KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

HEURISTIC CROSSOVER FOR PORTFOLIO SELECTION

By

GYAMERAH, ASANTE SAMUEL.

BSC.(Mathematics and Statistics)

A THESIS SUBMITTED TO THE DEPARTMENT OF MATHEMATICS, KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY IN PARTIAL FUFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF PHILOSOPHY (APPLIED MATHEMATICS)

April 23, 2014

Declaration

I hereby declare that this submission is my own work towards the award of the M. Phil 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 acknowledgement had been made in the text.

Gyamerah, Asante Samuel	....	....
Student	Signature	Date

Certified by:

Dr. J. Ackora-Prah	....	....
Supervisor	Signature	Date

Certified by:

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

Dedication

To my beloved parents, Mr. S.K. Gyamerah and Madam Agnes Awuah.

Acknowledgement

To God be the glory for the great things He has done.

I thank God for his guidance throughout this work and for giving me the strength to plod on.

Many thanks to my supervisor, Dr. J. Ackora-Prah, for his guidance and assiduous supervision of this work despite his busy schedule. May his heart desires be granted.

Thanks to the entire staff of the Department of Mathematics, KNUST especially the Head of Department, Prof. S.K. Amponsah and Dr. Peter Yirenchi for the knowledge they impacted in me that brought to bear in the completion of this work. God richly bless you all.

Finally, I cannot hold back my appreciation to my siblings, fellow students, friends, and loved ones especially Markus, Afrifa, Justice, Perpetual and Ruth Sefarkor, who in diverse ways contributed to the success of this work.

Abstract

The appropriate choice of an optimal portfolio is the principal problem of both the portfolio manager and the investor. We propose the suitability of Heuristic Crossover in Genetic Algorithm (GA) for the selection of an optimal portfolio of stocks from the Ghana Stock Exchange. In this book, we formulate a model to include practical constraints (floor-ceiling and cardinality constraints) other than Markowitz unconstrained Mean-Variance model for the selection of our optimal portfolio. We use heuristic crossover as an appropriate solution to optimize the risk-return trade-off and achieve an optimal solution for the portfolio selection and the allocation of weights to each portfolio.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

It is the wish of every investor to minimize the total risk of his/her investment whilst maximizing the return of his/her investment. The total risk of an investment in a portfolio of stocks can be reduced if different stocks are added to the portfolio since different stocks are not perfectly correlated. This means that the change in price of some stocks can compensate changes in the other direction of other stocks. The investor has to decide what stocks to include in the portfolio and in what proportions so as to maximize expected return and to minimize risk. This is called a Portfolio Selection Problem. The basis of Portfolio Selection is choosing a weighted group of assets from a large number of available securities so as to maximize the expected return to a given risk rate. We design an objective function f taking different factors into account: performance, risk, diversification and eventually more. f reflects our investment preferences.
Our Portfolio is the solution to the following optimization problem:

$$\text{minimize} \quad f(x), \; x \in \alpha$$

where α belongs to all possible solutions of the Portfolio.

Markowitz(1952) proposed the *Modern Portfolio Theory,(MPT)*, which states that by choosing a combination of assets to invest in, an investor could get higher returns with the same risk rate. Markowitz developed the theory of Optimal Portfolio Selection which was actually based on the investor's risk and preference. This theory is called the Mean-variance model, which is sometimes seen

as either minimizing an objective function of the Portfolio Variance (risk) for calculated estimate of return (profit) or maximizing an objective function of the Portfolio return for a given level of Portfolio Variance.
Markowitz theory has helped in the growth of modern financial theory especially portfolio optimization during the past four (4) decades. This theory made Harry Markowitz to be awarded the Nobel Prize in Economics in 1990. The underlying focus of Markowitz model on a Portfolio is to take the expected return of a Portfolio as the Investment return and the variance of the expected returns of a portfolio as the investment risk. According to Markowitz, for any specific return rate, the minimum investment risk can be derived by minimizing the variance of the Portfolio; or for any given risk level which the investor can tolerate, the maximum returns can be derived by maximizing the expected returns of a Portfolio.

Markowitz theory, however, does not account for certain real life market constraints like the cardinality constraint (the restrictions of a portfolio to a certain number of assets) and the boundary constraints (the ceil and the floor constraints). These constraints help in the practical selection of a Portfolio. His model is also computationally intensive since it considers the covariance of the returns between every pair of stocks in a Portfolio.

In order to help curtail for the above limitations of Mean-Variance model of Markowitz, Genetic Algorithm (GA) is been implemented in this study. GA tries to provide a solution to difficult problems through an iterative process which is based on Darwin's theory of evolution called Natural Selection/Survival of the fittest. Darwin's theory states that individuals with certain favourable traits are more likely to survive and reproduce thereby passing on certain traits to their offspring's whilst individuals with less favourable traits will gradually fade out from the population. John Holland and his colleagues invented GA in the early 1970's. In GA, the population comprises of a set of solutions instead of chromosomes. The characteristics of GA that distinguishes them from other heauristics are the following:

1. GA works with coding of the solutions instead of the solutions themselves. It therefore requires a good, efficient representation of the solutions in the form of a chromosome.

2. GA are nondeterministic, thus they are stochastic in decisions, which accounts for their robustness.

3. GA only require objective function values.

4. They search from a set of solutions, which is different to other metaheuristics like Tabu search and Simulated annealing that starts with a single solution and move to another solution by some transition. They therefore do a multi directional search in the solution space, which reduces the probability of been caught in a local optima.

1.2 Statement Of The Problem

The apt choice of a Portfolio in order to maximize the expected returns to a given risk rate is sometimes challenging in the field of finance. Moreover, the selection of a suitable weight for an asset from a large number of securities builds up another problem in the whole process.

Selection of an Optimal Portfolio is the main problem of financial investment decisions to both the investor and the Portfolio Manager. Different portfolio may show different behaviors over time, even when their estimated risk and returns are the same. It therefore becomes difficult when selecting the Optimal Portfolio.

Given a set of N selected securities $S_1, S_2, S_3, ..., S_N$, we want to search for the optimal set of securities an investor can invest into so as to maximize return and minimize risk.

Portfolio selection problems containing cardinality and boundary constraints are computationaly intensive when using the traditional methods. Nevertheless, genetic algorithm as a search heuristic gives a fast and robust way to solve the problem. However, in applying genetic algorithm to portfolio selection, different crossover types have distinct effect on the portfolio. The appropriate choice of a type of crossover will help an investor to select an optimal portfolio. This will greatly depend on the fitness function, $f(\mu, \theta)$ which will be defined.

1.3 Objectives Of The Research

The objective of this work is:

- To find a crossover type in Genetic Algorithm, that gives the optimal portfolio.
- To maximize the return and minimize the risk associated to a Portfolio.
- To find the appropriate allocation of weights (investment) to each portfolio:

1.4 Significance of The Study

The study will be significant in the following ways:

1. It will help the Portfolio Manager to know the crossover type which gives the Optimal Portfolio with the highest return and lesser risk.
2. It will assist the investor to make a decision on the appropriate allocation of weight to each stock to get the optimal portfolio.
3. It will confirm the efficiency and the robustness of Genetic Algorithm.

1.5 Methodology

The prices of equity stock time series data for five (5) randomly selected companies were taken from Ghana Stock Exchange for a period of six (6) years. The expected returns and covariance are calculated for each year over the selected period of time.

A population is randomly generated to search for the solution space of the problem. The population so formed is called the Candidate points of the chromosome. Genetic algorithm leads to a competition phenomenon between the chromosomes. A suitable encoding of chromosome, vector of real values is generated which can store the problem specific information to be solved. In binary encodings, the alphabet is {0,1}. Chromosomes are selcted for crossover and mutation based on their fitness values for the creation of new individuals. This new individuals or generations consist of better chromosomes that are more adapted to their environment. The less performed solution is removed from the solution. This procedure is iterated several times until the stopping criteria is met.

There are four stages in the Genetic Algorithm process namely:

1. **Evaluation**, which measures the fitness of each individual solution in the set of candidates.

2. **Selection**, which randomly selects fit individuals of the present set of generation for the next iteration.

3. **Crossover**, which randomly takes two individuals in the fit individuals at a certain Probability, P_c and combines them at a crossover point, thereby creating new individuals.

4. **Mutation**, which also randomly change the genes or bits of an individual with a certain probability, P_m which is usually lower than the Crossover probability, P_c introducing additional changes into the set of candidates.

Crossover, Mutation and Selection are called *the Operators* of GA.

1.6 Scope And Limitation Of The Study

This thesis is restricted to the objective of the research. There are limitations associated with Genetic Algorithms as a mathematical optimization technique, which include: the failure to converge, premature convergence, the choice of an appropriate representation technique, specification of the termination criterion. If convergence is not assured, then this drawback can eliminate the most significant advantage of GA, which is its robustness.

1.7 Thesis Outline

The thesis basically contains five(5) main chapters. Chapter one(1) discusses the background of the study, the statement of the problem, Objectives as well as Significance of the research, the methodology, the scope and limitations of the study. Chapter two(2) explores the reasons for choosing Genetic Algorithm as our search heuristic, limitations of Genetic Algorithm, and some previous research relevant to the study. Chapter three(3) introduces some basic definitions relevant to our study, the methodology used in carrying out this study, and the solving steps. Analysis and results are explored in chapter four(4). Chapter five(5) deals with the conclusion and recommendation. It also gives recommendation to areas that can be researched in the future by other researches.

Chapter 2

Literature Review

2.1 Introduction

The idea of the review of related literature is to discover facts, findings, concerning the area of study and how they can motivate the researcher to explore the unknown.

2.2 Origin Of Genetic Algorithm

GA's are inspired by genetic inheritance and Darwin's theory of natural evolution and survival of the fittest. Darwin's theory states that individuals with certain favourable traits are more likely to survive and reproduce thereafter passing on their traits to their offspring's whilst individuals with less favourable traits will slowly disappear from the population. In the natural world, organisms that are well adapted to their environment continue succession whilst the poorly adapted ones die off. GA is direct, parallel, stochastic method for global search and optimization, which imitates the evolution of the living beings. GA's are a subclass of evolutionary algorithm where the elements of the search space are binary strings or arrays of other elementary types. Evolutionary Algorithm (EA's) are also population-based metaheuristic optimization algorithms that use biology-inspired mechanisms like mutation, crossover, natural selection, and survival of the fittest in order to refine a set of solution candidates iteratively.

The term *Genetic Algorithm* was first used by John Holland in his book *Adaptation in Natural and Artificial Systems* in 1975 (Holland, 1975). The concept of Genetic Algorithms go back to the mid 1950's, where famous biologist like

Barricelli and Fraser, the computer scientist began to apply computer-aided simulations in order to gain more knowledge into genetic processes and the natural evolution and selection. Bremermann and Bledsoe also used evolutionary approaches based on binary strings genomes for solving inequalities, for function optimization and for determining the weights in neural networks in the early 1960's. At the end of that decade, important research of such search spaces was contributed by Bagley, 1967 (who introduced the term genetic algorithm), Rosenberg, Cavichchio Jr., and Frantz—all based on the idea of Holland at the University of Michigan. As a result of Holland's work, GA as a new appoach for problem solving could be formalized finally and became widely recognized and popular.

2.3 Why Genetic Algorithm

Genetic Algorithms (GA) has been used in many applications due to certain underlying importance it has over the traditional method of optimization.

GAs work with a string-coding of variables instead of the variables. The advantage of working with a coding of variables is that the coding discretizes the search space, even though the function may be continuous. On the contrary, since GAs require only function values at each discrete points, a discrete or discontinuous function can be handled with no additional cost.

GA is particularly suitable for complicated and non-linear optimization problems, which are difficult to solve by traditional method

GA's need only information relating to the quality of the solution produced by each parameter set (objective function values) unlike many optimization techniques which needs the derivative information or even the complete knowledge of the parameters and the problem structure.

GA's can be characterized by its efficiency and robustness. GA uses a direct method to search for the optimal or near-optimal solution to complex problems, which include some of the following features:

- Large amount of noise in the data
- Very large search space
- Presence of multiple optima
- Non-differentiability and Discontinuity of the objective function

Due to their parallel nature, GAs are also much more efficient in going through vast search spaces than traditional algorithms. This however extends the search in multiple directions of the solution space in a highly efficient manner. As a result, better performing candidates pass on their binary structure or better performing schemata (certain pattern or sequence of binary digits in a vector representation) to successive generations. This property of efficiency ensures a faster convergence to the optimum compare to other methods.

2.4 Limitations of Genetic Algorithm

The importance of genetic algorithms does not come without certain immanent risks and limitations.

The most apparent limitations of a Genetic Algorithm is its premature convergence usually to local optima and the failure to converge in heavily constrained or highly nonlinear problems. These limitations can eliminate the most useful part of GA, thus its robustness. Premature convergence emerge when an individual with a notably higher fitness score than all other individuals appear in early generations and reproduce far too quickly. However, premature convergence can be some what avoided by slight improvement to the standard genetic algorithm.

Genetic Algorithms are typically only able to find estimates of optimal solutions or they provide only optimal solutions not the exact or precise solutions. Hence if an exact optimum is desired, the solution provided by the GA can be used as the initial conditions to a more traditional, gradient-based optimization algorithm.

Small populations normally result in good solutions but periodically get stuck on local optima. Larger population size is less likely to be stuck by local optima but they also take longer time to find good solutions.

2.5 Review Of Related Literature

Genetic Algorithms have been used in many pragmatic problems in finance and investments, which sometimes require constructive and robust optimization techniques. Forecasting returns, model calibration, portfolio optimization, trading rule discovery, option pricing, emergence of economic markets and development of bidding scheme are some of the applications of genetic algorithms to complex problems in the financial and investment market (Pereira, 2000).

Holland explained that GA's are stochastic optimization algorithms based on the mechanisms of natural selection and Genetics.

Markowitz (1952) developed a quantitative model, also called the Modern Portfolio Theory (MPT) which states that by selecting the right and relevant combination of assets to invest in, an investor could get higher returns with the same risk rate. Markowitz also introduced the Mean-Variance model that minimizes the objective function of the Portfolio Variance for a significant level of an estimated return or maximize the portfolio expected return for a given risk rate.

However, a new model for Portfolio Selection was proposed by Xia et al. (1999) in which the expected returns of securities are studied as variables rather than as the arithmetic mean of securities. A genetic algorithm was then used to solve the optimization problem, which is hard to solve with the existing traditional algorithms due to its non-concavity and special structures. They compared their results with those derived from the traditional model of Markowitz. The comparison showed that the performance of the new models are more expedient than that of the mean-variance models of Markowitz.

Nonetheless, Chang et al. (1999) considered the problem of searching the efficient frontier related with the standard mean-variance portfolio optimization model. The model was extended to include cardinality constraints. Differences that arose in the shape of the efficient frontier when such constraint was present were discussed. Three heuristic algorithms: GA, tabu search, and simulated annealing was introduced to find the cardinality constrained effficient frontier.

Lin and Wang (2002) proposed a two mean-variance models for portfolio selection with fixed transaction costs and minimum transaction lots. The portfolio selection problems were modeled as a non-smooth nonlinear integer programming problem with single objective function and multiple objective functions respectively. A new genetic algorithm was designed to solve their proposed models. The results was illustrated via a numerical example that GA can be used to solve portfolio selection problems efficiently in practice.

Samanta and Roy (2005) reviewed some general applications of portfolio selection problem. Firstly, a multi-objective portfolio selection based model was studied and then another entropy objective function, Shannon's measure of entropy was added. Following this, the entropy-based problem was constructed in a generalized form. Fuzzy non-linear programming technique was then used to solve the problems. Their suggestion was that entropy may be used in other fields of operation research and engineering sciences.

Wang et al. (2005) proposed a non-linear stochastic optimization algorithm named Stochastic Portfolio Genetic Algorithm (SPGA) to find a profitable portfolio selection planning under risk. They noticed that their algorithm improved a conventional two-stage stochastic programming by integrating a genetic algorithm into a stochastic sampling procedure to solve the large-scale portfolio selection optimization. Their results drawn from a data collected from Taiwan Stock Exchange shows that a practical problem can be efficiently solved and that the expected return of SPGA outperforms the one in the market.

Zhou et al. (2006) used GA to identify the stocks that are likely to perform exceedingly than the market by having excess returns. Their experimental results revealed that the Ga optimization approach is very important and flexible when dealing with stick selection and this will help the investor when selecting valuable stocks.

A double-stage genetic optimization algorithm for portfolio selection was proposed by Lai et al. (2006). In the first stage, GA was used to rank the stock and the quality stock for portfolio optimization. In the second stage, optimal asset allocation for portfolio was realized by GA. The Simulations done revealed that their proposed two-stage GA is an effective portfolio optimization approach, which can supply valuable portfolio for investors. Their study also revealed that the number of stocks in a portfolio does nopt satisfy "the more, te better" principle, but a sparse number of stocks can improve the performance of a portfolio.

Zhang et al. (2006) applied adaptive GA to solve the portfolio selection problem in which there exist both probability constraint on the lowest return rate of portfolio and the lower-upper bounds constraints. The stochastic model of portfolio selection and its reliability decision was proposed. The adaptive genetic algorithm was applied to obtain the reliability decision of portfolio selection. They finalized it by giving a numerical example to illustrate the proposed effective means.

Wang et al. (2006) proposed a non-linear stochastic optimization algorithm named Stochastic Portfolio Genetic Algorithm (SPGA) to find a profitable portfolio selection planning under risk. They noticed that their algorithm improved a conventional two-stage stochastic programming by integrating a genetic algorithm into a stochastic sampling procedure to solve the large-scale portfolio selection optimization. Their results drawn from a data collected from Taiwan Stock Exchange shows that a practical problem can be efficiently solved and that the expected return of SPGA outperforms the one in the market.

Lin and Gen (2007) made a seminal observations that the mean-variance is oftenly used in the finance area to manage most Portfolio selection problem. They stated that the objective of the mean-variance approach is to determine the period optimal investing rate to each security based on the sequent return rate. They designed a multistage decision-based genetic algorithm to solve the corresponding optimization problems because of the non-concave maximization problem that cannot be solved by the existing traditional optimization methods. The experiment result showed that the proposed model was valid for the portfolio optimization problem

In his thesis work, Roudier (2007) goal was to develop robust portfolio optimization methods. He designed a multi-factor objective function reflecting an investment preferences and the subsequent optimization problem was solved using a genetic algorithm.

Anagnostopoulos and Mamanis (2008) developed a computational procedure in order to find the efficient frontier for the standard Markowitz mean-variance enriched with integer constraints. These constraints limit the portfolio to both contain a pre-established number of assets and the portion of the Portfolio held in a specified asset. The problem is solvd by modifying the multiobjective algorithm NSGA (Non-dominated Sorting Genetic Algorithm) that ranks the solution of each generation in layers or the foundation of Pareto non-domination.. The algorithm was applied to 60 assets of ATHEX and the computational results indicated that the procedure is encouraging.

Chang et al. (2009) introduced different risk measures for analyzing portfolio and integrates them into a GA framework. They then compared its performance to the mean-variance model in cardinality constrained efficient frontier. To achieve this target, they collected three different risk measures based upon the classical mean-variance of Markowitz; semi-variance , Mean absolute deviation and variance with skewness. They made a seminal observation that these portfolio optimization problems can be solved by GA if the mean-variance, semi-variance, mean absolute deviation and variance with skewness were used as the measures of risk. They concluded that investors should include only only one-third of their total assets into the portfolio which performs better than those contained more assets.

In their paper, Hachloufi et al. (2012) presented an approach based on the classifications of genetic algorithms for an optimal choice of a reduced size portfolio. This led to a financial gain surplus in terms of cost and taxes reduction, and performance at reduced design loads. Firstly, they classified the actions if classes known as under portfolio using the algorithm k-means. Following this, a dynamic optimization algorithm MinVArMAxVaL was applied on the class (under portfolio) that has the highest expected return and the lowest average Var. The objective of their algorithm was to minimize risk and maximize portfolio value at the ame time through some stages.

In a study than on 50 Supreme Tehran Stock Exchange Companies, Pandari et al. (2012) applied GA to select the best portfolio in order to optimize their objectives of the rate of return, return skewness, liquidity and Sharpe ratio. The obtained results were compared with the results of Markowitz classic model. The comparison showed that, the rate of return of the portfolio of GA model was less than that of the Markowitz classic model. They concluded that GA can lead to better results and could help investors and Portolio Managers to make the optimal portfolio selction as far as the selection of the best portfolio is concerned.

The selection of a portfolio encounters several extremely complex situations. Amongst them is the financial assets selection when interrelations (positive and/or negative) occur among the expected profitabilities of each of them. Genetic Algorithms are used to solve this situations due to its utility when offering solutions to sophisticated optimization problems. By using the Fuzzy Logic, Divya and Kumar (2012) planned to obtain a closer representation for the uncertainty that signalize Financial Market, thereby outlining an approach to solve Financial Assets selection problems for a portfolio in a non-linear and uncertainty environment, by the application of a Fuzzy logic and Genetic Algorithm to optimize the investment profitability.

Aftalion (2012) explores the use of Genetic Algorithm in optimizing resource allocation in a given Portfolio and he further discussed its relation to Modern Portfolio Theory. His solution presented the Efficient Frontier in a graphical representation by utilizing a Simple Genetic algorithm. However, he recommended that his studies needs to be broadened so as to obtain an optimal parameters for the genetic algorithm itself. He also clarify that his studies is meant for "educational purpose" rather than an applications to be used by professionals.

Sefiane and Benbouziane (2012) applied GA on a five(5) stock asset portfolio. Their result obtained confirmed the efficacy of GA for its rapid convergence towards the optimal solution and its fast algorithm time.

In their recent work, Sinha et al. (2013) generated an algorithm to construct an optimum portfolio from a vast pool of stocks listed in a single market index SPX 500 index. Their algorithm however selects stocks on the grounds of a priority index function created on company fundamentals and genetically give optimum weights to the stocks selected by searching for a genetically appropriate combination of return and risk on the grounds of historical data. Nonethelesss, it was clear that genetic algorithm was successfull in giving the optimum weights to stocks that were primarily screened through a predetermined priority index function. The portfolio constructed outperformed the market for the considered holding period by an appreciable margin.

Aliev et al. (2013) proposed a fuzzy portfolio selection model based on fuzzy linear programming solved by GA. GA provides for finding a global near-optimal solution with a reduction in computational complexity compared to the existing methods. Their model takes into consideration fuzzy expected return, investor's fuzzy risk preference and provides a probable trade-off between risk and return. In order to achieve this, they assigned degree of satisfaction between constraints, criteria and defining tolerance for the constraints in order to achieve the goal value in the objective risk function. Their experimental results showed high efficiency of their proposed method. They indicated that using deterministic and stochastic portfolio models to solve a portfolio construction problem leads to unrealistic results as both the expected return rate and the risk are vague.

In their paper, Misra and Sebastian (2013) applied GA to portfolio optimization of commercial bank (Bank of India). They indicated that portfolio of a commercial bank can be constrained by regulating prescription of exposure limits, risk weights and returns from each class of assets. Therefore, optimization of return, in case of the loan portfolio, presents a challenging problem due to its large set of local extremes. The application of genetic algorithm in their work was used as a possible solution to optimize the risk-return trade-off and an ideal solution for optimization of portfolio was achieved.

Chapter 3

Methodology

3.1 Introduction

This chapter describes the theory of the concept to be used, derivation and methods of analyzing the available data to satisfy the objectives of the study. It focuses on the detail and comprehensive understanding of a comparative study of Portfolio Optimization using Genetic Algorithm and the Markowitz Mean-Variance approach visa vie its application to Portfolio Selection.

3.2 Financial Concepts

Portfolio refers to any collection of financial assets such as bonds, stocks, and cash. Portfolios may be held by individual investors and/or managed by financial professionals, hedge funds, banks and other financial institutions. Nevertheless, it is universally accepted that the design of the portfolio is made according to the investors risk tolerance and the investment objectives, (Wikipedia, Portfolio definition, retrieved august 2013).

Security

A security or financial instrument is a tradable asset of any kind. It represents an ownership position in a publicly-traded corporation (stocks), a creditor relationship with governmental body or a corporation (bond), or rights to ownership as represented by an option. The company or entity that issues the security is known as the issuer.

Shares

The stock of a corporation is divided into shares, the total of which are clearly identified at the time of business formation. It can also be explained as a unit of ownership interest in a corporation or financial asset. Shares designate a fraction of ownership in a business.

Efficient Frontier

Efficient portfolio is the portfolio that provides the greatest expected return for a given level of risk, or equivalently, the lowest risk for a given expected return. Portfolios that lie below the efficient frontier are sub-optimal, because they do not provide enough return for the level of risk. Optimal Portfolios that include the efficient frontier tend to have a higher degree of diversification than the sub-optimal ones, which are typically less diversified. The efficient frontier concept was introduced by Harry Markowitz in 1952 and is the cornerstone of Modern Portfolio Theory(MPT).

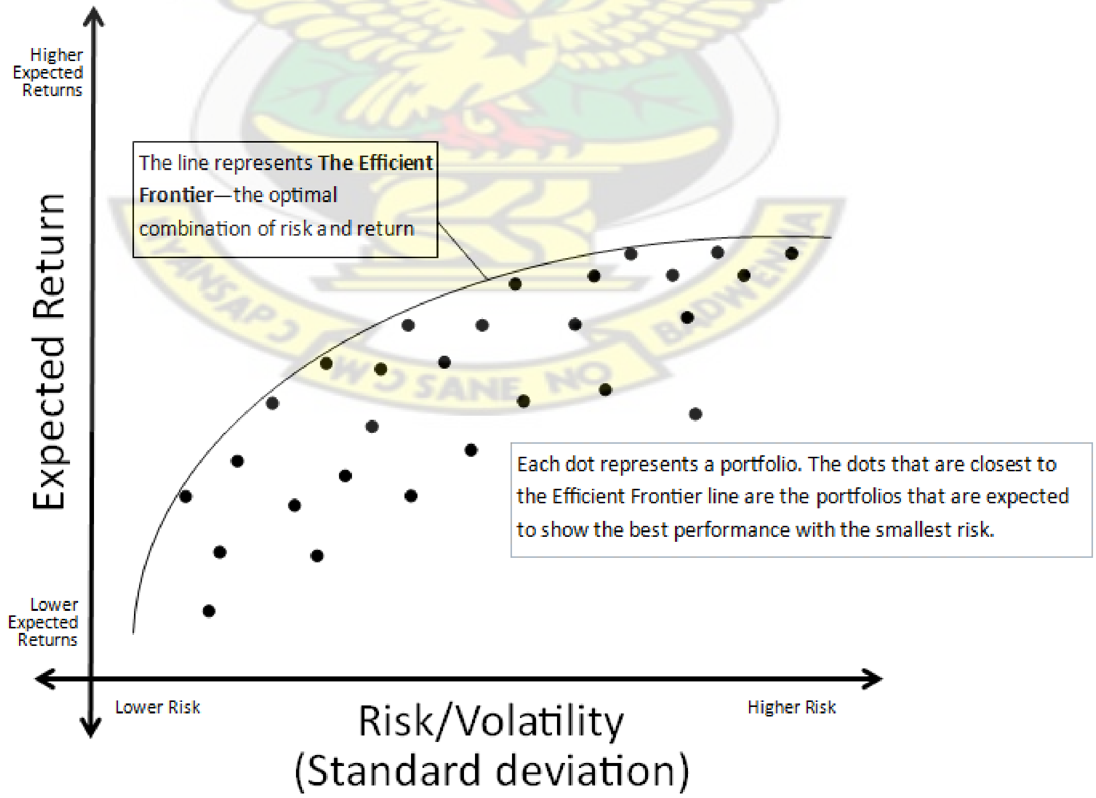

Figure 3.1: A diagram of Efficient Frontier

Stocks

A stock is a type of security that signifies ownership in a corporation and represents a claim on part of the corporations' assets and earnings.The price of a stock reflects the value of the company as estimated by the market. Usually, the larger the expected return in the future, the larger the risk because of the risk-averse nature of most investors; thus, there exist a positive relationship between the risk of a stock and the expected return of the same stock. Stocks are the foundation of almost every portfolio.

Portfolio of Stocks

The portfolio of stocks refers to the collection of individual stocks owned by an investor. The portfolio risk does not only take into account the weighted sum of the underlying risks, but also considers the correlation between all the stocks and is obtained by weighing the covariance matrix of the stock returns.

Prices

The Price of an asset at time,t is P_t. The price of a stock is taken as the closing price and is always taken either at the end of the day, the end of the week or the end of the month. Prices of a stock may often be described as logprices, which is defined as:

$$\log P_t = \log(P_t)$$

Return

The return of an asset is the relative change in its value over time, whether the price of an asset has increased (positive return) or decreased (negative return). The return of an asset in applications of finance is usually controlled on the idea of the Estimated Return.

The return on a stock at time, t is defined mathematically from time $t-1$ to time t as:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$$

Where P_t is the price at time,t and P_{t-1} is the price at time $t-1$.

Suppose Div_t is the dividend on the stock portfolio, then

$$R_t = \frac{P_t - P_{t-1} + Div_t}{P_{t-1}}$$

For small variation of P_t, we have $P_t \simeq P_{t-1}$ and the return at time,t is very small:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} = \frac{P_t}{P_{t-1}} - \frac{P_{t-1}}{P_{t-1}}$$

$$R_t = \frac{P_t}{P_{t-1}} - 1 \simeq 0$$

For a small x, the following first order approximation:

$$\log(1+x) \simeq x$$

In the same approach, for small price variations, it can also be written:

$$\log \frac{P_t}{P_{t-1}} = \log(1 + \frac{P_t}{P_{t-1}} - 1)$$

$$\log \frac{P_t}{P_{t-1}} = \log(1 + R_t) \simeq R_t$$

Risk

The risk of an asset is the probability that the real value of the return in the future will be different than the expected return. In Modern Portfolio Theory, the measure of risk is determined as the Variance or the Standard deviation of the rate of return of a particular asset.

3.3 Mean-Variance Optimization

Mean variance optimization (MVO) is a quantitative tool that permit an investor to allocate his/her wealth between different assets by considering the trade-off between risk and return. In mean-variance analysis, we consider two moments (mean and variance) in the formulation of our portfolio model. The mean, variance and covariance are all unknown.

3.3.1 Measures Of Mean

We define a method to calculate the first parameter in the MPT, eventhough the underlying problem is the risk. It is practicable to calculate the mean return of an investment with various methods, but mainly geometric and arithmetic mean return is used.

Arithmetic Mean

The arithmetic mean return is the simple average of a series of periodic returns. It pose the statistical property of been an unbiased estimator of the true mean of the specific distribution of returns. It is defined mathematically as;

$$\bar{r} = \frac{1}{n}\sum_{j=1}^{n} r_j = \frac{1}{n}\left(r_1 + r_2 + ... + r_n\right) \tag{3.1}$$

where,

$\bar{r}$ = arithmetic mean return, r_j = sample data, and n = Number of assets.

Geometric Mean

The geometric mean return is a compound annual rate. When periodic rate of return change from one period to another, the geometric mean return will have a value less than the arithmetic mean return. The geometric mean return of the data set $[r_1,\, r_2,\, r_3, ...,\, r_n]$ is;

$$\left(\prod_{j=1}^{n} r_j\right)^{\dfrac{1}{n}} = \sqrt[n]{r_1 \cdot r_2 \cdots r_n} \tag{3.2}$$

where,

$\bar{r}$ = arithmetic mean return, r_j = sample data, and n = Number of assets.

Geometric Mean versus Arithmetic Mean

The comparison between these two average methods is possible using Jensen's inequality.

Theorem 3.3.1 *For any random variable X, if $f(x)$ is a convex function, then $Ef(X) \geq f(EX)$. Equality holds iff, for every line $a + bX$ that is tangent to $g(x)$ at $x = EX$, $P(g(X) = a + bX) = 1$.*

The above theorem can be used to prove the variation between the two methods of averaging. If $b_1,\, b_2,\, b_3,\, ...,\, b_n$ are non-negative numbers, defined as;

$$b_A = \frac{1}{n}\,(b_1 + b_2 + ... + b_n)$$

$$b_G = [b_1 b_2 b_3 \cdots b_n]^{\frac{1}{n}}$$

An inequality relating these mean's is;

$$b_A \geq b_G$$

To apply the Jensen's inequality, we let X be a random variable with range $b_1,\, b_2,\, b_3,\, ...,\, b_n$ and $P = (X = a_j) = \dfrac{1}{n}, j = 1,\, 2, ..., n$. Jensen's Inequality prove that $E(\log X) \leq \log(EX)$, since $\log x$ is a concave function; hence,

$$\log b_G = \frac{1}{n}\sum_{j=1}^{n} \log a_i = E(\log X) \leq \log(EX) = \log\left(\frac{1}{n}\sum_{j=1}^{n} a_j\right) = \log a_A$$

hence $b_A \geq b_G$.

Markowitz suggested that Geometric mean method is better than the Arithmetic mean method when calculating the growth rate of a portfolio. He argues that Geometric mean method generates a more realistic result as compared to Arithmetic and Compounding average method.

Mean of An Asset

The mean (expected return) of a particular asset with prices P_1, P_2, $P_3, ...P_T$ is defined as:

$$\mu_i = \frac{\sum_{t=1}^{T-1} \ln\left(\frac{P_t+1}{P_t}\right)}{T-1} = \frac{\sum_{t=1}^{T-1} \ln\left(\frac{P_t}{P_t-1}\right)}{T-1}$$

$$\mu_i = \frac{\sum_{t=2}^{T} \ln(r_t)}{T-1}$$

This is equivalent to calculating the average return or mean of the asset. For example, if an asset increases 5% for four consecutive months, its total return obviously is not 5% but $\sqrt[4]{0.05}$.

3.3.2 Variance and Standard Deviation

Variance and Standard deviation are common measures of investment risk. They measure the variability of distribution of returns about their mean or expected value. The population variance, σ^2 can be calculated when the return, R_t for each period, the total number of periods, T, and the mean or expected value of the population's distribution, μ is known. However, in the field of finance, we use the sample variance because only a sample of returns data are always used.

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(r_i - \bar{r})^2$$

where,

s=The sample variance, r_i=The return of the data set, $i = 1, 2, ..., n$ and $\bar{r}$=The mean return

The standard deviation of a sample of measurement is the positive square root of the variance. The standard deviation grows as returns move further away the average. The standard deviation is a well founded measure of the returns that are below the mean Markowitz if the returns on a stock or portfolio are normally distributed. However, if the standard deviation is skewed, then the standard deviation fails to be meaningful.

Variance and Standard Deviation of An Asset

The risk of a particular asset with prices P_1, P_2, $P_3, ...P_T$ is defined as:

$$\sigma_i^2 = \frac{\sum_{t=1}^{T-1}\left(\ln\frac{P_t+1}{P_t} - \mu_i\right)^2}{T-1} = \frac{\sum_{t=1}^{T-1}(\ln(r_i) - \mu_i)^2}{T-1}$$

The expression above is equivalent to the assets return variance.

Covariance

Covariance is the degree to which returns on two risky assets move together. Assets which have a positive covariance of returns implies that the return on the assets move in tandem whilst a negative covariance means the returns of the asset move inversely. The covariance of two assets, i and j can be expressed as:

$$\sigma_{ij} = \rho_{ij}\sigma_i\sigma_j \tag{3.3}$$

ρ_{ij} is the correlation between assets i and j

3.3.3 Portfolio Expected Return and Risk

Return

The estimation of the expected returns of each asset is needed in the selection of a portfolio. The historical mean of the stock is taken to be the return.

A portfolio, P is a weighted composition with n weights $(W_1, W_2, ..., W_n)$. We let W_i be the proportion of the investors' total invested wealth on asset i. If r_i is a random variable which denotes the expected return of asset i and n is the total number of asset, then the return of a portfolio denoted by r_p, which consist of all these assets is given by:

$$r_p = W_1 r_1 + W_2 r_2 + W_3 r_3 + ... + W_n r_n = \sum_{i}^{n} W_i r_i \tag{3.4}$$

We take the expectation of both sides of the above equation,

$$E(r_p) = E\left[\sum_{i=1}^{n} W_i r_i\right] = \sum_{i=1}^{n} W_i E(r_i) \tag{3.5}$$

Let $E(r_p) = R$ and $E(r_i) = \mu_i$

$$\Rightarrow R = \sum_{i=1}^{n} W_i \mu_i \tag{3.6}$$

Risk

In Modern Portfolio Theory, Markowitz uses the variance of the return of an asset as a measure of the risk. The variance of a random variable,thus the return over time of an asset, is the measure of the variability of distribution of returns over its mean or expected value. In practice, the risk is defined for a sample of data as:

$$\sigma_i^2 = \frac{1}{n-1} \sum_{i=t-n}^{t-1} r_i - \bar{r}$$

For a given asset i, the risk can be defines as:

$$\sigma_i^2 \ = \ E[(r_i - E[r_i])^2]$$

where $\bar{r}$ is the average of the period (moving average) and r_i is the value of the return at time i.

The variance of the portfolio, P can also be written as:

$$\sigma_p^2 \ = \ E\left[(r_p - \mu_p)^2\right] \tag{3.7}$$

from equations 3.4 and 3.6,

$$\sigma_p^2 \ = \ E\left[\left(\sum_{i=1}^{n} W_i r_i - \sum_{i=1}^{n} W_i \mu_i\right)^2\right] \ = \ E\left[\left(\sum_{i=1}^{n} W_i (r_i - \mu_i)\right)^2\right] \tag{3.8}$$

$$\sigma_p^2 \ = \ E\left[\sum_{i=1}^{n} W_i^2 (r_i - \mu_i)^2 + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} W_i W_j (r_i - \mu_i)(r_j - \mu_j)\right] \tag{3.9}$$

The covariance of assets i and j is defined as:

$$\sigma_{ij} = cov[r_i, r_j] = E[(r_i - E[ri])](r_j - E[r_j])$$

$$\sigma_{ij} = cov[r_i, r_j] = E[(r_i - \mu_i)(r_j - \mu_j)] = \sigma_{ji}$$

$$\Rightarrow \ \sigma_{ij} = \sigma_{ji}$$

$$cov[r_i, r_i] = var[r_i]$$

But $\sigma_i = r_i - \mu_i$, $\sigma_{ij} = (r_i - \mu_i)(r_j - \mu_j)$,
where σ_i and σ_{ij} is the standard deviation of asset i and σ_{ij} is the covariance between asset i and j.

$$\sigma_p^2 \ = \ \sum_{i=1}^{n} W_i^2 \sigma_i^2 \ + \ 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} W_i W_j \sigma_{ij} \tag{3.10}$$

From equation 3.3, we can rewrite equation 3.10

$$\sigma_p^2 = \sum_{i=1}^{n} W_i^2 var(r_i) + \sum_{i=1}^{n} \sum_{i=1, j\neq 1}^{n} W_i W_j \rho i j \sigma_i \sigma_j \tag{3.11}$$

Obviously $var(r_i) = \sigma_{ii}$, hence the above equation can be written as:

$$\sigma_p^2 = \sum_{j=1}^{n} W_i W_j \rho_{ij} \sigma_i \sigma_j = \sum_{i=1}^{n} \sum_{j=1}^{n} W_i W_j \sigma_{ij} \tag{3.12}$$

We let ϑ denote the covariance matrix so that

$$\sigma_p^2 == W^T \vartheta W$$

Where:

$$\sigma_{ij} = \frac{\sum_{t=1}^{T-1} [(r_t^i - \mu_i)(r_t^i - \mu_i)]}{T-1}$$

Introducing the natural logarithm to compensate for absolute price changes:

$$\sigma_{ij} = \frac{\sum_{t=1}^{T-1} \left[\left(\ln\left(\frac{P_{t+1}^i}{P_t^i} \right) - \mu_i \right) \left(\ln\left(\frac{P_{t+1}^j}{P_t^j} \right) - \mu_j \right) \right]}{T-1}$$

$$\sigma_{ij} = \frac{\sum_{t=1}^{T-1} \left[\left(\ln\left(r_t^i \right) - \mu_i \right) \left(\ln\left(r_t^j \right) - \mu_j \right) \right]}{T-1}$$

P_t^i is the price of asset i at time and P_t^j means the price of asset j at time t.
r_t^i is the price of asset i at time r_t^j means the price of asset j at time t.
The natural logarithm is used in the above equation in order to use relative rather than absolute price changes.

3.3.4 Correlation

It is easier to describe the co-movement of two assets as a correlation rather than covariance. Correlation summarizes the relationship of one asset to other assets.

A correlation between assets can be positively correlated, negatively correlated or may have no correlation between them. The risk is greatest when the correlation coefficient is 1. From equation 3.11, we consider a portfolio of 2 assets and write the variance σ_p^2 of our portfolio as a function of weights (w_1, w_2) .

$$\sigma_p^2 = (\sigma_1 w_1)^2 + (\sigma_2 w_2)^2 + 2w_1 w_2 \rho_{12}\sigma_1\sigma_2 \tag{3.13}$$

(w_1, w_2) is subject to constraints:

- $w_1 + w_2 = 1$
- $\forall i,\ w_i \geq 0$

We replace w_2 by $(1 - w_1)$, giving us the following expression of σ_p:

$$\sigma_p^2 = (\sigma_1 w_1)^2 + (\sigma_2(1 - w_1))^2 + 2w_1(1 - w_1)\rho_{12}\sigma_1\sigma_2 \tag{3.14}$$

Perfect Correlation between assets,$(\rho_{ij} = +1)$

$$\sigma_p^2 = (\sigma_1 w_1)^2 + (\sigma_2(1 - w_1))^2 + 2w_1(1 - w_1)\sigma_1\sigma_2$$

which factorizes into:

$$\sigma_p^2 = (\sigma_1 w_1 + \sigma_2(1 - w_1))^2 = \sum_{i=1}^{n} w_i\sigma_i$$

In this special case, the Portfolio standard deviation is the weighted average of asset standard deviations. Hence the maximal portfolio standard deviation is the simple weighted average of component standard deviations when $\rho = 1$. In simple terms, the variance of a portfolio is greatest when $\rho_{ij} = +1$. So for any other ρ there will be a lower σ_p^2.

No Correlation between assets, $(\rho_{ij} = 0)$

$$\sigma_p^2 = (\sigma_1 w_1)^2 + (\sigma_2(1 - w_1))^2 = \sum_{i=1}^{n} \sigma_i^2 w_i^2$$

A correlation coefficient of zero means that there is no linear relationship between the two stocks' returns.

Perfect anti-Correlation between assets,($\rho_{ij} = -1$)

$$\sigma_p^2 = (\sigma_1 w_1)^2 + (\sigma_2(1-w_1))^2 - 2w_1(1-w_1)\sigma_1\sigma_2$$

which can factorize into:

$$\sigma_p^2 = (\sigma_1 w_1 - (\sigma_2(1-w_1))^2$$

$$\sigma_p = |(\sigma_1 w_1 - (\sigma_2(1-w_1))|$$

If we choose $w_1 = \dfrac{\sigma_2}{\sigma_1\sigma_2}$ and $w_2 = 1 - w_1$,

$$\sigma_p = \frac{\sigma_2}{\sigma_1+\sigma_2}\sigma_1 - \left(1 - \frac{\sigma_2}{\sigma_1+\sigma_2}\right)\sigma_2$$

$$\sigma_p = \frac{\sigma_1\sigma_2 - \sigma_2^2 - \sigma_1\sigma_2 + \sigma_2^2}{\sigma_1+\sigma_2} = 0$$

This implies that if assets are perfectly negatively correlated, the portfolio has zero variance.

3.3.5 Diversification

Diversification refers to a risk management technique that mixes a wide range of variety of investments within a portfolio. The principal objective behind this technique contends that a portfolio of diverse kinds of investments will, on average, yield higher returns and pose a lower risk than any individual investment found within the portfolio. Diversification strives to smoothen out unsystematic risk (risk which is unique to a particular asset, also called idiosyncratic/diversifiable risk) in a portfolio so that the positive performance of some investments will neutralize the negative performance of other investments. Diversification is really actualized when the securities in the portfolio are not perfectly correlated,

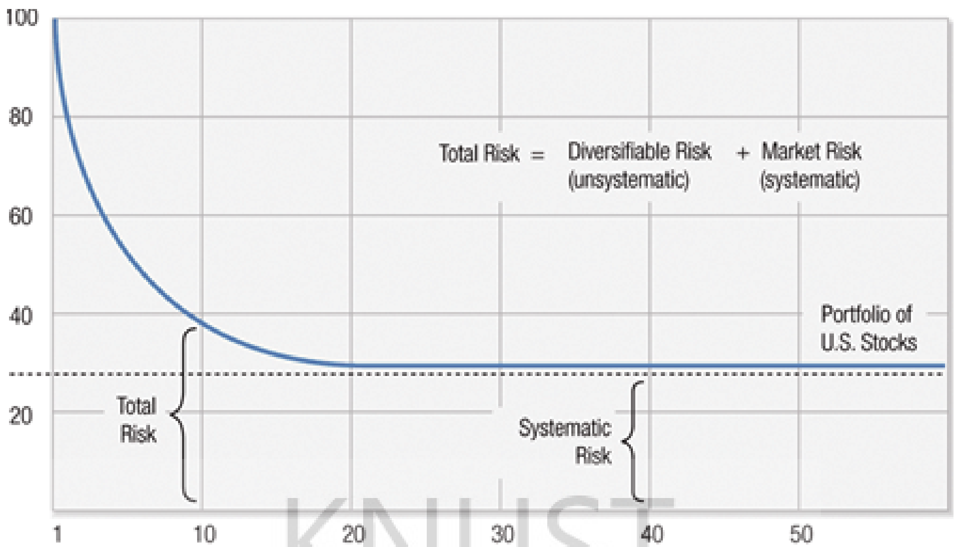

Figure 3.2: A diagram of Portfolio Diversification

(Investopedia, definition of diversification, retrieved august, 2013). But market risk (systematic risk) cannot be eliminated since it is common to all assets in the market.

Portfolio diversification works best when financial markets are operating normally; diversification provides less reduction of risk during market agitation, such as the credit contagion of 2008.

The first part of equation 3.11 describes the portfolio risk portion which constitute the risk of individual assets. It is called the *Specific risk*.

Theorem 3.3.2 *If the returns on assets in a portfolio are uncorrelated, specific risk can be reduced by diversification.*

Proof

Consider a portfolio wint$n$ assets. Assume that $W_i = \frac{1}{n}$ and $\rho_{ij} = 0$, $\forall i, j$, then the risk of the portfolio is:

$$\sigma_p^2 = \sum_{i=1}^{n} \sigma_i^2 \left(\frac{1}{n}\right)^2 = \frac{\sum_i^n \sigma_i^2}{n^2}$$

The mean of the risks is:

$$\bar{\sigma} = \frac{\sigma_i}{n}$$

Assume we let the risk of all assets equal to the highest risk, σ_{max}. As $n \to \infty$, $\sigma_p^2 \to 0$ and $\bar{\sigma_i^2} = constant$, then

$$\frac{\sum \sigma_{max}^2}{n^2} \leq \frac{\sum \sigma_{max}}{n}$$

$$\frac{n\sigma_{max}^2}{n^2} \leq \frac{n\sigma_{max}}{n}$$

$$\frac{\sigma_{max}^2}{n} \leq \sigma_{max}$$

The second part of the equation 3.11, is called the *Market/non-diversifiable risk* describes the portfolio risk shared by the assets. Suppose the variance is other than zero, then the systematic risk cannot be reduced below the average by adding more assets.

Theorem 3.3.3 *Systematic risk cannot be reduced by Diversification.*

Proof

Consider a portfolio with n assets, assume that $W_i = \frac{1}{n}, \sigma_{ii} = s$ and $\sigma_{ij} = a$ for every i and j. Then the risk of the portfolio becomes:

$$\sigma_p = \frac{\sum_i^n s^2}{n^2} + \frac{\sum_{i \neq J} a}{n^2}$$

$$\sigma_p = \frac{ns^2}{n^2} + n(n-1)\frac{a}{n^2}$$

$$\sigma_p = \frac{s^2}{n} + a + \frac{a}{n}$$

$$\sigma_p = a + \frac{s^2 - a}{n}$$

No matter the increament of n (the number of assets), the risk will never be less than the average covariance,a.

Equally Weighted Portfolio

This is a weighting method that gives the same weight or advantages to each stock in a portfolio. Each asset in a portfolio is given an equal weight irrespective of the risk or return associated to that asset. The expected return of an equally-weighted portfolio is also the expected return from selecting one of the n portfolio securities at random. The portfolio with the greatest "likely return" is not actually the portfolio with the least risk.

$$W_i = W_j = \frac{1}{n}$$

that is: $\sum_{i=1}^{n} \frac{1}{n} = 1$

The Portfolio variance then becomes

$$\sigma_p^2 = n\frac{1}{n^2}\sigma_i^2 + n(n-1)\frac{1}{n}\frac{1}{n}\bar{\sigma_{ij}} = \frac{1}{n}\sigma_i^2 + \frac{n-1}{n}\bar{\sigma_{ij}}$$

As $n \to \infty$, the formula grows asymptotically:

$$\lim_{n\to\infty}(\sigma_p^2) = \lim_{n\to\infty}\left(\frac{1}{n}\sigma_i^2\right) + \lim_{n\to\infty}\left(\frac{n-1}{n}\bar{\sigma_{ij}}\right) = \bar{\sigma_{ij}}$$

3.4 Portfolio Optimization-A Theoretical Perspective

3.4.1 Capital Asset Pricing Model (CAPM)

The CAPM was formulated by Nobel Laureate, William Sharpe in his book "Portfolio Theory and Capital Markets" in the 1970s. The model begins with the concept that an individual investment consist of two types of risk; Systematic and Unsystematic risk. Whilst the later can be curtailed for with diversification the former cannot be eliminated by diversification. CAPM, however, pave a way to measure this systematic risk, it finds out a theoretically suitable rate of return of an asset, if that asset is to be addded to an already well-diversifiable risk of the asset. The model takes into account the asset sensitivity to the market risk which is normally represented as the Market beta(β) vis-á-vie the expected return of the market and the theoretical risk free asset. Mathematically, it is defined as;

$$\bar{R}_i = r_f + \beta_i(\bar{r}_m - r_f)$$

Where r_f is the risk free rate of return, $\bar{r}_m$ is the expected market return, β_i is the beta of the asset, $(\bar{r}_m - r_f)$ is the equity market premium.

Beta is the only pertinent measure of a stock's relative volatility. A portfolio with high beta stocks will move higher than the market in either direction and vice versa. When $\beta = 1$, then the share price moves exactly in line with the stock market.

$$\beta_i = \frac{Cov(R_i, R_m)}{Var(R_m)}$$

This beta can be extended to represent the risk relationship between the market porttfolio and any portfolio:

$$\beta_p = \sum_i w_i \beta_i$$

Beta is a CAPM index that is used to evaluate the risk-return trade-off of a portfolio.

3.4.2 Sharpe Ratio (SR)

Nobel Laureate William Sharpe developed Sharpe ratio, originally called reward-to-variability ratio in 1966 and later revised in 1994. Given any two investments, the Sharpe ratio can help choose the investment type that delivers a higher return while considering its risk. The higher the Sharpe ratio, the better the return per unit of risk taken. A negative Sharpe ratio is obtained when a misappropriate level of risk is taken to generate a positive return. It is defined as:

$$S_P = \frac{E(R_P) - R_F}{\sigma(R_P)}$$

where,

$E(R_P)$=the expected return of the portfolio

R_F=the return on the risk-free asset; and

$\sigma(R_P)$=the standard deviation of the portfolio return

The SR is a raw number and it is of relevance only when it is compared with other SR for other stocks over the same time range and more importantly the same objectives. If the portfolio is well diversified, then its SR is close to that of the market.

The Sharpe Ratio can be used as an analysis tool to determine the performance of a portfolio, since it measures the amount of risk and return that can be traded by balancing the portfolio with the risk-free asset (Lin and Gen, 2007).

3.4.3 Risk Free Asset

It is also called riskless asset. This asset is described as a hypothetical asset, which pays a risk-free return to the investor, with a variance and standard deviation equal to zero. It is a security which's deviation of the mean return is 0. This type of asset is normally issued by the government and can be referred to as Government Bond or Treasury Bill (T-Bill). That is, assumption is made that government does not go bankrupt.

The presence of a riskfree asset allows for the portfolio manager to combine it with a portfolio at the efficient frontier.

3.5 Markowitz Mean-Variance Model

The difficulty in optimally selecting a portfolio among n assets was formulated by Markowitz in 1952 as a constrained quadratic minimization problem.

Markowitz suggested that, for a fixed set of expected returns (μ_i) and covariance (σ_{ij}) of the returns of all assets i and j, each and every investor can find a return-variance combination that better fits his/her expectations, uniquely limited by the constraints of the specific problem.

In the original model of Markowitz, short sales were excluded and thus,

$$W_i \geq 0,\ i = 1, 2, 3, \cdots, n$$

and the invested amount of capital was limited to a budget. Let W_i be the constant fraction of the Portfolio value invested in asset i so $\sum_{i=1}^{n} W_i = 1$. Markowitz unconstrained mean-variance model can be formulated as:

$$\text{Minimize} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} W_i W_j \sigma ij \tag{3.15}$$

Subject to

$$R = \sum_{i=1}^{n} W_i \mu_i \tag{3.16}$$

$$\sum_{i=1}^{n} W_i = 1 \tag{3.17}$$

$$W_i \leq 1, i = 1, 2, 3, \cdots, n \tag{3.18}$$

where: n is the number of assets available,
W_i is the proportion of the portfolio held in asset i,
μ_i is the expected return of asset $i\,(i = 1, 2, \cdots, N)$,
σ_{ij} is the covariance between assets i and j.
Equation 3.15 minimizes the total variance (herein called risk measure) associated with the portfolio and is the Objective function. The first constraint, equation 3.16 is the requirement placed on the expected return of the portfolio. The second, equation 3.17 indicates that the invested amount in each portfolio was limited to a budget and that 100% of the budget be invested in the portfolio. The non-negativity constraint, equation 3.18 is set to ensure that no short sales (selling assets that you have borrowed) are made.

The model is a quadratic programming problem and it can be solved optimally using software tools. With different values of R, the above optimization problem can be solved continuously. In that instance, a set of efficient points is traced out. This efficient set is called the efficient frontier or the Pareto-Optimal frontier and is a curve that lies between the global minimum risk portfolio and the maximum return portfolio. Any portfolio that lies above or below this efficient portfolio cannot be classified as optimal and the investor sole aim is to select a portfolio that lies along this frontier.

3.6 Genetic Algorithm, (GA)

In nature, life begins with a single cell, which then divides continuously until a mature and grown individual is formed after genetic information has been reproduced. Unfit individuals however die out of the system living the fit individuals in the system. This natural selection process of simulation of survival of the fittest led to the discovery of genetic algorithm by John Holland, his students and Colleagues. Holland's original aim was to formally study the phenomenon of adaptation as it occurs in nature and to develop ways in which the mechanisms of natural adaptation might be imported into computer systems.

3.7 Basic Definitions Of Some Genetic Algorithm Terms

Below are some definitions of certain terms that are relevant to this study:

Gene - A gene is a single encodings of part of the solution space. A gene is the basic informational unit in a genotype. Each gene stands for a parameter value. A gene can be a bit, a real number, or any structure depending on the genome.

Chromosome and Genome - A chromosome is a string *set* of genes of an individual. Each chromosome is the set of parameters. A genome is the collection of all chromosomes or individuals.

Individual - An individual is any point to which the fitness function can be applied. An individual is often referred to as a genome and its corresponding vector entries are called genes.

Population - A pool *array* of individuals exhibiting equal or similar genome structures, which allows the application of genetic operators like selection, crossover, and mutation. For example, if the size of the population is 50 and the number of variables is 2, the population is represented y a 50 by 2 matrix.

Fitness Function - The fitness function is the function to be optimized. It is also known as Objective Function for standard optimization algorithms. Objective function provides a measure of performance with respect to a set of parameters. The fitness function transforms that measure of performance into an allocation of reproductive opportunities.The fitness value of an individual is the value of the fitness function for that individual.

Parents and Children - To create the next generations, the GA selects certain individuals in the current population called parents and uses them to create better individuals in the next generation called children. Normally, the algorithm is most probable to select parents that have better fitness value.

Allele and Locus - An allele is a value of specific gene can be found in a chromosome and the locus is the position where a specific gene can be found in a chromosome.

Search Space - All possible solutions to the problem is referred to as Search

Space.

3.8 How GA Works

Genetic Algorithm is implemented in a computer simulation environment in which randomly selected populations (called chromosomes) of candidate solutions (individuals or phenotypes) to an optimization problem evolves towards an optimal solution.

Traditionally, solution is encoded into binary strings of 0's and 1's, but other encodings are also possible. Each individual in a population is a point in the search space. A new population is formed with the same size by evaluating the fitness of every individual in the population and then multiple individuals are randomly selected and modified in each generation (iteration). Crossover and mutation are applied on the selected generation to make the conception of the new population. The generation formed consists of better chromosomes that are well adapted to their environment as represented by the fitness function. The new generation is then used in the next generation until a maximum number of generations are produced or a stopping criteria is met. Algorithmically, the basic GA is outlined as follows:

1. An initial population is randomly created
2. The fitness value of each member in the current population is calculated.
3. Converts the raw fitness scores to usable range of values.
4. selects parents based on their fitness value.
5. With a crossover probability P_c, perform crossover at a randomly chosen point to form offsprings.
6. With a mutation probability P_m, perform mutation at a randomly chosen point to form new offsprings.

7. Replaces the current population with the offsprings to form the new populations.

8. The algorithm stops when the stopping criteria is met else return to step 2

3.8.1 Initialization

An initial population (P_0) of the solution candidate consists of random selection of a number of individuals. Deciding the *population size* to be used for the problem is the most important aspect in choosing the initial population.

The choice of an appropriate population size has an important effect on the performance of the algorithm. Small population size will cause GA to perform poorly and this may lead to premature convergence to local maxima or minima. However, a large population size will overcome this problem. It is notable that the larger population size, the more fitness evaluations per generation calculated thereby increasing the computational time, leading to extremely slow speed of convergence. Typical population size ranges from 20 to 200. A 'definitive' population size for most optimization problems would be 50 members.

P_0 is chosen once the population size, n is decided upon. P_0 can be chosen methodically or randomly. The methodical approach allow the user to direct the population in certain areas of the search space based on the past experience of the likely optimal value. Though the methodical approach reduces variance (more effective), but it is biased because of the user defined values.The randomized method on the other hand is unbiased but less efficient because the individuals are created randomly on the search space.

3.8.2 Encoding

Encoding of chromosomes in Genetic Algorithm depends largely on the problem to be solved. Before the implementation of GA on any specific problem, a method is needed to encode potential solution to that specific problem in a form that a computer can process. Each chromosome represents a solution in the search space. Solutions are typically represented as binary strings, arrays of numbers, or strings of alphanumeric characters. These types of representations allow for multiple pieces of information concerning specific aspects of each solution to be stored all in one place. When these solutions are operated upon, each number can be individually altered in the quest to have a better solution. These strings or arrays representing candidate solutions do not necessarily have to be of fixed length although the majority of work with genetic algorithms is focused on fixed-length character strings.

There are several methods of encoding, which include: Binary Encoding, Value Encoding, Permutation Encoding, Tree Encoding etc. The most used way of encoding is the Binary Encoding because it is relatively simple and it gives many possible chromosomes even with a small number of alleles.

Binary Encoding

In Binary encoding, every chromosome is a string of bits: thus 0 and 1. The length of the string is normally determined according to the expected level of solution accuracy. Binary encoding was the first encoding to be used in GA because of its relative simplicity. However, it is often not natural for a lot of problems and corrections must be made after crossover and/or mutation.
Example of binary encoding as represented by a 4 bit binary string is as below:

Table 3.1: Binary encoding

Numeric Value	4 bit
12	[1 1 0 0]
6	[0 1 1 0]
9	[1 0 0 1]
1	[0 0 0 1]
5	[0 1 0 1]

The accuracy obtained with a 4-bit coding is $\frac{1}{16}$ of the search space. Increasing the string length by 1-bit increases to $\frac{1}{32}$

Value Encoding

To encode problems which involve complex values, such as real numbers, value encoding can be used. Every chromosomes is a sequence of certain values. The values can be anything related to the problem, such as: real numbers, characters or objects. It can be expressed as shown in Fig. 3.2

Table 3.2: value encoding

Chromosome 1	2.6872	0.2468	3.7242	5.7686	8.6342
Chromosome 2	(forward)	(back)	(left)	(left)	(right)

Permutation Encoding

Permutation encoding is appropriate in ordering problems such as the travelling salesman problem or task ordering problem. In Permutation encoding, every chromosome is a string of numbers, which symbolizesnumbers in a sequence. Example is illustrated below:

Table 3.3: Permutation encoding

Chromosome A	2	6	4	3	1	5	7
Chromosome B	1	7	2	5	6	3	4

Tree Enoding

For evolving programs or expressions, tree encoding is the best encoding technique. Every chromosome is a tree of some objects, such as functions and commands in programming lanquage.

3.8.3 Fitness Function

The fitness function is normally used to reconstruct the objective function value into a measure of relative fitness. The objective function, however, provides a

measure of how individuals have performed in the given problem domain. The fitness function is derived from the objective function and used in successive genetic operations. The performance for each candidate can be calculated by evaluating the fitness function.
The table below shows the evaluation of the fitness function $F = 3y^2 + 7$ Which is to be maximized for 5 chromosomes:

Table 3.4: Evaluation of fitness function

Chromosome	Decimal Value	Fitness
$X_1 = [1\ 1\ 0\ 0]$	$y_1 = 12$	439
$X_2 = [0\ 1\ 1\ 0]$	$y_2 = 6$	115
$X_3 = [1\ 0\ 0\ 1]$	$y_3 = 9$	250
$X_4 = [0\ 0\ 0\ 1]$	$y_4 = 1$	10
$X_5 = [0\ 1\ 0\ 1]$	$y_5 = 5$	82

From Table 3.4, X_1 has the highest fitness function value as compared to X_4 which had the least fitness value.

Fitness function therefore quantifies the optimality of a solution (chromosome) so that a peculiar solution may be stratified against all the other solutions. The function depict the closeness of a given 'solution' to the desired result.

3.8.4 Selection

The Selection Operator is also called the Reproduction Operator. The same number of chromosomes is chosen in the parent population to create the next generation. According to Darwin's theory of evolution "survival of the fittest" - best individuals do survive and create new offspring, hence the better-performed chromosomes are chosen to continue the generation so as to have better offspring. Selection has two main objectives (Popov, 2005):

- To provide chance to an individual or solution with comparatively poor fitness function value to be part in the creation process of the next generation, thereby perpetuating the global character of the search process. This prevents a single individual from dominating the population and thus brings it to local extremum.

- To select individuals (chromosome or solution) with higher fitness value, which will form part of the generation of the next population.

The selection algorithm should achieve zero bias whilst maintaining a minimum spread and not contributing to an increased time complexity of the GA. The difference between the reproduction probability of a low fitness individual and a high fitness in a given selection strategy is called the *selection pressure*. If the selection pressure is too high, there is the propensity that the population will be dominated with few individuals that are not optimal but have a higher fitness value than the rest of the current population, thereby leading the search to local optima. However, if the selection pressure is too low, the search will move too randomly through the search space (Yao et al., 1999).

The selection can be done by many methods, such as roulette wheel selection, Boltzman Selection, Tournament Selection, Steady State Selection and Rank Selection.

Roulette Wheel Selection

Roulette wheel selection is also known as the Fitness Proportionate Selection. Roulette Wheel Selection is the most used selection. It mimics a roulette wheel game, which includes 37 coloured and numbered pockets on the wheel and a small marble throwned to choose a number randomly.

Each chromosome has a sector in the roulette wheel of a size, which is proportional to the fitness value of the given chromosome. That is the chromosome, which has a higher evaluation, takes a larger area of the sector in the wheel. The probability of each individual to be selected is calculated as the proportion of its function to the sum of the fitness unction of all individuals in the current generation. The basic roulette wheel selection method is Stochastic Sampling with Replacement (SSR). The segment size and selection probability of SSR remain constant throughout the selection phase and selection of individuals' are made according to the individuals' fitness value.

This selection technique works in such a way that more fit individuals (Chromosomes) is more likely to be chosen but not guaranteed. The reproduction probability for each individual (chromosome) is calculated as:

$$P_i = \frac{f_i}{\sum_{i=1}^{n} f_i}$$

each time a single chromosome is selected for the new population. Where f_i is the fitness of the individual i, n is the size of the population.
This is achieved by generating a random number, r from the interval $[0, 1]$. if $r \leq p$ then select the first chromosome, otherwise select the i_{th} chromosome such that $P_{i-1} \leq r \leq P_i$. The Expected count= $n \times P_i$.

Consider Table 3.4

$$\sum_{i=1}^{5} f_i = 439 + 115 + 82 + 50 + 250 = 896$$

$$P_1 = \frac{439}{896} = 0.49\,; \quad P_2 = \frac{115}{896} = 0.13\,; \quad P_3 = \frac{82}{896} = 0.09$$

$$P_4 = \frac{10}{896} = 0.01\,; \quad P_5 = \frac{250}{896} = 0.28$$

The above can be summarised in the table below:

Table 3.5: Selection of best chromosome

Chromosome	Decimal Value	Fitness	P_i	Expected Count$(n \times P_i)$
$X_1 = [1\ 1\ 0\ 0]$	$y_1 = 12$	439	0.49	2.45
$X_2 = [0\ 1\ 1\ 0]$	$y_2 = 6$	115	0.13	0.65
$X_3 = [1\ 0\ 0\ 1]$	$y_3 = 9$	250	0.09	0.45
$X_4 = [0\ 0\ 0\ 1]$	$y_4 = 1$	10	0.01	0.05
$X_5 = [0\ 1\ 0\ 1]$	$y_5 = 5$	82	0.28	1.40
Total		896	1.00	5.00

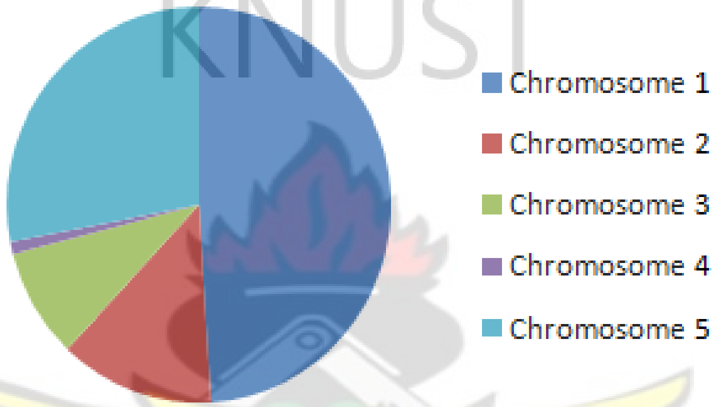

Figure 3.3: A diagram of the Roulette Wheel Selection

Tournament Selection

Tournament selection is a selection type that require running several ‘tournaments’ among few individuals chosen at random from the population. The individual with the best fitness is selected for crossover. If there is a larger tournament size, weak individuals will have a smaller chance to be selected. The steps below is the pseudocode for tournament selection:

1. Randomly choose n (tournament size) individuals from the population.

2. The best individual from the tournament is chosen with probability,P.

3. The second best individual with probability $P(1 - P)$ is chosen

4. Choose the third best individual with probability $P((1 - P)^2)$

Tournament selection while slower and more complicated, it is efficient to code and allows the selection pressure to be easily adjusted.

Rank-Based Selection

Unlike the roulette wheel selection in which the best individual with the best fitness value has a higher chance of been selected , individuals under rank-based selection has equal chances of been selected. Individuals are sorted according to the value of their fitness function and are then assigned a rank. The probability that individual j is selected is inversely proportional to its position in the sorted list. The selection probabilty for each individual is calculated according to the following non-linear function: $P = \beta(1-\beta)^{rank-1}$, where β is a user defined co-efficient.

Baker (1985), proposed a linear ranking method that each individual in the population is ranked in increasing order of fitness, from 1 to N. The user then chooses the expected value Max of the individual with rank N, with $Maxe_o$. The expectedd value of each individual, I in the population at time t is given by:

$$ExpVal(i,t) = min + (max - min)\frac{rank(i,t)}{N-1}$$

Where Min is the expected value of the individual with rank 1. Baker recommended that $Max = 1.1$. Rank selection can lead to slower convergence because the best chromosomes do not vary much from the others.

Boltzman Selection

A selection approach similar to simulated annealing in which a continuosly varying temperature controls the rate of selection according to a present schedules. Selection pressure is low indicating high temperature at start which signifies that every individual is capable of reproducing. The temperature is successively lowered implying an increase in selection pressure thereby allowing GA to narrow into the best part of the search space whilst perpetuating a suitable degree of diversity.

Simulated annealing is a method used to minamize or maximize a function. The method feign the process of slow cooling of molten metal to attain the mimimum function value in a minimization problem. The process of cooling is simulated by controlling a temperature like a variable introduced with the idea of Boltzman probability distribution. The system in thermal equilibrium at a temperature, T has its energy distribution based on the probability defined by $P(E) = exp(-K/KT)$ where k is boltzman constant. The above expression propose that a system at a higher temperature has almost uniform probability at any energy state but at a lower temperature, the tendency of beeing at a higher energy state is low. To control the convergence of the algorithm, the temperature T must be controlled vis avie the assumption that the search process follows a Boltzman probability.

Steady State Selection

The basis of this selection type is that large part of individuals should survive to the next generation. In every generation, adequately small individuals with high fitness value are selected in creating new offsprings. Th individuals with low fitness values are then removed and the new offspring is placed in their place. The rest of the population survives to new generation. Steady states GA are normally used in developing rule-based sytems (e.g. Classifier systems) in which remembering the already learnt is important and members of te population mutually (rather than independently) solve the given problem.

Elitist Selection

Elitist Selection is a selection strategy where a limited number of individual with the best fitness values are chosen to pass on to the next generation avoiding the crossover and mutaton operators. Elitism prevents the random destruction by crossover or mutation operators of individuals with good genetics. The number of elite individuals should not be too high, otherwise the population will tend to degenerate.

3.8.5 Crossover

Crossover is sometimes referred to as Recombination. Crossover is an evolution process in which two parent chromosomes combine to form one offspring chromosome containing trait of each parent.
The first step in the reproduction process is the crossover. The genes of the parents are used to form entirely new chromosomes from crossover by combining the information extracted from the parents (two chromosomes). The typical recombination for the GA is an operation requiring two parents, however other schemes with more than two parents are also available. Our work will be based on two parents. The basis of Crossover is to show that new chromosomes might be better than both of the parents if it takes the best trait from each of the parents.

The Crossover rate (P_c) is the probability that the Crossover operator will be applied to a specific chromosome during a generation. This factor is set fairly high and the typical values of P_c are in the interval $[\,0.5,\ 1\,]$. Whenever a Crossover probability is used, $100P_c$ percent strings in the solution (population) are used in the Crossover operation, and $100(1 - P_c)$ percent of the remaining population remains as they were in the current population. The importance of Crossover is not actualized when the population size is made to be small. Hence the need to create a higher population size to improve the efficiency of the Genetic Algorithm search, which results in optimal performance.

The Crossover operator is of many types: One-Point Crossover, Two-Point Crossover, Uniform Crossover, Arithmetic Crossover, Heuristic Crossover, and Intermediate Crossover.

One-Point Crossover (Single-Point Crossover)

The most simplest form of crossover type. In one-point crossover, a single crossover point on both parents' chromosome strings is selected. All data beyond that point in either chromosome string is interchanged between the two parent chromosomes. The resulting individuals are the offsprings. The crossover would then look as shown below:

Consider the two parents selected for single crossover,

Parent A [1 0 1 | 1 0 0 1 1 1]

Parent B [0 1 1 | 0 1 0 0 0 1]

Interchanging the parent chromosomes after single crossover produces the offspring below:

Offspring A [1 0 1 | **0 1 0 0 0 1**]

Offspring B [0 1 1 | **1 0 0 1 1 1**]

| is the chosen crossover point.

Two-Point Crossover

Two-Point crossover randomly selects two-points on the parent chromosome strings. Everything between the two points is swapped between the parent chromosomes giving two offsprings. Two-Point crossover reduces bias and is less likely to obstruct schemas with large defining lengths and can combine more schemas than single-point crossover.

Consider the two parents selected for two-point crossover:

Parent A [1 0 | 1 0 0 | 1 1 1 0]

Parent B [0 1 | 0 1 0 | 0 0 1 0]

Interchanging the parent chromosomes between the crossover points, the offspring produced are:

Offspring A [1 0 **0 1 0** 1 1 1 0]

Offspring B [0 1 **1 0 0** 0 0 1 0]

Three-Point Crossover

A crossover operator in which offspring is derived from three parents. They are randomly chosen. Each bit of first parent is verified with bit of second parent whether they are same. If same, then the bit is taken for the offspring. The following three parents:

Parent A [1 1 0 1 0 0 1]

Parent B [1 0 0 0 1 1 0]

Parent C [0 1 1 0 0 1 0]

produces the offspring below:

Offspring [1 **1** 0 **0** **0** 1 **0**]

Multi-Point(N-Point) Crossover

Two ways in this crossover is observed. These are even number of cross-sites and odd number of cross-sites. Whilst cross-sites are picked randomly around a circle and information is exchanged in even number cross-sites, a different cross-point is always assumed at the beginning of the string.

Arithmetic Crossover

A crossover operator that linearly combines two parent chromosome strings to produce two new offspring according to the following equations:

$Offspring\ A = \alpha * Parent\ A + (1 - \alpha) * Parent\ B$

$Offspring\ B = (1 - \alpha) * Parent\ A + \alpha * Parent\ B$

Where α is a random weighting factor chosen before each crossover operator.

Consider two parents (each of 3 float genes) selected for crossover

Parent A (0.6) (0.8) (0.3)

Parent B (0.2) (0.5) (1.6)

Applying the above equations and assuming the weighting factor, $\alpha = 0.8$, two resulting offsprings are produced. The possible set of offsprings after arithmetic crossover would be:

$Offspring\ A$ (0.52) (0.74) (0.56)

$Offspring\ B$ (0.28) (0.56) (1.34)

Heuristic Crossover

A type of crossover that uses the fitness values of the two parent chromosomes to ascertain the direction of the search. The offspring are created according to the equation below:

$Offspring\ A = BestParent + \beta * (BestParent - WorstParent)$

$Offspring\ B = BestParent$

Where β is a random number between 0 and 1

It is practicable that Offspring A will not be feasible. This is possible if β is chosen such that one or more of its genes will fall outside of the allowing lower and upper bounds. In compliance with this, heusristic crossover has a user settable parameters (n) for the number of times used and a β that results in a feasible chromosomes. If a feasible chromosome is not produced after n times, the worstparent is returned as offspring A.

Uniform Crossover

The Uniform crossover uses a fixed mixing ratio between two parents. Contrary to one-point and two-point crossover, the uniform crossover enables the parent chromosomes to contribute to the gene level rather than segment level with some probability-known as the mixing ratio.

Consider the two parents selected for crossover:

Parent A [1 0 1 0 0 1 1 1 0]

Parent B [0 1 0 1 0 0 0 1 0]

If the mixing ratio is exactly 0.5, then half of the genes in the offspring will come from Parent A and other half will come from Parent B. The feasible set of offspring after Uniform crossover would be:

Offspring A [**0** 0 1 0 0 **0** 1 1 0]

Offspring B [0 1 0 1 0 0 **1** 1 **1**]

In single or double point crossover, genomes that are near each other tend to survive together, whereas genomes that are far apart tend to be separated. The technique used here eliminates that effect. Each gene has an equal chance of coming from either parent. This is sometimes called **Scattered or random crossover**.

Intermediate Crossover

The mathematical representation of this crossover is:

$C_1 = \gamma.P_1 + (1-\gamma).P_2$

$C_2 = (1-\gamma).P_1 + \gamma.P_2$

$\gamma = (1+2.\alpha).r - \alpha$

P_1, P_2 are the chromosome of the parents, C_1, C_2 are the chromosomes of the offspring , α is the exploration coeeficeient which is user defined ($\alpha \geq 0$)

The coeficient α permits the user to change the area in which the value of the offspring gne can appear. When α is above 0, it is asured that the value of the resultant gene is between the values of the corresponding genes of the parents.

Partially Matched Crossover (PMX)

This crossover type is used for ordered list of chromosomes, an ordered list of the cities to be travelled for a travelling salesman problem. Unlike the N-Point crossover, this type of crossover preserves a given order in the chromosome. The crossover points are selected at random and PMX proceeds by position wise exchanges. The two crossover points give matching selection. It affects cross by position-by-position exchange operations. Parents are mapped to each other, hence it can also be called *partially mapped crossover*.

3.8.6 Mutation

Mutation of chromsomes takes place after crossover is performed. Mutation is a genetic operator used to perpetuate genetic diversity from generations to generations.It alters one or more gene value in a chromosome from its initial state. This result in entirely new gene values being added to the gene pool. Having obtained these new gene values, its then realizable for GA to arrive at an optimal solution than was previously possible.

Mutation occurs during evolution according to a user-definable mutation probability, P_m usually set to reasonably low value, say 0.01. If the P_m is set to be high, the search will turn into a primitive random search but if P_m is set to be low, there is a danger of premature convergence. Mutation is an important part of the GA process because it helps to prevent the population from stagnating at any local optima. To prevent premature convergence of poorly bhaved search space, higher mutations rate are required. Most mutations will rapidly die out, while the few mutations that gives valuably new information will be assimilated into the population.

The Mutation operator is of many types: Flip-Bit, Boundary, Uniform, Non-Uniform and Gaussian Mutation.

Flip Bit

A mutation operator that inverts the value of the chosen gene (0 goes to 1 and 1 goes to 0). This mutation operator can only be used for binary genes. Consider the two original offsprings selected for mutation.

Original Offspring 1 [1 1 0 1 1 0 0 0 1]

Original Offspring 2 [0 1 0 1 0 1 1 0 1]

Invert the value of the chosen gene as 0 to 1 and 1 to 0.

The mutated offspring produced are:

Mutated Offspring 1 [1 1 0 1 1 0 **1** 0 1]

Mutated Offspring 2 [**1** 1 0 1 0 **0** 1 0 1]

Boundary

A mutation operator that replaces the value of the chosen gene with either the lower or upper bound for that gene (chosen arbitrary). This mutation operator can only be used for integer and float genes.

Uniform

It replaces the value of the chosen gene with a uniform arbitrary values selected between the user-specified lower and upper bounds for that gene. It is also used for integer and float genes only. It creates the mutated children using uniform mutations at multiple points. Mutated genes are uniformly distributed over the range of the gene. The new value is not a function of the parents value for the gene.

Non-Uniform

This operator increases the probability such that the amount of mutation will be near to 0 as the generation number increases. It prevents the population from converging in the early stages of the evolution then allows GA to fine tune the solution in the later stages of evolution. This operator can only be used for integer and float genes.

Gaussian

This mutation operator can only be used for integer and float genes. It adds a unit gaussian distributed random variables to the chosen gene. The new gene value is clipped should it fall outside the user-specified lower or upper bounds for that gene.

3.8.7 Search Termination

It is difficult to specify the convergence criteria of GA because of its stochastic nature. As the fitness of a population may remain constant for a number of generations before a superior individual is found, the application of conventional termination criteria becomes difficult. Most common stopping criteria are:

1. Fixed number of generations reached
2. Computation time is reached
3. A solution is found that satisfies the minimum criteria
4. No improvement in solutions for a specified number of generations.

3.9 Schema Theorem

A schema is a set of binary strings that match the template for schema **H**. A template is made up of 1s, 0s, and *s where * is the 'dont care' or 'wildcard' symbol that matches either 0 or 1. For instance, the schema H=10*1* represents the set of binary strings: 10010, 10011, 10110, 10111. The number of fixed bits in the template is called the order of a schema represented as $o(H)$. If $H = 10*1*$ then $o(H) = 3$. The defining length, $\delta(H)$ is the distance between its first and the last non '*' gene in Schema H. If $H = *1*01$ then $\delta(H) = 5 - 2 = 3$. if $H = 0****$ then $\delta(H) = 1 - 1 = 0$. Suppose x is an individual that belongs to the schema H, then we say that x is an instance of $H(x \in H)$.

The Schema theorem was proposed by John Holland in the 1990's. Schema theorem is widely taken to be the foundation for the working principles of GA. In Schema theorem, the search space is partitioned into subspaces of different degrees of generality, and models are formulated to estimate the expected growth in the next generation from the number of individuals in the population. The basis of the Schema theorem is that short, low-order schemata with above average fitness increase exponentially in successive generations.

The generalized Schema theorem, considering proportionate selection, crossover and mutation is defined as;

$$E[m(H,k+1)] \geq m(H,k)\frac{f(H,k)}{\bar{f}(k)}\left(1-p_c\frac{\delta(H)}{l-1}\right)(1-p_m)^{o(H)}$$

Where $E[m(H,k+1)]$ is the expected number of chromosomes that represent the schema H at generation k+1, $m(H,k)$ denote the number of chromosomes (solution candidates) in generation k that has Schema H , $f(H,k)$ is the average fitness score evaluated from the objective function of chromosomes belonging to Schema H at generation K, $\bar{f}(k)$ is the average fitness of H in the Kth generation, p_c is the crossover probability, p_m is the mutation probability, $\delta(H)$ is the defining length and $o(H)$ is the order of Schema H.
The average fitness value of the individual, $f(H,k) = \frac{\sum_{x\in H} f(x)}{m(H,k)}$. The term $m(H,k)\frac{f(H,k)}{\bar{f}(k)}$, $\{1-p_c\frac{\delta(H)}{l-1}\}$ and $(1-p_m)^{o(H)}$ account for the proportionate selection, crossover and mutation operation respectively.

3.10 Building Block Hypothesis (BBH)

BBH states that GA works best when short, low-order, highly fit schemas recombine to form even more highly fit higher-order schemas (Goldberg, 1989b). The capacity to create fitter and fitter partial solutions by merging building blocks is considered to be the main source of GAs search power.
The building block hypethesis is composed of :

- An explanation of a heuristic that carry out adaptation by identifying and recombining “building blocks”, thus short, low-order, highly fit schemas.
- A theorem that GA performs adaptation by implicitly and effetively implementing this heuristic.

Highly fit schemata are sampled and crossed over. The crossover schemata are then sampled to form strings of potentially higher fitness. In this way, high performing strings are built from the best partial solutions of past samplings.

GA however seeks near optimal performance through justaposition of short, low-order, high-performance schemata or building blocks (Goldberg, 1989a).

Chapter 4

Analysis and Results

4.1 Introduction

In this Chapter, we formulate an objective function to solve the portfolio selection problem. Three types of crossovers (Arithmetic, Heuristic and Uniform) are applied on this objective function to select the optimal portfolio and the appropriate crossover.

4.2 Portfolio Selection Model

Suppose an investor wants to allocate his/her wealth among n risky securities based on historical data and he/she wants to minimize the risk under a given level of portfolio return subject to some specified constraints: 100% of the budget of the investor be invested in the portfolio, exactly K number of assets are held, a lower and upper limit which ensures that if any asset i is held, its proportion, W_i must lie between the upper and lower limit.

We assume that historical data for n stocks are obtained at period, T. These historical data obtained are the various closing prices of the various asset i at period t, where $t = 1, 2, 3, ..., T$. The return for an asset in the model which is given as its logarithmic return value relates the closing price of the asset at periods $t-1$ to t (represented by P_n^t and P_n^{t-1}) as:

$$r_n^t = \log(\frac{P_n^t}{P_n^{t-1}})$$

We propose a model for our work to include practical constraints. We introduce a risk aversion parameter λ $(0 \leq \lambda \leq 1)$ that describes the sensitivity of the investor to risk. Our propose model is:

$$\text{Minimize} \quad \lambda \left[\sum_{i=1}^{n} \sum_{j=1}^{n} W_i W_j \sigma ij \right] - (1-\lambda) \left[\sum_{i=1}^{n} W_i \mu_i \right] \tag{4.1}$$

Subject to

$$\sum W_i = 1 \tag{4.2}$$

$$\sum z_i = k \tag{4.3}$$

$$\varepsilon_i z_i \leq W_i \leq \sigma_i z_i, \quad i = 1, 2, 3, \cdots, n. \tag{4.4}$$

The Objective function is to be modeled to search for the less performed solution on the fitness scale, implicating that the procedure with least objective function should lead to better solution.

$\lambda = 0$ represents maximum expected return and $\lambda = 1$ represents minimum risk, thus the investor is very conservative. Values of λ that satisfy $0 < \lambda < 1$ represent an excellent tradeoff between risk and return, that generates solutions between the two extremes. The objective function allows the selection of efficient portfolios. The selection of λ by the Portfolio manager is equivalent to specifying his attitude towards the risk or in the broad terms his utility function. The greater the factor, λ the more risk aversed the investor is.

Cardinality constraint is introduced in Equation 4.3. It ensures that exactly K assets are held. This constraint is also imposed to facilitate the portfolio management and to reduce its management costs. A model of this type of constraint is called "the asset paring problem". It ensures that exactly K-assets are held in the portfolio.

Floor-Ceiling constraint, equation 4.4 define a lower and upper limit, ε_i and σ_i respectively which can be allowed to be held for each asset in the portfolio. It ensures that if any of assets i is held $(z_i = 1)$, its proportion W_i must lie between ε_i and σ_i, while if none of assets i is held $(z_i = 0)$, its proportion $W_i = 0$. Thus, we must have $0 \leq \varepsilon_i \leq \sigma_i \leq 1 \quad (i = 1, 2, \cdots, N)$. Floor constraints (lower bound) is introduced to avoid administration cost for very small holdings while ceiling constraints (upper bound constraints) are introduced in the model to avoid excessive exposure to a specific asset as part of institutional diversification policy. Introducing zero-one decision variables:

$$z_i = \begin{cases} 1, & \text{if any asset } i \text{ is held,} \\ 0, & \text{otherwise.} \end{cases} \tag{4.5}$$

4.2.1 Portfolio Representation

We use vector of real valued variables for each weight in the portfolio to represent the portfolio in our GA. The real-valued representation has the benefit of directly representing the portfolio. Hence the transformation from the genome to the problem solution is very straightforward. The output for the system is a portfolio which comprises of 5 weights that will be allocated to 5 stock . These weights is restricted: $\sum_{i=1}^{5} W_i = 1$ and $0 \leq W_i \leq 1$ (thus, the total weights should not exceed the total resources and weights must be non-negative). The '$0'$' represent the floor constraint and the '$1'$' represent the ceil constraint.

4.2.2 Optimization Model: Genetic Algorithm Specification

A random population size of 50 chromosomes were generated and each describes an asset. A vector of real valued variables are used to represent the chromosomes. Elitism was set to 3 of the most fittest chromosomes.
A roulette wheel selection was used in selecting the chromosomes. 3 types of crosssover: Arithmetic, Heuristic, and Uniform were used. Crossover and Mutation fraction was set to 0.9 and 0.01 respectively and adaptive feasible mutation function was used. The feasible region is bounded by the constraints.

A plot interval of 5 was used. The stopping criteria is set to a maximum of 100 generations reached or the best chromosome. A risk aversion parameter, λ of 0.5 is used.

4.3 Simulation and Presentation of the Results

We randomly selected 5 companies data from the Ghana's Stock Exchange (GSE) and used it to demonstrate the efficiency of heuruistic crossover in the selection of an optimal portfolio. The objective of the fitness function in the GA method is set to minimize the volatility and maximize the return of the portfolio, and as a result the value that scores less on the fitness scale led to the optimal solution.

4.3.1 The Data

The historical prices from a 5 stock portfolio for a period of 6 years, $2007-2012$ is taken. These historical prices consist of the daily closing prices (the last value for any specific day) of the various stocks. Their corresponding historical return and covariance is calculated as the input data. These historical data are taken from companies that have different economic characteristic. The portfolio variance and the portfolio average return is calculated using the historical data below:

Table 4.1: Portfolio annual average returns

Year	STOCK 1	STOCK 2	STOCK 3	STOCK 4	STOCK 5
2012	-0.002447224	0.000543547	0.000324513	0.00071918	0.000225605
2011	1.36E-05	-6.17E-05	-0.00071292	-0.002628853	-0.000658221
2010	0.000764859	0.000689607	-4.27E-12	0.00119565	0.002395387
2009	-0.000431354	-0.002004711	-4.27E-12	0.000382691	-0.000723365
2008	0.000661889	0.000533048	-0.001825326	0.001103673	0.000174978
2007	0.001245823	0.0016855	0.000340635	0.000678282	0.001297666

For each asset i, the mean (μ_i) and the variance (σ_i^2) is calculated. The covariance with other assets j, (σ_{ij}) is also computed. For each portfolio, p the mean

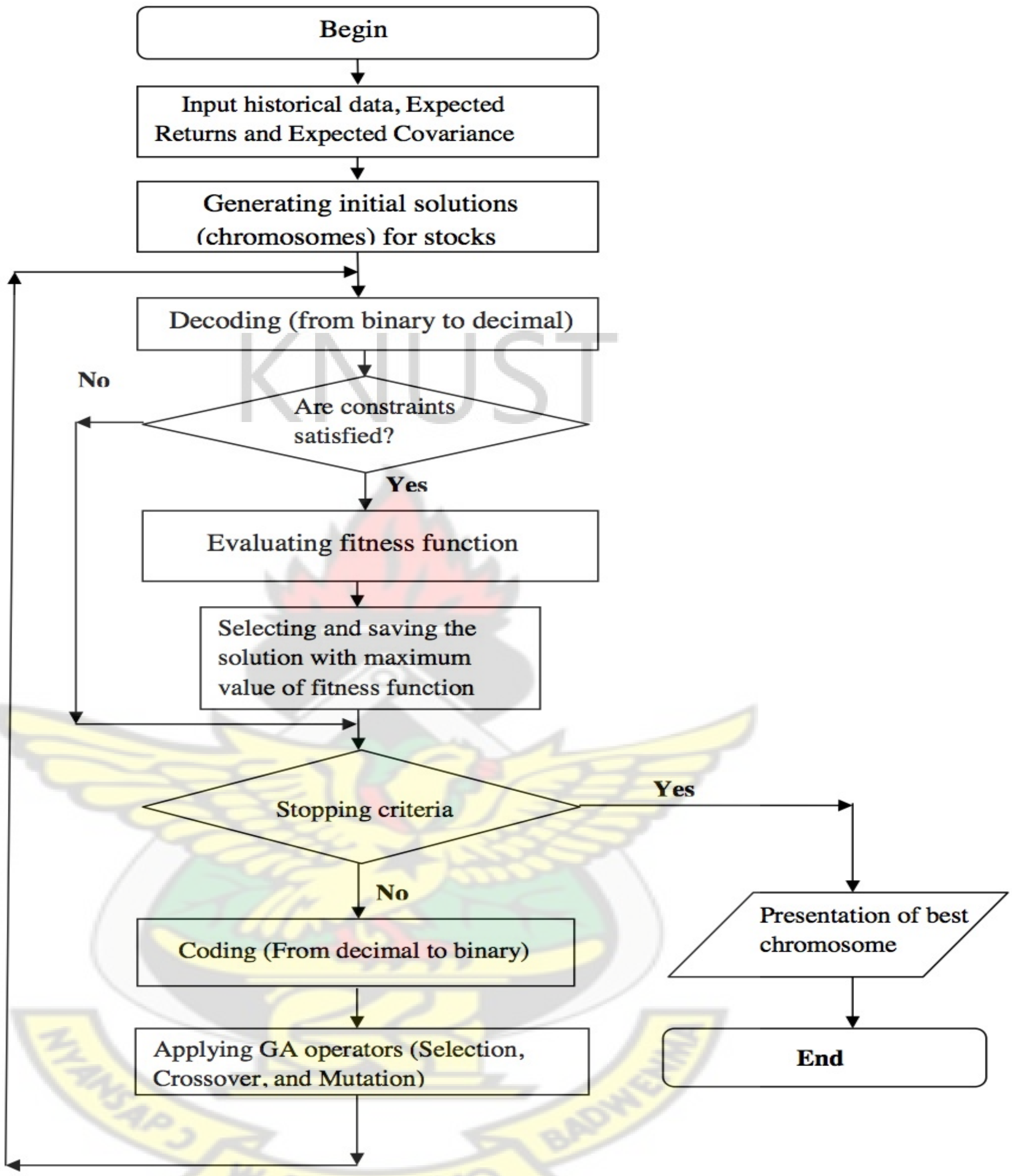

Figure 4.1: Flowchart of GA based portfolio optimization algorithm

(μ_p), covariance, (σ_{ij}) and the variance ($\sigma^2(r_p)$) are calculated.

The covariance matrix, variance and mean return for each asset portfolio are given in the tables below:

Table 4.2: The Covariance matrix

	STOCK 1	**STOCK 2**	**STOCK 3**	**STOCK 4**	**STOCK 5**
STOCK 1	1.75E-06	5.12E-07	-1.69E-07	1.15E-07	6.81E-07
STOCK 2	5.12E-07	1.52E-06	9.47E-07	4.02E-07	9.85E-07
STOCK 3	-1.69E-07	9.47E-07	1.11E-06	9.88E-08	7.73E-07
STOCK 4	1.15E-07	4.02E-07	9.88E-08	2.07E-06	9.80E-07
STOCK 5	6.81E-07	9.85E-07	7.73E-07	9.80E-07	1.45E-06

Table 4.3: The Variance for each asset

	STOCK 1	**STOCK 2**	**STOCK 3**	**STOCK 4**	**STOCK 5**
Variance	1.75E-06	1.52E-06	1.11E-06	2.07E-06	1.45E-06

Table 4.4: The mean returns, r_i for each asset

	STOCK 1	**STOCK 2**	**STOCK 3**	**STOCK 4**	**STOCK 5**
mean return	-3.12E-05	0.00023167	-0.00064554	0.00024283	0.00045283

4.3.2 The Result

Through simulations, 3 different crossovers are applied on our objective function. All simulations in this work were executed using Matrix Laboratory, MATLAB (MATLAB is a product of The Mathworks, Inc.) version R2012a. MATLAB is an application with tools for numerical computation and a fourth-generation programming language with data visualization; serving as a accessible "laboratory" for computations and analysis. GA method, which is a stochastic technique (based on the use of random numbers) forms the basis of these simulations. Illustrated below are the results obtained via the 3 crossover techniques:

Arithmetic Crossover

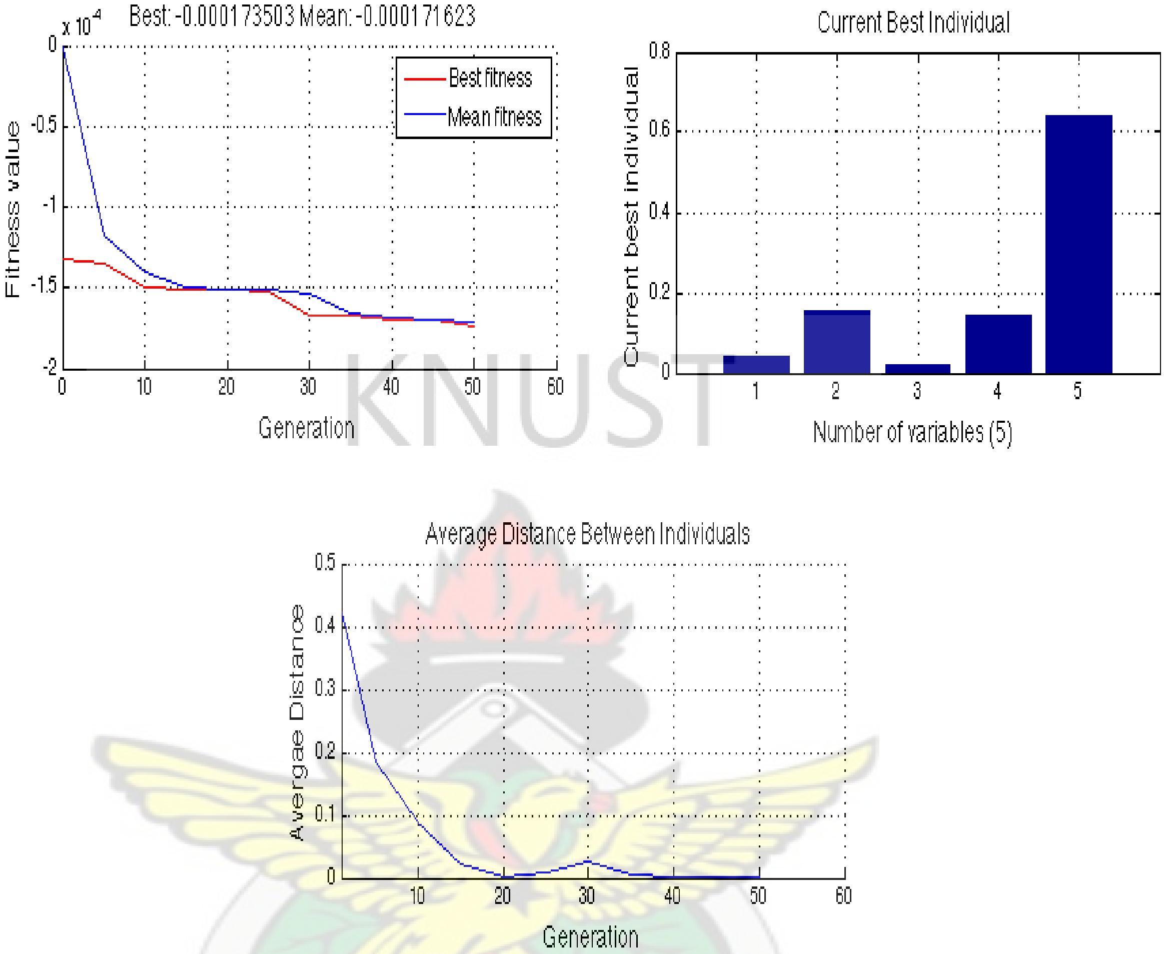

Figure 4.2: The variations of the GA functions according to generation under Arithmetic crossover technique

Objective function value = -1.735049154920278e-04

Average Return of Portfolio = 3.481543945314851e-04

Variance of Portfolio = 1.144563547429597e-06

Portfolio Weights :

Weight 1= 0.038056758381922

Weight 2= 0.155418010272420

Weight 3= 0.019858002318151

Weight 4= 0.144597019644021

Weight 5= 0.642712653783081

Heuristic Crossover

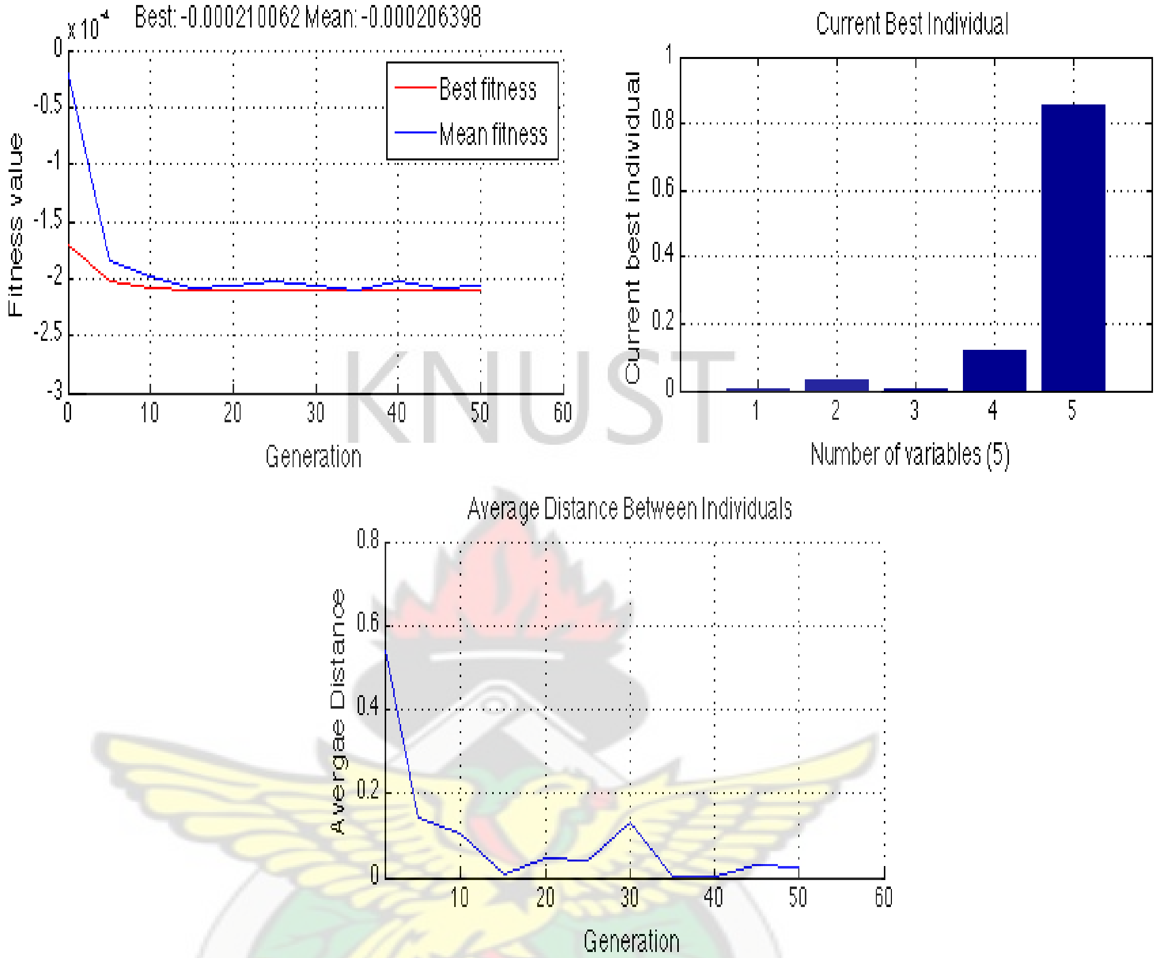

Figure 4.3: The variations of the GA functions according to generation under Heuristic crossover technique

Objective function value = -2.100619086026940e-04

Average Return of Portfolio = 4.214555303246128e-04

Variance of Portfolio = 1.331713119224894e-06

Portfolio Weights :

Weight 1= 0.000150118405472

Weight 2= 0.029845672108102

Weight 3= 0.000000003905862

Weight 4= 0.117642837123397

Weight 5= 0.852361368457167

Uniform/Scattered Crossover

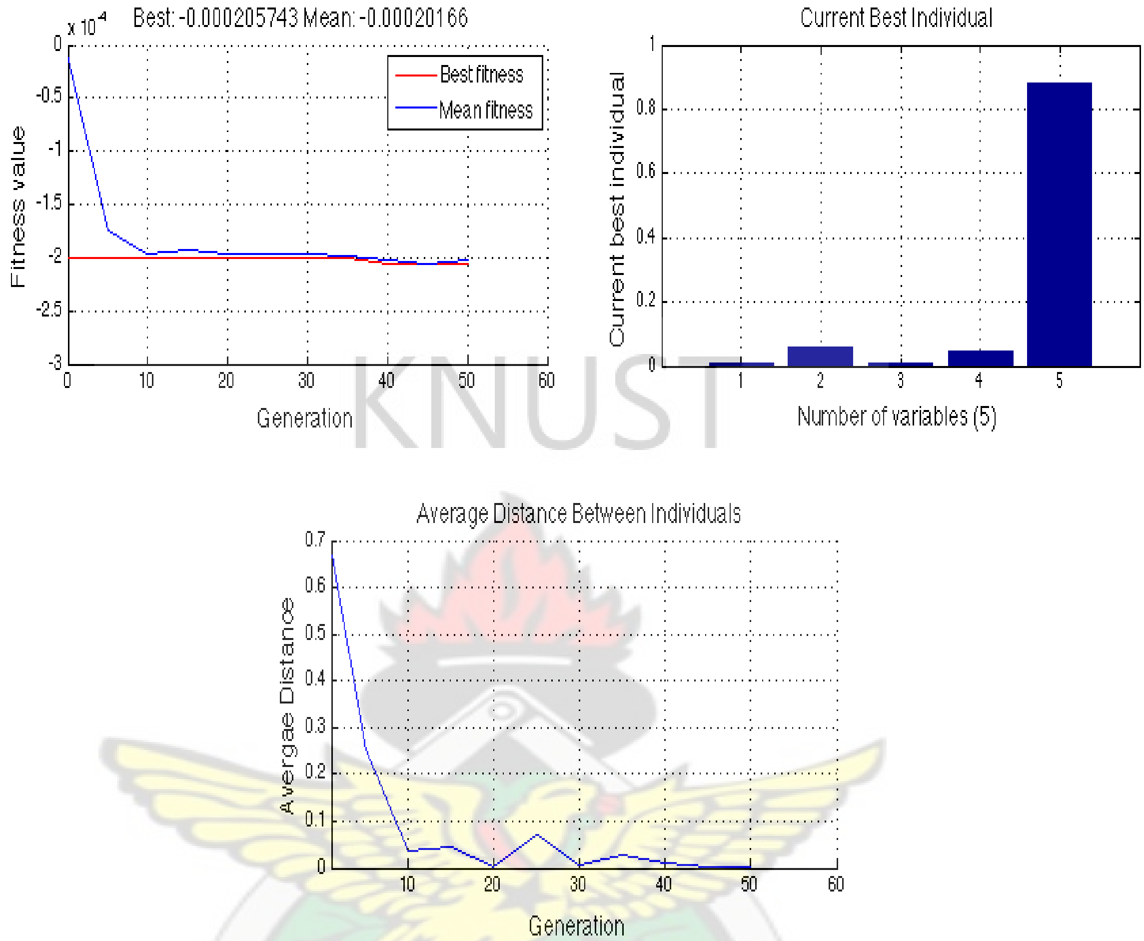

Figure 4.4: The variations of the GA functions according to generation under uniform crossover technique

Objective function value = -2.057425150293183e-04

Average Return of Portfolio = 4.128158217522129e-04

Variance of Portfolio = 1.332791693576277e-06

Portfolio Weights :

Weight 1= 0.010833630409913

Weight 2= 0.055925224208046

Weight 3= 0.012315752652018

Weight 4= 0.043848245554371

Weight 5= 0.877805105388565

4.3.3 Analysis

For the implementation of Genetic Algorithm, an objective function was formulated to evaluate which among the three crossover techniques (thus, Arithmetic, Heuristic and Uniform) score less on the fitness scale, and accordingly should lead to the optimal portfolio.

With regards to our fitness function value, the Heuristic crossover led to the best choice of weights, return and risk. The Heuristic crossover scores less on the fitness scale with $-2.100619086026940e-04$, whereas the Arithmetic and Uniform scored $-1.735049154920278e-04$ and $-2.057425150293183e-04$ respectively.

The Heuristic crossover had the highest return of $4.214555303246128e-04$ and a corresponding weights ($W_1 = 0.000150118405472, W_2 = 0.029845672108102, W_3 = 0.000000003905862, W_4 = 0.117642837123397, W_5 = 0.852361368457167$). An investor wishing to get this maximum return and its associated lower risk should invest 0.02% of his total wealth in Stock 1, about 2.98% of his total wealth in Stock 2, 11.76% of his total wealth in Stock 4, and 85.24% of his total wealth in Stock 5 and can choose not to invest in stock 3.

The Arithmetic crossover had the lowest return of $3.481543945314851e-04$ and the lowest risk of $1.144563547429597e-06$. To get the above return and risk, the allocation of weight to each stock is: $W_1 = 0.038056758381922, W_2 = 0.155418010272420, W_3 = 0.019858002318151, W_4 = 0.144597019644021, W_5 = 0.642712653783081$.

In Uniform crossover, a proportion of weights ($W_1 = 0.010833630409913, W_2 = 0.055925224208046, W_3 = 0.012315752652018, W_4 = 0.043848245554371, W_5 = 0.877805105388565$) should be allocated respectively to each of the five (5) stocks respectively in order to get a return of $4.128158217522129e-04$ and a risk of $1.332791693576277e-06$.

Chapter 5

Conclusion and Recomendation

5.1 Conclusion

We applied Arithmetic, Heuristic, and Uniform crossover on portfolio of stocks from Ghana Stock Exchange which includes practical constraints (floor-ceil and boundary constraints). The results we obtained showed that the Heuristic crossover gives better results than the two other crossovers with a maximum return $4.214555e-04$ and a minimum risk of $1.331713e-06$. Our results show that Heuristic crossover is very useful when an investor wants to allocate his total wealth in an investment to yield a maximum return and lesser risk.

The efficiency and robustness of genetic algorithm is also confirmed. A disclosure in this research was the flexibility of the GA to produce the solution to the problem using the different crossover types.

5.2 Recomendation

As an efficient optimization tool for portfolio selection, heuristic crossover is very useful when an investor wants to allocate his/her total wealth on a portfolio to yield a maximum return and lesser risk. A further research is needed to compare the results of GA methods with regards to different selection strategies using the same model.

REFERENCES

Aftalion, T. (2012). Genetic algorithms for portfolio optimization. In *Graduate Artificial Intelligence*.

Aliev, R., Abiyev, R., and Menekay, M. (2013). Fuzzy approach to portfolio selection using genetic algorithms. *Intelligent Automation and Soft Computing*, 14(4):525–540,.

Anagnostopoulos, K. P. and Mamanis, G. (2008). A multiobjective genetic algorithm for portfolio selection with integer constraints. *SPOUDAI*, 58(1-2):185–200.

Baker, J. H. (1985). Adaptive selection methods for genetic algorithms. *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, pages 100–111.

Chang, T.-J., Meade, N., Beasley, J., and Sharaiha, Y. (1999). Heuristics for cardinality constrained portfolio optimisation. *Computers and Operations Research*, 27 (2000):1271–1302.

Chang, T.-J., Yang, S.-C., and Chang, K.-J. (2009). Portfolio optimization problems in different risk measures using genetic algorithm. *Expert Systems with Applications*, 36:10529–10537.

Divya, P. and Kumar, P. R. (2012). The investment portfolio selection using fuzzy logic and genetic algorithm. *International Journal of Engineering Research and Applications*, 2(5):2100–2105.

Goldberg, D. E. (1989a). Genetic algorithms in search, optimization, and machine learning. *Reading, MA:Addison-Wesley*.

Goldberg, D. E. (1989b). Sizing populations for serial and parallel algorithm. *Pro-*

ceedings of the Third International Conference on Genetic Algorithms, pages 70–79.

Hachloufi, M. E., Guennoun, Z., and Hamza, F. (2012). Stocks portfolio optimization using classification and genetic algorithms. *Applied Mathematical Sciences*, 6(94):4673–4684.

Holland, J. H. (1975). Adaptation in natural and artificial systems. *Ann Arbor, MI: University of Michigan Press.*

Lin, C.-M. and Gen, M. (2007). An effective decision-based genetic algorithm approach to multiobjective portfolio optimization problem. *Applied Mathematical Sciences*, 1(5):201–210.

Lin, D. and Wang, S. (2002). A genetic algorithm for portfolio selection problems. *AMO - Advanced Modeling and Optimization*, 4(1):13–27.

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, 7(1):77–91.

Misra, A. K. and Sebastian, V. J. (2013). Portfolio optimization of commercial banks-an application of genetic algorithm. *European Journal of Business and Management*, 5:2222–2839.

Pandari, A. R., Azar, A., and Shavazi, A. R. (2012). Genetic algorithms for portfolio selection problems with non-linear objectives. *African Journal of Business Management*, 6(20):6209–6216.

Pereira, R. (2000). *Genetic Algorithms optimization for Finanace and Investments.*

Popov, A. (2005). *Genetic Algorithms for optimization-programs for Matlab.*

Roudier, F. (2007). Portfolio optimization and genetic algorithms. Master's thesis, Swiss Federal Institute of Technology.

Samanta, B. and Roy, T. K. (2005). Multi-objective portfolio optimization model. *Tamsui Oxford Journal of Mathematical Sciences*, 21 (1):55–70.

Sefiane, S. and Benbouziane, M. (2012). Portfolio selection using genetic algorithm. *Journal of Applied Finance and Banking,*, 2(4):143–154.

Sinha, P., Chandwani, A., and Sinha, T. (2013). Algorithm of construction of optimum portfolio of stocks using genetic algorithm. *Journal of finance.*

Wang, S.-M., Chen, J.-C., Wee, H.-M., and Wang, K.-J. (2006). Non-linear stochastic optimization using genetic algorithm for portfolio selection. *International Journal of Operations Research*, 3(1):16–22.

Yao, X., Liu, Y., and Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3(2):82–102.

Zhang, W.-G., Chen, W., and Wang, Y.-L. (2006). The adaptive genetic algorithms for portfolio selection problem. *International Journal of Computer Science and Network Security*, 6(1):196–200.

Zhou, C., Yu, L., Huang, T., Wang, S., and Lai, K. K. (2006). Selecting valuable stock using genetic algorithm. *International Journal of Operations Research*, pages 688–694.

Appendix A

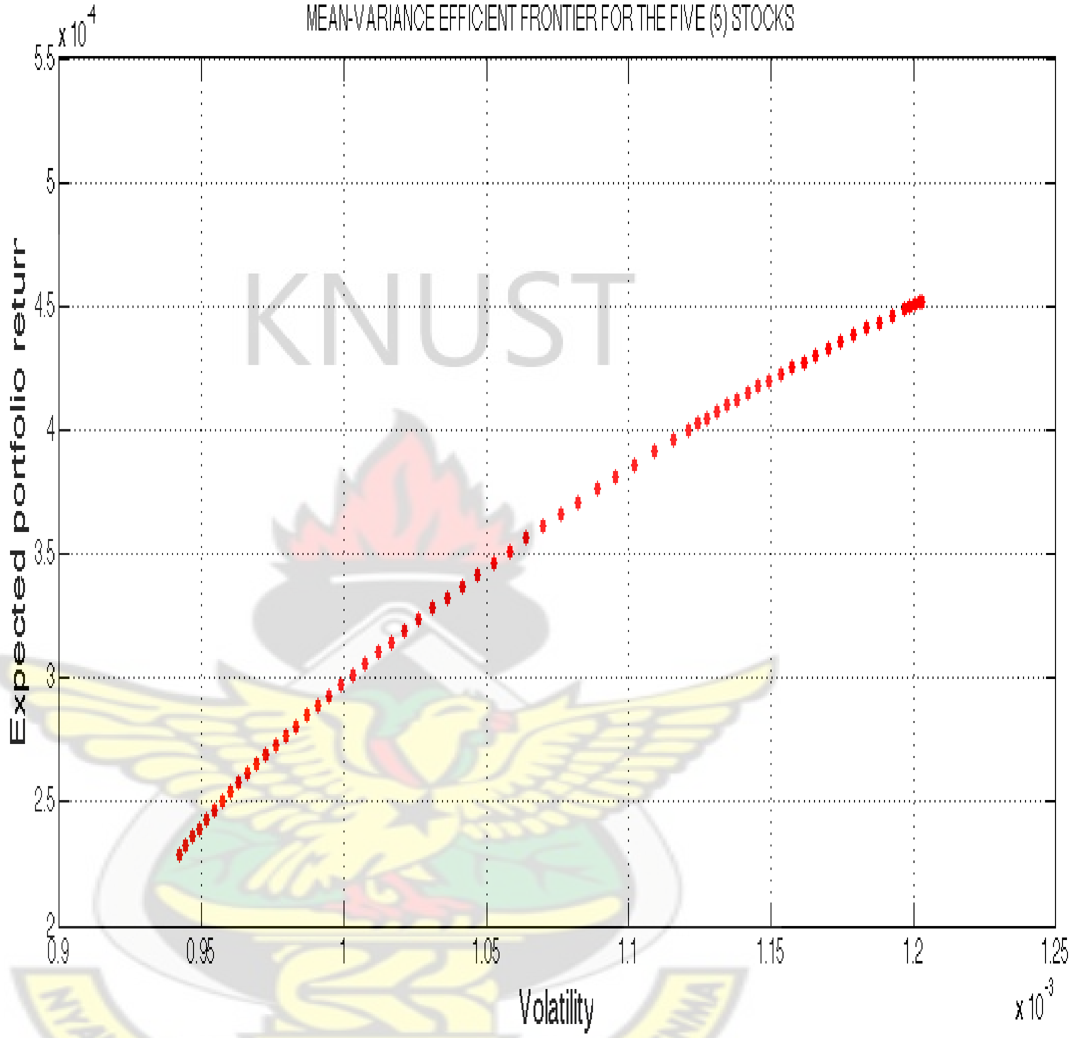

Figure 5.1: Efficient frontier as computed from the stocks

Appendix B

```
function[f] = portx(Wts)

global expRet expCov

A=0.5;

format long

Wts

Ret=expRet*(Wts')

Var=Wts*expCov*(Wts')

f =(A*(Wts*expCov *(Wts')))-((1-A)*(expRet*(Wts')))


function [x,fval,exitflag,output,population,score] = genetics(nvars...

Aeq,beq,lb,ub,PopulationSize_Data,EliteCount_Data,CrossoverFraction_Data)

% Constraint definitions used by |ga|

Aeq = ones(1,5); beq = 1;      % weights sum to 1

lb  = zeros(1,5);              % weights are positive

ub  = ones(1,5);               % weights are below one

PopulationSize=50; EliteCount=3; CrossoverFraction=0.9;


% Start with the default options

options = gaoptimset;

% Modify options setting

options = gaoptimset(options,'PopulationSize', PopulationSize_Data);

options = gaoptimset(options,'EliteCount', EliteCount_Data);

options = gaoptimset(options,'CrossoverFraction', CrossoverFraction_Data);

options = gaoptimset(options,'SelectionFcn', @selectionroulette);

options = gaoptimset(options,'CrossoverFcn', {  @crossoverheuristic [] });

options = gaoptimset(options,'MutationFcn', @mutationadaptfeasible);

options = gaoptimset(options,'Display', 'off');
```

```
options = gaoptimset(options,'PlotFcns', {  @gaplotbestf @gaplotbestindiv ...
@gaplotdistance });
[x,fval,exitflag,output,population,score] = ...
ga(@portx,nvars,[],[],Aeq,beq,lb,ub,[],[],options);

function [x,fval,exitflag,output,population,score] = genetics(nvars...
Aeq,beq,lb,ub,PopulationSize_Data,EliteCount_Data,CrossoverFraction_Data)
% Start with the default options
options = gaoptimset;
% Modify options setting
options = gaoptimset(options,'PopulationSize', PopulationSize_Data);
options = gaoptimset(options,'EliteCount', EliteCount_Data);
options = gaoptimset(options,'CrossoverFraction', CrossoverFraction_Data);
options = gaoptimset(options,'SelectionFcn', @selectionroulette);
options = gaoptimset(options,'CrossoverFcn', {  @crossoverarithmetic [] });
options = gaoptimset(options,'MutationFcn', @mutationadaptfeasible);
options = gaoptimset(options,'Display', 'off');
options = gaoptimset(options,'PlotFcns', {  @gaplotbestf @gaplotbestindiv ...
@gaplotdistance });
[x,fval,exitflag,output,population,score] = ...
ga(@portx,nvars,[],[],Aeq,beq,lb,ub,[],[],options);

function [x,fval,exitflag,output,population,score] = genetics(nvars...
Aeq,beq,lb,ub,PopulationSize_Data,EliteCount_Data,CrossoverFraction_Data)
PopulationSize=50; EliteCount=3; CrossoverFraction=0.9;
% Start with the default options
options = gaoptimset;
% Modify options setting
options = gaoptimset(options,'PopulationSize', PopulationSize_Data);
```

```
options = gaoptimset(options,'EliteCount', EliteCount_Data);
options = gaoptimset(options,'CrossoverFraction', CrossoverFraction_Data);
options = gaoptimset(options,'SelectionFcn', @selectionroulette);
options = gaoptimset(options,'CrossoverFcn', {  @crossoverscattered [] });
options = gaoptimset(options,'MutationFcn', @mutationadaptfeasible);
options = gaoptimset(options,'Display', 'off');
options = gaoptimset(options,'PlotFcns', {  @gaplotbestf @gaplotbestindiv ...
@gaplotdistance });
[x,fval,exitflag,output,population,score] = ...
ga(@portx,nvars,[],[],Aeq,beq,lb,ub,[],[],options);
```