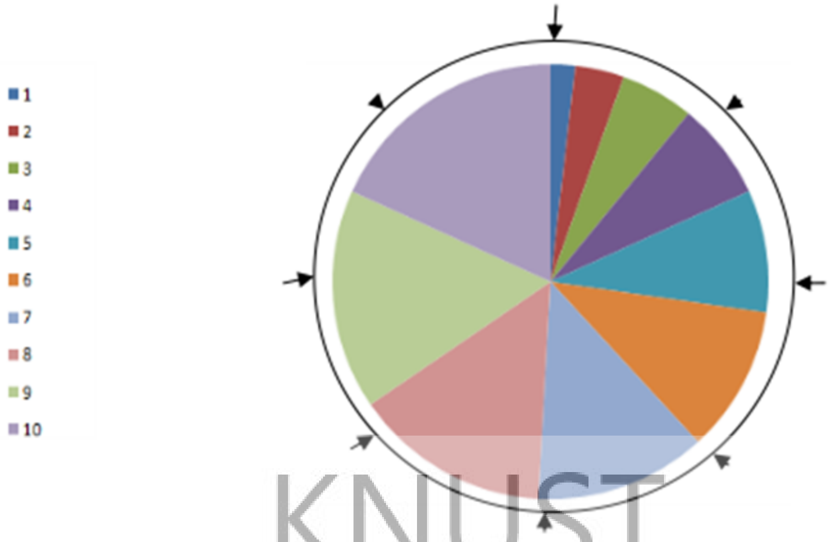
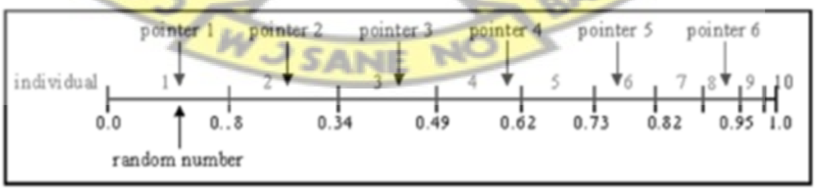


Figure 3.7: Roulette Wheels **B**

- Implementation:
- Consider N the number of individuals to be selected, then the distance between the pointers are 1/N.
- The position of the first pointer is given by a randomly generated number in the range [0, 1/N].
- For 6 individuals to be selected, the distance between the pointers is 1/6=0.125.
- Sample of 1 random number in the range [0, 0.167]: 0.1.
- After selection the mating population consists of the individuals, 1, 2, 3, 4, 6,



8.

5- Elitism

- The first best chromosome (2, 3, . . .) are copied directly to the new population.

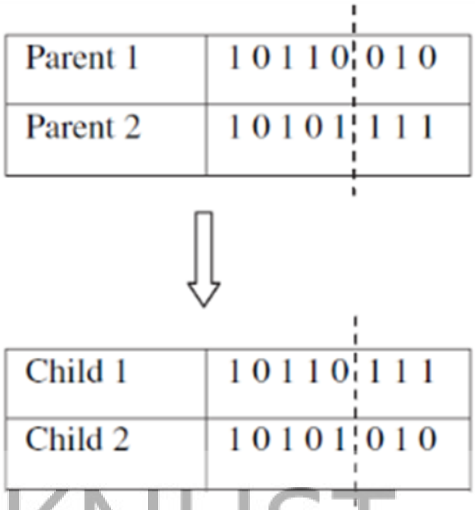
- The rest of the new population is generated through one of the previous techniques.
- Because the classical selection techniques and crossover do not guarantee better offspring, good individuals (better solutions) can be lost if they are not selected.
- Elitism significantly improves the GA's performance.

3.5.2 Crossover (Recombination)

Crossover is the process of taking two parent solutions (after selection process) and producing from them a child or an offspring. Reproduction makes clones of good strings but does not create new ones. Crossover operator is applied to the mating pool with the hope that it creates a better offspring. Crossover is a recombination operator that proceeds in three steps: Select at random a pair of two individual strings for the mating. A cross site is selected at random along the string length. The position values are swapped between the two strings following the cross site. There are various techniques for crossover.

Single Point Crossover

The traditional genetic algorithm uses single point crossover, where the two mating chromosomes are cut once at corresponding points and the sections after

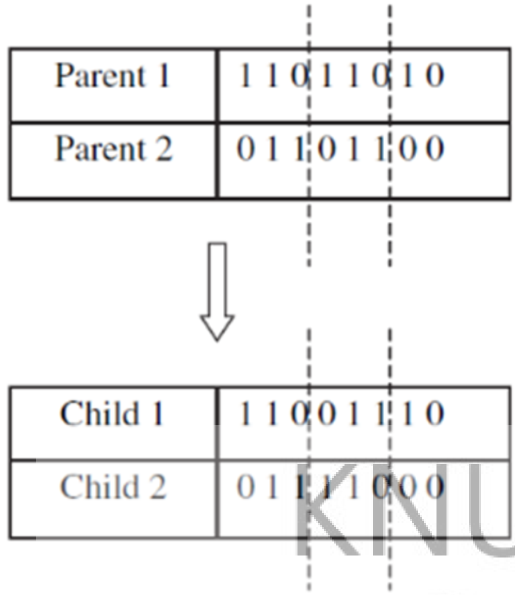


the cuts are swapped.

Here, a cross-site or crossover point is selected randomly along the length of the mated strings and bits next to the cross-sites are exchanged.

Two Point Crossover

Apart from single point crossover, many different crossover algorithms have been devised, often involving more than one cut point. Adding further crossover points reduces the performance of the GA. An advantage of having more crossover points is that diverse offspring are created, i.e., searching is more diverse. In two-point crossover, two crossover points are chosen and the contents between these points are exchanged between two mated parents.



Originally, GAs were using one-point crossover which cuts two chromosomes in one point and splices the two halves to create new ones. But with this one-point crossover, the head and the tail of one chromosome cannot be passed together to the offspring. So, if both the head and the tail of a chromosome contain good genetic information, none of the offsprings obtained directly with one-point crossover will share the two good features. Using a 2-point crossover is generally considered better than 1-point crossover.

Multi-Point Crossover (N-Point crossover)

In multipoint crossover, N random cross-sites are chosen. There are two ways in this crossover: One is even number of cross-sites and the other odd number of cross-sites. In the case of even number of cross-sites, cross-sites are selected randomly. In the case of odd number of cross-sites, a different cross-point is always assumed at the string beginning.

Uniform Crossover

Each gene in the offspring is created by copying the corresponding gene from one or the other parent chosen according to a random generated binary crossover mask of the same length as the chromosomes. Where there is a 1 in the crossover mask, the gene is copied from the first parent, and where there is a 0 in the mask the gene is copied from the second parent.

Parent 1	1 0 1 1 0 0 1 1
Parent 2	0 0 0 1 1 0 1 0
Mask	1 1 0 1 0 1 1 0
Child 1	1 0 0 1 1 0 1 0
Child 2	0 0 1 1 0 0 1 1

A new crossover mask is randomly generated for each pair of parents. The Uniform Crossover uses a fixed mixing ratio between two parents. Unlike one-point and two-point crossover, the Uniform Crossover enables the parent chromosomes to contribute the gene level rather than the segment level. If the mixing ratio is 0.5, the offspring has approximately half of the genes from first parent and the other half from second parent, although cross over points can be randomly chosen.

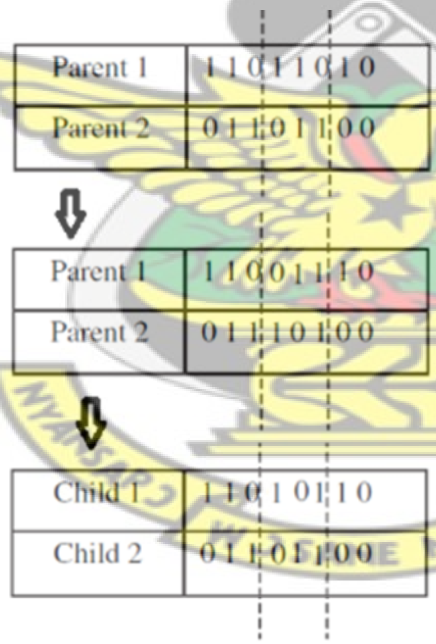
Three Parent Crossover

In this crossover technique, three parents are randomly chosen. Each bit of the first parent is compared with the bit of the second parent. If both are the same, the bit is taken for the offspring otherwise; the bit from the third parent is taken for the offspring.

Parent 1	1 1 0 1 0 0 0 1
Parent 2	0 1 1 0 1 0 0 1
Parent 3	0 1 1 0 1 1 0 0
Child	0 1 1 0 1 0 0 1

Shuffle Crossover

- Shuffle crossover is related to uniform crossover.
- A single (or multiple) crossover position is selected. But before the variables are exchanged, they are randomly shuffled in both parents.
- This removes positional bias as the variables are randomly reassigned each time crossover is performed



Precedence Preservative Crossover (PPX)

- PPX was independently developed for vehicle routing problems.
- For instance, let's consider six operations A–F.

- Parent1 = (A, B, C, D, E, F), and Parent2 = (C, A, B, F, D, E)
- The operator works as follows:
- A vector representing the number of operations is randomly filled with elements of the set 1, 2; 1 for parent1 and 2 for parent2.
- In this strategy, parents and offsprings are permutation lists: no redundancy.
- First we start by initializing an empty offspring.
- We start selecting operations (alleles) from parents following the selected random vector.
- The selected operation is appended to the offspring.
- After an operation is selected, it is deleted in both parents.
- The step is repeated until both parents are empty and the offspring contains all operations involved.

Parent Permutation 1	A	B	C	D	E	F
Parent Permutation 2	C	A	B	F	D	E

Select Parent no. (1/2)	1	2	1	1	2	2
Offsring Permutation	A	C	B	D	F	E

Ordered Crossover

- Given two parent chromosomes, two random crossover points are selected partitioning them into a left, middle and right portion.
- Offspring 1 inherits its left and right section from parent 1.
- The middle section is determined by the genes in the middle section of parent 1 in the order in which the values appear in parent 2.

- A similar process is applied to determine child 2.

Parent 1:	4	2	1	3	6	5	Child 1:	4	2	3	1	6	5
Parent 2:	2	3	1	4	5	6	Child 2:	2	3	4	1	5	6

Partially-Matched Crossover (PMX)

- In Partially Matched Crossover, two strings are aligned, and two crossover points are selected uniformly at random along the length of the strings.
- The two crossover points give a matching selection.
- The crossover is performed as position-by-position exchange operations.

Parent A	4	8	7	3	6	5	1	10	9	2
Parent B	3	1	4	2	7	9	10	8	6	5

- Consider the above example, two crossover points were selected at random: positions 4 and 6.
- Therefore, the genes are exchanged: the 3 and the 2, the 6 and the 7, the 5 and the 9 exchange places. So, child A looks like:

Parent A	4	8	7	3	6	5	1	10	9	2

- Child B also is formed as by exchanging the same genes in parent B:

Parent B	3	1	4	2	7	9	10	8	6	5

- Therefore, each offspring contains ordering information partially determined by each of its parents.
- PMX can be applied to problems with permutation representation.

Crossover Probability (P)

- Crossover probability is a parameter to describe how often crossover will be performed.
- If crossover probability is 100%, then all offspring are made by crossover.
- If it is 0%, whole new generation is made from exact copies of chromosomes from old population
- In this case, the new generation is not always the same as the old generation. Mutation can play a role in modifying the old generation.
- Usually, it is good to leave some old population to next generation. This can be set by selecting a crossover probability less than 100%.

3.6 MUTATION

Premature convergence is a critical problem in most optimization techniques, consisting of populations, which occurs when highly fit parent chromosomes in the population breed many similar offspring in early evolution time. Crossover operation of genetic algorithms (GAs) cannot generate quite different offspring from their parents because the acquired information is used to crossover the chromosomes. An alternate operator, mutation, can search new areas in contrast to the crossover. Crossover is referred as exploitation operator whereas the mutation is exploration one. Like crossover, mutation can also be performed for all types of encoding techniques. After crossover, the strings are subjected to mutation. If crossover is supposed to exploit the current solution to find better ones, mutation is supposed to help for the exploration of the whole search space. Mutation helps to maintain genetic diversity in the population. Mutation introduces new genetic structures in the population. Mutation prevents the algorithm to be trapped in a local minimum. Mutation ensures ergodicity: A search space is said to be ergodic if there is a non-zero probability of generating an optimal solution from any population state. There are many different forms of mutation for the different kinds of representation.

3.6.1 Flipping

- Flipping of a bit involves changing 0 to 1, and 1 to 0 based on a mutation chromosome generated.

Parent	**1**	0	1	1	**0**	1	0	**1**
Mutation Chromosome	**1**	0	0	0	**1**	0	0	**1**
Child	**0**	0	1	1	**1**	1	0	**0**

- For a 1 in mutation chromosome, the corresponding bit in parent chromosome is flipped (0 to 1 and 1 to 0) and child chromosome is produced

3.6.2 Interchanging

Two random positions of the string are chosen and the bits corresponding to those positions are interchanged

Parent	1	**0**	1	1	0	*1*	0	1
Child	1	**1**	1	1	0	**0**	0	1

3.6.3 Reversing

A random position is chosen and the bits next to that position are reversed.

Parent	1	0	1	1	0	1	0	1
Child	1	0	1	1	0	1	1	0

3.6.4 Mutation Probability (P_m)

- The mutation probability decides how often parts of chromosome will be mutated.
- If mutation probability is 100%, whole chromosome is changed. If it is 0%, nothing is changed.
- Mutation should not occur very often, because then GA will in fact change to random search.
- The mutation probability concerns the number of individuals that can be mutated not the number of chromosomes inside one individual

3.7 REPLACEMENT

Replacement is the last stage of the breeding cycle. After selecting two parents from a fixed size population (N), usually they produce two offspring after crossover and mutation. Now with 4 individuals (2 old and 2 new), only 2 of them should be considered for the next GA iteration. GA convergence is greatly affected by the technique used to decide which individual stay and which are replaced. Basically, there are two kinds of methods for maintaining the population; generational updates and steady state updates. The basic generational update scheme consists in producing N children from a population of size N to form the population at the next time step (generation). So, the new population of children completely replaces the parents. Clearly this kind of update implies that an individual can only reproduce or crossover with individuals from the same generation. In a steady state update, new individuals are inserted in the population (2N) as soon as they are created. The insertion of a new individual usually necessitates the replacement of another population member. The individual to be deleted can be chosen as the worst member of the population, or as the oldest member of the population. This will lead to a very strong selection pressure. Another alternative is to replace the most similar member in the existing population.

3.7.1 Random Replacement

The children replace two randomly chosen individuals in the population. This can be useful for continuing the search in small populations.

3.7.2 Weak Parent Replacement

In weak parent replacement, a weaker parent is replaced by a strong child. With the 4 individuals only the fittest two, parent or child, return to population. This process improves the overall fitness of the population when paired with a selection technique that selects both fit and weak parents for crossing; otherwise, a weak individual will never be replaced.

3.7.3 Both Parents

The child replaces the parent. In this case, each individual only gets to breed once. This leads to a problem when combined with a selection technique that strongly favors fit parents: Those parents (the best fit) are disposed of.

3.8 SEARCH TERMINATION(CONVERGENCE CRITERIA)

We shall be looking at the salient point at which all the searches would converge. By this we mean the criterion followed for the algorithm to terminate and produce the desired results. The various stopping condition are listed as follows:

Maximum generations: The genetic algorithm stops when the specified number of generations is have evolved (e.g., 100 generations).

Elapsed time: The genetic process will end when a specified time has elapsed (e.g., 60 minutes).

No change in fitness: The genetic process will end if there is no change to the population's best fitness for a specified number of generations (e.g., after 100 generations, the best fit is the same).

Stall generations: The algorithm stops if there is no improvement in the objective function for a sequence of consecutive generations (e.g., nearly fixed fitness for 100 generations).

Stall time limit: The algorithm stops if there is no improvement in the objective function during an interval of time (e.g., nearly fixed fitness for 60 minutes). Convergence value: The algorithm stops if a specific individual (or individuals) have its fitness less than a convergence value (e.g., convergence value = 10). Next are methods that use the convergence value:

3.8.1 Best Individual

A best individual convergence criterion stops the search once the maximum fitness in the population drops below the convergence value. This brings the search to a faster conclusion guaranteeing at least one good solution.

3.8.2 Worst individual

Worst individual terminates the search when the least fit individuals in the population have fitness less than the convergence value. This guarantees the entire population to be of minimum standard.

3.8.3 Sum of Fitness

The search is stopped when the sum of the fitness in the entire population is less than or equal to the convergence value in the population record. This guarantees that virtually all individuals in the population will be within a particular fitness range.The population size has to be considered while setting the convergence value.

3.8.4 Median Fitness

Here at least half of the individuals will be better than or equal to the convergence value. This should give a good range of solutions to choose from.

3.9 WHEN TO USE GENETIC ALGORITHM

There are two requirements that must be met before an evolutionary algorithm can be used for a particular problem. Firstly, we need a way to encode candidate solutions to the problem. The simplest encoding, and that used by many genetic algorithms, is a bit string. Each candidate is simply a sequence of zeros and ones. This encoding makes cross-over and mutation very straightforward, but that does not mean that you cannot use more complicated representations. In fact, we will see several instances of more advanced candidate representations in later chapters. As long as we can devise a scheme for evolving the candidates, there really is no restriction on the types that we can use. Genetic programming (GP) is a good example of this. GP evolves computer programs represented as syntax trees.

The second requirement for applying evolutionary algorithms is that there must be a way of evaluating partial solutions to the problem - the fitness function. It is not sufficient to evaluate solutions as right or wrong, the fitness score needs to indicate how right or, if your glass is half empty, how wrong a candidate solution is. So a function that returns either 0 or 1 is useless. A function that returns a score on a scale of 1 - 100 is better. We need shades of grey, not just black and white, since this is how the algorithm guides the random evolution to find increasingly better solutions. Evolutionary algorithms are typically used to provide good approximate solutions to problems that cannot be solved easily using other techniques. Many optimisation problems fall into this category. It may be too computationally-intensive to find an exact solution but sometimes a near-optimal solution is sufficient. In these situations evolutionary techniques can be effective. Due to their random nature, evolutionary algorithms are never guaranteed to find an optimal solution for any problem, but they will often find a good solution if one exists.

One example of this kind of optimisation problem is the challenge of timetabling. Schools and universities must arrange room and staff allocations to suit the needs of their curriculum. There are several constraints that must be satisfied. A member of staff can only be in one place at a time, they can only teach classes that are in their area of expertise, rooms cannot host lessons if they are already occupied, and classes must not clash with other classes taken by the same students. This is a combinatorial problem and known to be NP-Hard. It is not feasible to exhaustively search for the optimal timetable due to the huge amount of computation involved. Instead, heuristics must be used. Genetic algorithms have proven to be a successful way of generating satisfactory solutions to many scheduling problems.

Evolutionary algorithms can also be used to tackle problems that humans don't really know how to solve. An EA, free of any human preconceptions or biases, can generate surprising solutions that are comparable to, or better than, the best human-generated efforts. It is merely necessary that we can recognise a good solution if it were presented to us, even if we don't know how to create a good solution. In other words, we need to be able to formulate an effective fitness function.

Engineers working for NASA know a lot about physics. They know exactly which characteristics make for a good communications antenna. But the process of designing an antenna so that it has the necessary properties is hard. Even though the engineers know what is required from the final antenna, they may not know how to design the antenna so that it satisfies those requirements. NASA's Evolvable Systems Group has used evolutionary algorithms to successfully evolve antennas for use on satellites. These evolved antennas have irregular shapes with no obvious symmetry (one of these antennas is pictured below). It is unlikely that a human expert would have arrived at such an unconventional design. Despite this, when tested these antennas proved to be extremely well adapted to their purpose.

3.10 BUILDING BLOCK HYPOTHESIS AND THE SCHEMA THEOREM

A near-optimal performance through the juxtaposition of short, low-order, high-performance schemata, called the building blocks (BBs), is sought by a genetic algorithm. One of the most important criteria for how a genetic algorithm works has been the building block hypothesis. The following six conditions for a genetic algorithm success have been proposed (Goldberg, Deb and Clark, 1992): Identify GAs which are the processing-building blocks; ensure an adequate initial supply of raw BBs; ensure growth of superior BBs; ensure the mixing of BBs; ensure good decisions among competing BBs and solve problems with bounded BB difficulty. Making sure that the GA is well supplied with a sufficient supply of the BBs required to solve a given problem is one of the important conditions. It is worthy to ensure also that proportion of the good ones in the population grow. The usual approach in guaranteeing the increase in market share of good BBs in a population is the schema theorem (Holland and Dejong (1997)).

Holland's Schema Theorem(Holland 1975)us classically given as follows. Let H represent a particular schema as defined by Holland(1975),
L be the length of the chromosome, and $L(H) \leq L-1$ be the defining length of the schema;
$p(H) = \sum_{x \in H} p(x)$ be the frequency of schema H in the population, and $\bar{u}(H) = \sum_{x \in (H)} u(x)p(x)/p(H)$ be the marginal fitness of schema H.

Theorem 3 (The Schema Theorem, Holland 1975) In a genetic algorithm using a proportional selection algorithm and single point crossover occurring with probability r, the following holds for each schema H:

$$p(H)^1 \geq p(H)\frac{\bar{u}(H)}{\bar{u}}(1 - r\frac{L(H)}{L-1})$$

Now, Price's Theorem can be used to obtain the Schema Theorem by using:

$$F(x,(H)) = \begin{cases} 1 & \text{if } x \in H \\ 0 & \text{if } iH \end{cases}$$

and $\theta(y,z,H) = \sum_x F(x,H)T(x \leftarrow y,z)$. which represent the fraction of offspring of parents y and z that are in schema H. Then $p(H) = \bar{F}(H)$, and

Corollary 4(Schema Frequency Change)

$$p(H^1) = \bar{\theta}(H) + Cov[\theta(y,z,H), w(y)w(z)/\bar{w}^2]$$

Two sources can be seen to contribute to a change in Schema frequency:

1. linkage disequilibrium, i.e. the schema frequency minus the product of

the frequencies of the alleles comprising the schema.Negative linakage disequilibrium would produce$\bar{\theta}(H) > p(H)$; and

2. covariance between parental fitnesses and the proportion of the offspring in the schema.

This equation can be made more informative by rewriting $\theta(y, z, H)$ in terms of a "disruption" coefficient. A value $\alpha_H \in [0, 1]$ can be defined that places a lower bound on the faithfulness of transmission of any Schema H:

$$\theta(\mathbf{y}, z, H) \geq \frac{1}{2}(1 - \alpha_H)[F(\mathbf{y,H}) + F(z, H)]$$

and $\alpha_H = 1 - \min_{yinH \text{ or } zinH} \left[\theta(y, z, H) \frac{2}{F(y, H) + F(z, H)} \right]$

Actually, α can be defined for any subset of the search space ("predicate" in Vose(1991) or "forma" in Radcliffe(1991)). For Holland schemata under single-point crossover,$\alpha_H = rL(H)/(L - 1)$ (the rate that crossover disrupts schema H). Using (12) we obtain:

Theorem 4(Schema,Covariance Form)

The change in the frequency of any subset H of the search space(i.e. a schema)over one generation in bounded below by:

$$p(H)^1 \geq \{p(H) + Cov[F(\mathbf{y}, H), w(x)\sqrt{w}]\}(1 - \alpha_H).$$

Therefore, if

$$Cov[F(\mathbf{y,}H), \frac{w(x)}{w}] > \frac{\alpha_H}{1 - \alpha_H}$$

then schema H will increase in frequency.

Proof.

$$\begin{aligned}
\bar{F}(H^1) &= \sum_{x,y,z} F(x,H)T(x \leftarrow y,z)\frac{w(y)w(z)}{\bar{w^2}}p(y)p(z) \\
&= \sum_{y,z} \theta(y,z,H)\frac{w(y)w(z)}{\bar{w^2}}p(y)p(Z) \\
&\geq \frac{1}{2}(1-a_H)\sum_{y,z} F[F(y,H)+F(z,H)]\frac{w(y)w(z)}{\bar{w^2}}p(y)p(z) \\
&= (1-\alpha_H)\sum F(y,H)w(y)p(y)\sqrt{w} \\
&= (1-\alpha_H)[\bar{F}(H)+Cov(y,H),w(x)\sqrt{w}]
\end{aligned}$$

Thus, if there is a great enough covariance between fitness and being a member of a schema, the schema will increase in frequency.
Although both application of price's Theorem - to schema frequency change and change in the fitness distribution-involve covariances with parental fitness values, the crucial point is that the covariance term(from(13)), $Cov[F(y,H),w(x)/\bar{w}]$, and the covariance term(from(7)), $Cov[\theta(y,z,w),w(y)w(z)\sqrt{w}^2]$, are independently defined. So conditions that produce growth in the frequencies of different schemata are independent of conditions that produce growth in the upper tails of the fitness distribution.

For example, consider a fitness function with a random distribution being the one-sided stable distribution of index $\frac{1}{2}$(Feller, 1971):R(w)=2$N(a/\sqrt{w})-1$, where $N(y)$ is the Normal distribution and a is the scale parameter. This distribution is a way of generating “needles in the haystack” on all length scales. A GA with this fitness function will generically have schemata that obey(10), even though it is still random search.

3.11 NO FREE LUNCH THEOREM

In this section we analyze the connection between algorithms and cost functions. We have dubbed the associated results "No Free Lunch"(NFL)theorems because they demonstrate that if an algorithm perform well on a certain class of problems then it necessarily pays for that with degraded performance on the set of all remaining problems. Additionally, the name emphasizes the parallel with similar results in supervised learning[Wo196a, Wo196b].

The precise question addressed in this section is: "How does the set problems $F_1 \subset F$ for which algorithm a_1 performs better than algorithm a_2 compare to the set $F_2 \subset F$ for which the reverse is true?" To address this question we compare the sum over all f of $P(d_m^y|f, m, a)$ is independent of a when we average over all cost functions:

Theorem 1 *for any pair of algorithms* a_1 and a_2,

$$\sum_f P(d_m^y|f, m, a_1) = \sum_f P(d_m^y|f, m, a_2).$$

An immediate corollary of this result is that for any performance measure $\Phi(d_m^y)$, the average over all f of $P(\Phi(d_m^y)|f, m, a)$ is independent of a. The precise way that the sample is mapped to a performance measure is unimportant.

This theorem explicitly demonstrate that what an algorithm gains in performance on one class of problems it necessarily pays for on the remaining problems; that is the only way that all algorithms can have the same f-averaged performance.

A result analogous to Theorem 1 holds for a class of time-dependent cost functions.

The time- dependent functions we consider begin with an initial cost function f_1 that is present at the sampling of the first z value.

Before the beginning of each subsequent iteration of the optimisation algorithm, the cost function is deformed to a new function, as specified by a mapping $T : F \times N \to F^2$ We indicate this mapping with the notation T_1, . SO the function present during the ith iteration is $f = T^1(f_1).T_1$ is assumed to be a (potentially i-dependent)bijection between F and F. We impose bijectivity because if it did not hold,the evolution of cost functions could narrow it on a region of f's for which some algorithms may perform better than others. This would constitute an an priori bias in favor of those algorithms, a bias whose analysis we wish to defer to future work.

How best to assess the quality of an algorithm's performance on time-dependent cost functions is not clear. Here we consider two schemes based on manipulations of the definition of the sample. In scheme 1 the particular Y value in $d^y_m(j)$ corresponding to a particular z value $d^y_m(j)$ is given by the cost function that was present when $d^y_m(j)$ was sampled. IN contrast, for scheme 2 we imagine a sample D^y_m given by the Y values from the present cost function for each of the z values in d^y_m. Formally if $d^y_m = d^y_m(1), ..., d^y_m(m)$, then in scheme 1 we have $d^y_m = f_1(d^y_m(1)), ..., T_{m-1}(f_{m-1})(d^y_m(m))$, and in scheme 2 we have $D^y_m = f_m(d^y_m(1), ..., f_m(d^y_m(m)), ...,$ where $f_m = T_{m-1}(f_{m-1})$ is the final cost function.
In some situations it may be that the members of the sample "live" for a long time, on the time scale of the evolution of the cost function. In such situations it may be appropriate to judge the quality of the search algorithm by D^y_m; all those previous elements of the sample are still "alive" at time m, and therefore their current cost is of interest. On the other hand, if members of the sample live for only a short time on the time scale of evolution of the cost function, one may instead be concerned with things like how well the "living" member(s) of the sample track the changing cost function. In such situations, it may make more sense to judge the quality of the algorithm with the d^y_m sample,

Results similar to Theorem 1 can be derived for both schemes. By analogy with that theorem, we average cost over all possible ways a cost function may be time-dependent, i.e., we average over all T(rather than over all f). Thus we consider $\sum rP(d_m^y|f_1T, m, a)$ where f_1 is the initial cost function. Since T only takes effect for $m > 1$, and since f_1 is fixed, there are priori distinctions between algorithms as far as the first member fo the popultation is concerned. However after redefining samples to only contain those elements added after the first iteration of the algorithm.

Theorem 2

For all $d_m^y, D_m^y, m > 1$,algorithmsa_1 and a_2, and initial cost functions f_1,

$$\sum_r p(d_m^y|f_1, T, m, a) = \sum P(d_m^y|f_1, T, m, a)$$

and

$$\sum_r p(D_m^y|f_1, T, m, a_1) = \sum p(D_m^y|f_1, T, m.a_2.)$$

So in particular, if one algorithm outperforms another for certain kinds of evolution operators, then the reverse must be true on the set of all other evolution operators.

Although this particular result in similar to the NFL result for the static case, in general the time-dependent situation is more subtle. In particular, with time-dependence there are the time-dependent situation is more subtle. In particular, with time-dependence there are situations in which there can be a priori distinctions between algorithms even for those members of the population arising after the first. For example, in general there will be distinctions between algorithms when considering the quantity $\sum_f P(d^y m|f, T, m, a)$.To see this, consider the case where X is a set of contiguous integers and for all iterations T is a shift operator, replacing $f(x)$ by $f(x-1)$ for all x(with min $(x) - 1 \equiv max(x)$). For such a case we can construct algorithms which behave differently a priori. For example, take a to be the algorithm that first samples f at x_1,next at $x_1 + 1$

and so on, regardless of the values in the population. Then for any f, d_m^y is always made up of identical$Y$ values. Accordingly, $\sum_f P(d_m^y|f, T, m, a)$ is non-zero only for d_m^y for which all values $d_m^y(i)$ are identical. Other search algorithms, even for the same shift T, do not have this restriction on Y values. This constitutes an a priori distinction between algorithms.

3.12 COMPARISON WITH OTHER OPTIMIZATION TECHNIQUES

GAs operate with coded versions of the problem parameters; the coding of solution set and not the solution. Almost all conventional optimization techniques search from a single point (centralized) but, Gas always operate on a whole population of points (distributed). It improves the chance of reaching the global optimum and also helps in avoiding local stationary point. GA uses fitness function for evaluation rather than derivatives. As a result, they can be applied to any kind of continuous or discrete optimization problem. GAs use probabilistic transition operates while conventional methods for continuous optimization apply deterministic transition operates i.e., GAs do not use deterministic rules.

3.13 FLOYD-WARSHALL ALGORITHM

The weight of an edge between vertex i and vertex j in graph $G = (V, E)$ where

$$w_{ij} = \begin{cases} 0 & \text{if } i = j \\ \text{the weight of a directed edge}(i, j) & \text{if } i \neq j \text{ and } (i, j) \in E \\ \infty & \text{if } i \neq j \text{ and } (i, j) \notin E \end{cases}$$

- A shortest path does not contain the same vertex more than once
- For a shortest from i to j such that any intermediate vertices on the path are chosen from the set $\{1, 2, , \cdots, k\}$, there are two possibilities:

1. k is not a vertex on the path, so the shortest such path has length d_{ij}^{k-1}

2. k is a vertex on the path, so the shortest such path has length $d_{ik}^{k-1} + d_{kj}^{k-1}$

- So we see that we can recursively define $d_{ij}^{(k)}$ as

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if} k = 0 \\ \min(d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1}) & \text{if } k \geq 1 \end{cases}$$

The Floyd-Warshall Algorithm

Floyd- Warshall (W)

$n = W.\ rows$

$D^{(0)} = W$

for k=1 to n

let $D^{(k)} = (d_{ij}^{k-1})$ *be a new matrix*

for i=1 to n

for j=1 to n

$d_{ij}^{k-1} = \min(d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1})$

return $D^{(n)}$

3.14 FLOW CHART SHOWING GENETIC APPLICATION TO TSP

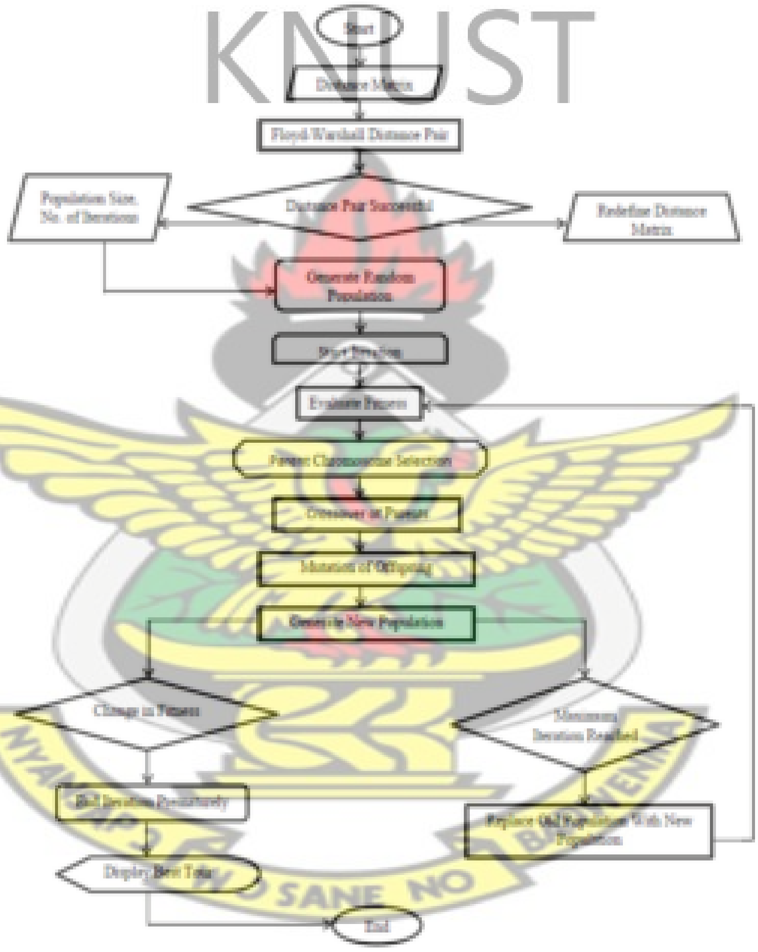


Figure 3.8: flow chart for Floyd – Warshall

CHAPTER 4

DATA COLLECTION AND ANALYSIS

4.1 INTRODUCTION

This chapter shall be devoted to data collection and analysis. We shall determine the optimum route for the selected tourist sites in the Kwahu zone of Ghana. We shall also analyze the data using the MATLAB (Matrix Laboratory) program. The pseudocode for genetic algorithm shall be implemented to establish the optimal route. In this study, the grid points of any tourist sites shall be taken.

4.2 GENETIC ALGORITHM MODEL FOR TRAVELLING SALESMANPROBLEM (TSP)

The travelling salesman problem has to do with a salesman travelling n locations and returns to the starting location with the minimal cost, with a very low or no probability of crossing the location on more than one occasion. Genetic algorithm (GA) could be used to solve the travelling salesman problem (TSP) because the locations are random. The most important aspect is to find the shortest distances between N different tourist sites. Tour is the path that a salesman takes. The movements of tourists have always been under scrutiny. The movements of tourists could be modelled as a Travelling Salesman Problem (TSP). Data on the grid points of the locations was collected to create a set of routes the tourists shall use to minimize the overall travelling distance of the tourists. Testing every possibility for N location tour shall be N!. This implies testing 20 locations, we would have had to measure 20! = 2432902008176640000 different tours, which would have taken years. In any case, genetic algorithm can be used to find a solution in the shortest possible time, although it might not find the best solution, it can find a near perfect solution for a 100 city tour in less than a minute. There are couples of basic steps to follow in solving the travelling salesman problem using Genetic Algorithm.

4.3 DATA AND ENCODING

The Tourist centres to be considered are; Our lady of good counsel, Crucifixion of Christ, Peace reconciliation center, Joseph & Jesus shop, Angel statue, Butuase waterfalls, Buruku shrine, Obuo da buo so (kutuso), Stone chair, River Afram, Alligator Rock, Gate-way rock of stone city, Head on shoulder rock, Paragliding centre, obuo da buo so (Nkeetepa), Odweanoma shrine, Akyeakyewaa falls, Highest habitable point in Ghana and the Abetifi shrine & national monument.

We shall use MATLAB to plot the graph of all the cities using their local grid points (coordinates)

Table 4.1: Coordinates of the tourist sites

SITES	SITES NAMES	X	Y
A	Odwenanoma Shrine	747079	727406
B	Paragliding Center	747405	727400
C	Abetifi Shrine & National Monument	748944	738027
D	Highest Habitable Point -Abetifi	748827	738850
E	Akyeakyewaa Falls	750685	740628
F	Oku Abena Water Falls	753883	737585
G	Stone Chair	753682	737396
H	Obuo Da Buo So (Kotoso)	760272	743910
I	Buruku Shrine	762176	740135
J	River Afram	774954	743076
K	Obuo Da Buo So (Nketepa)	784008	738941
L	Head on Shoulder Rock	783993	738917
M	Alligator Rock	786860	736105
N	Gate-Way Rock Of Stone City	786596	735301
O	Catholic Shrine	758049	736032
P	Butuase Water Falls	758806	736997

Table 4.2: Distance square matrix of all sites in kilometres

SITES	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
A	-	**0.3**	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
B	0.3	-	17		∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
C	∞	17	-	1.5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
D	∞	∞	1.5	-	7	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
E	∞	∞	∞	7	-	7	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
F	∞	∞	∞	∞	7	-	1	∞	∞	∞	∞	∞	∞	∞	∞	∞
G	∞	∞	∞	∞	∞	1	-	15	∞	∞	∞	∞	∞	∞	∞	∞
H	∞	∞	∞	∞	∞	∞	15	-	7	∞	∞	∞	∞	∞	∞	∞
I	∞	∞	∞	∞	∞	∞	∞	7	-	29		∞	∞	∞	∞	∞
J	∞	∞	∞	∞	∞	∞	∞	∞	29	-	25	∞	∞	∞	∞	∞
K	∞	∞	∞	∞	∞	∞	∞	∞	∞	25	-	0.5	∞	∞	∞	∞
L	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0.5	-	0.4	∞	∞	∞
M	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0.4	-	2	∞	∞
N	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	2	-	40	∞
O	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	40	-	3.5
P	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	3.5	-

We work towards finding a set of least cost routes such that all tourist sites are visited. The Floyd-Warshall's algorithm to estimate the approximate distances between the sites since we do not have to work with infinitesimal values in the distance matrix.

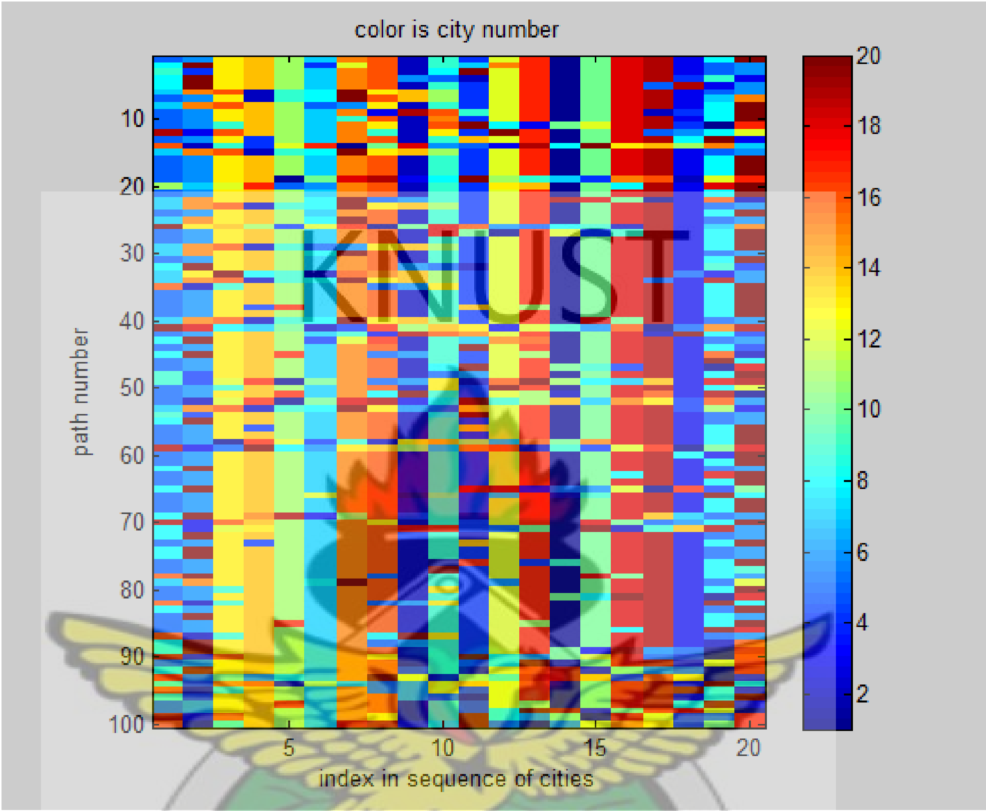


Figure 4.1: A plot of the corresponding distance square matrix using MATLAB

Table 4.3: Distance square matrix of all sites in kilometres using Floyd-Warshall

SITES	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
A	0	0.3	17.3	18.8	25.8	32.8	33.8	46.8	53.8	80.8	102.8	103.3	103.7	105.9	145.7	147.2
B	0.3	0	18.8	25.8	32.8	33.8	46.8	53.8	80.8	102.8	103.3	103.7	105.9	145.7	147.2	147.2
C	17.3	18.8	0	1.5	8.5	15.5	16.5	29.5	36.5	63.5	85.5	86	86.4	88.4	128.4	129.9
D	18.8	25.8	1.5	0	7	14	15	28	35	62	84	84.5	84.9	86.9	126.9	128.4
E	25.8	32.8	8.5	7	0	7	8	21	28	55	77	77.5	77.9	79.9	119.9	120.4
F	32.8	33.8	15.5	14	7	0	1	14	21	48	70	70.5	70.9	72.9	112.9	114.4
G	33.8	33.8	16.5	15	8	1	0	13	20	47	69	69.5	69.9	71.9	111.9	113.9
H	46.8	53.8	29.5	28	21	14	13	0	7	34	56	56.5	56.9	58.9	98.9	100.4
I	53.8	80.8	36.5	35	28	21	20	7	0	27	49	49.5	49.9	51.9	91.9	92.4
J	80.8	102.8	63.5	62	55	48	47	34	27	0	22	22.5	22.9	24.9	64.9	66.4
K	102.8	103.3	85.5	84	77	70	69	56	49	22	0	0.5	0.9	2.9	42.9	44.4
L	103.3	103.7	86	84.5	77.5	70.5	69.5	56.5	49.5	22.5	0.5	0	0.4	2.4	42.4	43.9
M	103.7	105.9	86.4	84.9	77.9	70.9	69.9	56.9	49.9	22.9	0.9	0.4	0	2	42	43.5
N	105.9	145.7	88.4	86.9	79.9	72.9	71.9	58.9	51.9	24.9	2.9	2.4	2	0	40	41.5
O	145.7	147.2	128.4	126.9	119.9	112.9	111.9	98.9	91.9	64.9	42.9	42.4	42	40	0	1.5
P	147.2	147.2	129.9	128.4	120.4	114.4	113.9	100.4	92.4	66.4	44.4	43.9	43.5	41.5	1.5	0

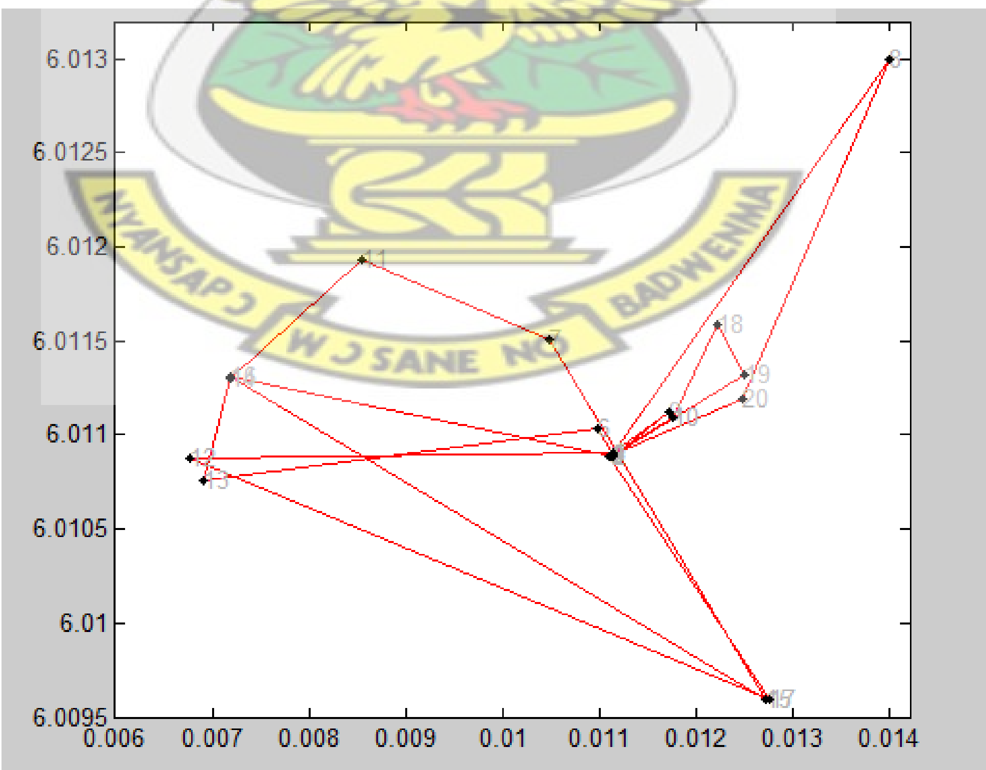


Figure 4.2: Graph showing the optimal route (tour) using MATLAB code

The optimal tour route under the study is as illustrated below:

Odwenanoma Shrine → Paragliding Center → Abetifi Shrine & National Monument → Highest Habitable Point (Abetifi) → Akyeakyewaa Falls → Oku Abena Water Falls → Stone Chair → Obuo Da Buo So (Kotoso) → Buruku Shrine → River Afram → Obuo Da Buo So (Nketepa) → Head on Shoulder Rock → Alligator Rock → Gate-Way Rock Of Stone City → Catholic Shrine → Butuase Water Falls.

The optimal distance under the selected tourist sites is 147.2km.

4.4 FINDINGS

The study has obtained the optimal route for touring the sites under study. The roads within this route should be given the much needed priority in terms of development and maintenance if tourism in the Kwahu zone is to be improved. The paragliding centre and Butuase Water Falls are possible starting points. Accommodation facilities at Adawso could be improved or established since they can also serve as stop overs. Apart from the paragliding centre, Adawso could also have served as a perfect starting point for tourists. Using the sites in issue as priority areas shall be a major cost cutting adventure. Touring sites for leisure is quite an expensive exercise and therefore minimizing the cost of such a project shall automatically save resources. The cost of transportation is not fixed and even differs from location to location.

4.5 SUMMARY

This chapter presented the data collected and the analysis of the data obtained under the study. The next, which is the final chapter of the study shall focus on the conclusions and recommendations of the study.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 INTRODUCTION

This chapter presents conclusions and recommendations of the study. The main objective of the study was to analyze and determine the optimal route for selected tourist centres in the Kwahu zone of Ghana. The objective stemmed from the fact that tourism is a very viable economic venture and cost associated with it should not be very high. The major tourist sites were captured in a network and Genetic algorithm used to find the optimal route required to connect the sites.

5.2 CONCLUSIONS

In this study, we exposed the major tourist sites selected in the form of a network and used the Genetic algorithm to determine the optimal route, containing all the tourist centres for which the total distance covered is a minimum. The study revealed that the optimal route to be covered has an approximate distance of One Hundred and Forty – Seven and two – tenth (147.2) kilometres. The study also showed that with this minimum distance, the total expenses on touring these sites would be significantly reduced. This consequently will make resources prudently utilized.

5.3 RECOMMENDATIONS

Since tourism has always being a critical economic exercise for the government and the masses, it is strongly recommended that the roads within the optimal route be given much attention in terms of maintenance. After all, the associated benefits would be much more than the cost incurred. It is also highly recommended that the district assemblies incorporate this in their plans as regards tourism improvements

REFERENCES

Abdelmaguid T.F., amd Dessouky M. M. (2006). A genetic algorithm approach to the integrated inventory distribution problem, Cairo University, Egypt. Daniel J. Epstein Department of Industrial and Systems Engineering, 3715 McClintock Ave, University of Southern California, Los Angeles, CA 90089-0193, USA

Agarwal A. (2012). Secret Key Encryption Algorithm Using Genetic Algorithm. International Journal of Advanced Research in Computer Science and Software Engineering Volume 2, Issue 4, April 2012

Ahmed ElSayed, A. Kongar, E. and Gupta,S. M. (2010). A genetic algorithm approach to end-of-life disassembly sequencing for robotic disassembly. Mechanical and Industrial Engineering Faculty Publications. Paper 10. http://hdl.handle.net/2047/d20000258 by Northeastern University.

Aickelin U. W. E. (1999). Genetic Algorithms for multiple choice optimization problems, Thesis submitted to the University of Wales In candidature for the Degree of Doctor of Philosophy.

Al-Dulaimi B. F. and Ali H. A. (2008). Enhanced Traveling Salesman Problem Solving by Genetic Algorithm Technique (TSPGA).

Altenberg L. (1995). The Schema Theorem and Price's Theorem. In Foundations of Genetic Algorithms 3, ed. Darrell Whitley and Michael Vose. Morgan Kaufmann, San Francisco, pp. 23-49.

Amponsah S. K. and Darkwah K. F. (2007). Function optimization over integral domain: comparative performance of Elite Genetic Algorithm For small iterations and small Generation size. Journal of science and technology, volume 27 no.1

April 2007.

Amponsah S.K., Amoako G., Darkwah F. K. and Agyeman E. (2010). Location of Ambulance Emergency medical service in the Kumasi metropolis, Ghana. African Journal of mathematics and computer science Research vol.4(1),pp.18-26,January,2011.

Barrero F. D., Gonzalez-Pardo, A., Camacho, D. and R-Moreno M. D. (2010). Distributed Parameter Tuning for GeneticAlgorithms. ComSIS Vol. 7, No. 3, June 2010.

Berghman L., Goossens D. and Leus R. (2009). Efficient solutions for Mastermind using genetic algorithms. Computers and Operations Research 36(6), 1880–1885 (2009). URL http://www.scopus.com/inward/record.url?eid=2-s2.0-56549123376&partn

Bermúdez R. and Cole M. H. (2002). A Genetic Algorithm Approach to Door Assignments in Breakbulk Terminals," Working Paper – University of Arkansas, Fayetteville – Department of Industrial Engineering, 2002.

Bhasin H. and Ahuja G. (2012). Harnessing Genetic Algorithm for Vertex Cover Problem. International Journal on Computer Science and Engineering (IJCSE), Vol. 4 No. 02 February 2012

Bhasin H., Kumar N. and Munjal D. (2013). Hybrid Genetic Algorithm for Maximum Clique Problem. International Journal of Application or Innovation in Engineering & Management (IJAIEM) (Volume 2, Issue 4,

Bjarnadóttir Á. S. (2004). Solving the Vehicle Routing Problem with Genetic Algorithms. A thesis submitted to Informatics and Mathematical Modeling, IMM Technical University of Denmark, DTU

Blanco A., Delgado M. and Pegalajar M. C. (1999). A Genetic Algorithm to Obtain the Optimal Recurrent Neural Network. International Journal of Approximate Reasoning 23 (2000) 67-83

Burfield C. (2013). Floyd-Warshall Algorithm. February 20, 2013.

Cantú-Paz E. (2013). A Survey of Parallel Genetic Algorithms. Department of Computer Science and Illinois Genetic Algorithms Laboratory University of Illinois at Urbana-Champaign.

Cavalier T. M. and Joseph H. W. (2012). A Genetic Algorithm for the Split Delivery Vehicle Routing Problem. American Journal of Operations Research, 2012, 2, 207-216

Chakraborty R. C. (2010). Fundamentals of Genetic Algorithms. www.myreaders.info. June 2010.

Chang P. C., Shih-Hsin C. and Kun-Lin L. (2005). Two-phase sub population genetic algorithm for parallel machine-scheduling problem. Expert Systems with Applications 29 (2005) 705–712.

Chaudhry,I. A., Munem D. and Khan A. (2012). Minimizing makespan for a no-wait flowshop using genetic algorithm. National University of Sciences and Technology, H-12 Islamabad 46000, Pakistan .Sadhana Vol. 37, Part 6, December 2012, pp. 695–707.

Dawood A. K. S., Kumar M. R. P. and Saravanakumar K. (2012). Optimization of CNC Turning Process Parameters on INCONEL 718 Using Genetic Algorithm. IRACST – Engineering Science and Technology: An International Journal (ESTIJ), ISSN: 2250-3498, Vol.2, No. 4, August 2012

Dridi H., Kammarti, R., Ksouri, M. and Borne P. (2011). Multi-Objective Optimization for the m-PDPTW: Aggregation Method With Use of Genetic Algorithm and Lower Bounds. Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844 Vol. VI (2011), No. 2 (June), pp. 246-257

Fang H. L. (1994). Genetic Algorithms Timetabling and Scheduling. A Thesis submitted to University of Edinburgh (1994),

Feng-Cheng Y. and Wei-Ting W. (2012). A Genetic Algorithm-Based Method For Creating Impartial Work Schedules For Nurses. International Journal of Electronic Business Management, Vol. 10, No. 3, pp. 182-193 (2012)

Fu Z., Bruce L., Golden B. L., Lele S., Raghavan S. and Wasil E. A. (2003). A Genetic Algorithm-Based Approach for Building Accurate Decision Trees INFORMS Journal on Computing © 2003 INFORMS Vol. 15, No. 1, Winter 2003 pp. 3–22

Fu L., Sun D. and Rilett L. R. (2005). Heuristic shortest path algorithms for transportation applications: State of the art. Computers & Operations Research 33 (2006) 3324–3343.

Gallotta M. (2007). Genetic Algorithms: Colour Image Segmentation, Computer Science Department, University of Cape Town, Private Bag, Rondebosch, 7700, South Africa.

Garkaz M. and Abdollahi A. (2010). The Investigation of Possibility of the Use of Genetic Algorithm in Predicting Companies' Bankruptcy. 2010 International Conference on Business and Economics Research vol.1 (2011) © (2011) IACSIT Press, Kuala Lumpur, Malaysia.

Goletsis Y., Papaloukas C., Fotiadis D. I., Likas A. and Michalis L. K. (2004). Automated Ischemic Beat Classification Using Genetic Algorithms and Multicriteria Decision Analysis. IEEE Transactions on Biomedical Engineering, No 10, Vol. 51, October 2004

Gupta S. M. and McGovern S. M. (2005). A balancing method and genetic algorithm for disassembly line balancing. European Journal of Operational Research (2005)

Gyamfi D. (2013). A Genetic Algorithm Model For Travelling Salesman Problem (TSP). A thesis submitted to the School of Graduate Studies Kwame Nkrumah University of Science and Technology, August 2013.

Herman M. (2011). The Building Block Hypothesis And GA Variants. Informatics Forum 1.42, 04/10/2011.

Keskinidis S. (2002). Pattern Identification of Time Series Events: A Parallel Genetic Algorithm Approach. Faculty of Information Technology Queensland University of Technolgy, November 2002

Konanur P. (2005). Application of Genetic Algorithms to a Multi-Agent Autonomous Pilot for Motorcycles. A Thesis submitted to the faculty of the Indiana University South Bend

Krüger D. and Scholl A. (2007). A heuristic solution framework for the resource constrained multi-project scheduling problem with sequence-dependent transfer times. Working and Discussion Paper Series School of Economics and Business Administration Friedrich-Schiller-University Jena

Kumar V., Murthy A. N. N. and Chandrashekara K. (2009). Scheduling of Flexible Manufacturing Systems Using Genetic Algorithms: A Heuristic Approach. J. Ind. Eng. Int., 7(14), 7-18, Summer 2011. © IAU, South Tehran Branch

Kuo R. J., Tung-Lai H. and Zhen-Yao C. (2009). Evolutionary Algorithm based Radial Basis Function Neural Network for Function Approximation 978-1-4244-2902-8/09/ © 2009 IEEE

Kusiak A. and Shah S. (2006). Cancer gene search with data-mining and genetic algorithms. Intelligent Systems Laboratory, MIE, 2139 Seamans Center, The University of Iowa, Iowa City, IA 52242-1527, USA. Computers in Biology and Medicine 37 (2007) 251 – 261.

Lazar A. and Reynolds R. G. (2003). Heuristic knowledge discovery for archaeologcal data using genetic algorithms and rough sets, Artificial Intelligence Laboratory, Department of Computer Science, Wayne State University, (2003).

Lobo L. M. R. J. and Sharadchandra R. K. (2012). Parallelization of Genetic Algorithm Using Hadoop. International Journal of Engineering Research & Technology (IJERT). Vol. 1 Issue 9, November- 2012

Malhotra R., Singh N. and Singh Y. (2011). Genetic Algorithms: Concepts, Design for Optimization of Process Controllers. Computer and Information Science, Vol. 4, No. 2; March 2011.

Marian R. M., Luong L. S. H. and Dao S .D. (2011). Modeling and Optimization of Distribution Networks Using Hybrid Genetic Algorithms: A Comparative Study. Proceedings of international Multiconference of Engineers and Computer scientist 2011 vol I, IMECS 2011, March 16-18,2011,Hong Kong.

Maric M. (2008). An Efficient Genetic Algorithm For Solving The Multi-Level Uncapacitated Facility Location Problem Faculty of Mathematics, University of Belgrade Studentski trg 16/IV.

Mulunda C. K., Adede A. O. and Wagacha P. W. (2013). Genetic Algorithm Based Model in Text Steganography. Genetic Algorithm Based Model in Text Steganography. The African journal of information systems Volume 5, Issue 4, October 2013

Noor N. B. T.M (2004). A Study On Genetic Algorithm, A Thesis submitted to the University of Technology MARA.

Parvez W. and Dhar S. (2013). Path Planning Optimization Using Genetic Algorithm. International journal of Computational Engineering Research, Vol,03, Issue 4

Rakkiannan T. and Palanisamy B. (2012). Hybridization of Genetic Algorithm with Parallel Implementation of Simulated Annealing for Job Shop Scheduling. American Journal of Applied Sciences 9 (10): 1694-1705, 2012/ ISSN 1546-9239 © 2012 Science Publication.

Ramirez E .R. (2010). Using Genetic Algorithms To Solve High School Course Timetabling Problems. A Thesis Presented to San Diego State University (2010)

Resende M. G. C, José de Magalhães J. M. and Gonçalves F. J. (2004). A Genetic Algorithm for the Resource Constrained Multi-Project Scheduling Problem AT&T Labs Technical Report TD-668LM4, October 29, 2004.

Sadjadi F. A. (2004). Comparison of fitness scaling functions in genetic algorithms with applications to optical processing. Optical Information Systems II, Proceedings of SPIE Vol. 5557 (SPIE, Bellingham, WA, 2004)

Scaria A. and Mahesh S. (2010). Multi-Objective Flow Shop Scheduling using Genetic Algorithm. Department of Mechanical Engineering, College of Engineering, Trivandrum

Slavov T. and Roeva O. (2012). Application of Genetic Algorithm to Tuning a PID Controller for Glucose Concentration Control. WSEAS TRANSACTIONS on SYSTEMS Issue 7, Volume 11, July 2012

Sodsee (2012). A Multi-Objective Bisexual Reproduction Genetic Algorithm for Computer Network Design. King Mongkut's Institute of Technology North Bangkok World Academy of Science, Engineering and Technology 14 2008

Soni A. and Agrawal S. (2012). Using Genetic Algorithm for Symmetric Key

Generation in Image Encryption. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). Volume 1, Issue 10, December 2012

Süer G. A., Yang X., Alhawari O. I., Santos J. and Vazquez R. (2012). A Genetic Algorithm Approach for Minimizing Total Tardiness in Single Machine Scheduling. International Journal of Industrial Engineering and Management (IJIEM), Vol. 3 No 3, 2012, pp. 163-171

Tuzkaya G., Çaglar E.G., Bildik E. and Gülsün B.(2012). A Hybrid Genetic Algorithm Simulated Annealing Approach for the Multi-Objective Vehicle Routing Problem with Time Windows. Copyright © 2012, IGI Global.

Varshney R. G. and Tyagi N. (2012). A Model To Study Genetic Algorithm For The Flowshop Scheduling Problem. Journal of Information and Operations Management Volume 3, Issue 1, 2012, Pp-38-42.

Wu, Z. Y. and Katta P. (2009). Applying Genetic Algorithm to Geometry Design Optimization: Improving Design by Emulating Natural Evolution. Technical Report, Bentley Systems, Incorporated, Watertown, CT, USA.

www.wikipedia.org

Yahya A. A. A. (2007). Fusion Of Similarity Measures Using Genetic Algorithm For Searching Chemical Database. A project report submitted to the Faculty of Computer Science and Information System, University Technology Malaysia, November 2007