

Optimum Portfolio Selection Using a Hybrid Genetic Algorithm and Analytic Hierarchy Process: An Application to Amman Stock Exchange

Al Hallaq SS^{1,*}, Ajlouni MM² and Al Saadi MF³

¹Department of Financial Economics, Arab Open University, Kuwait

²Department of Banking and Finance, Arab Open University, Kuwait

*Corresponding author: Al Hallaq SS, Department of Financial Economics, Arab Open University, Kuwait; E-mail: [hallaqsaad\(at\)yu.edu\(dot\)jo](mailto:hallaqsaad(at)yu.edu(dot)jo)

Received date: 19 November 2020; Accepted date: 27 November 2020; Published date: 31 November 2020

Citation: Al Hallaq SS, Ajlouni MM, Al Saadi MF. Optimum Portfolio Selection Using a Hybrid Genetic Algorithm and Analytic Hierarchy Process: An Application to Amman Stock Exchange. SunText Rev Arts Social Sci 1(2): 108.

DOI: <https://doi.org/10.51737/2766-4600.2020.008>

Copyright: © 2020 Al Hallaq SS, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract

The aim of this study is to investigate the ability of a hybrid genetic algorithm (HGA) and analytic hierarchy process (AHP) in selecting the optimum portfolio. This of course, helps investors to decide the most appropriate investment alternatives. For that purpose, the study creates portfolios using daily returns of the companies listed in Amman Stock Exchange, for the period from January 1, 2015 to December 31, 2015. The results show that HGA can identify portfolios that are in the efficient frontier. HGA has advantages over disadvantages for the portfolio selection cases in which the scale of the problem or the nonlinear constraints cannot be solved by linear or quadratic models. In addition, the results reveal that AHP can select the optimum portfolio among the portfolios obtained by HGA.

Keywords: Hybrid genetic algorithm; Analytic hierarchy process; Portfolio selection; Optimum portfolio; Amman stock exchange

Introduction

Selecting investment portfolio is one of the most important research areas in modern finance science; it is seeking to better allocation of funds between baskets of securities. Portfolio selection was introduced first by "Harry Markowitz" in 1952, in his paper "portfolio selection." He explained the concept of diversification, and suggested that investors focus on portfolio selection depending on their overall risk-reward characteristics [1]. Markowitz explained the best way to allocate the money over risky asset in a fixed position. The model of Markowitz consists of only two factors, which are the expected return and variance, and presume investors are risk averse. The idea of the model is that an investor cannot achieve a high return without increasing portfolio risk [2].

Since Markowitz until today, many attempts been conducted in the portfolio management to find new mathematical approaches to select the optimum portfolio. The computational capacity of the 21st century and the wide availability of computers have allowed the development of a new generation of intelligent computer

techniques such as hybrid genetic algorithm (HGA). HGA is a problem-solving technique motivated by biological evolution, based on an artificially simulated natural selection process or the survival of the fittest, known as Darwinian evolution. HGA can only identify the portfolios being in the efficient frontier and has been used in many recent works to find the optimum portfolios, due to multiple objective functions in portfolio selection. Therefore, the Analytic Hierarchy Process (AHP) approach is tied to the HGA to select the optimal portfolio based on all criteria between the portfolios obtained by HGA. The main objective of this study is to investigate the ability of HGA and AHP in selecting the optimum portfolio using data of the Jordanian companies listed in Amman Stock Exchange (ASE) [3-5].

Problem of the study

The huge amount of data and the complex function in discrete sampling space, for example, nonlinear function and constraints methods like a gradient method, simplex method, and dual simplex, the portfolio selection and optimization problem proves

to be not efficiently solved. For the above-mentioned reasons, the current study switched to HGA and AHP for finding the optimum portfolio. Thus, the study problem can be expressed by the following research questions:

- Can HGA and AHP be used in selection of optimum portfolio?
- Can Efficient Frontier (EF) be identified using HGA?
- Can AHP select the optimum portfolio among the portfolios obtained by HGA?

Objectives of the study

The main objective of this study is to investigate the ability of hybrid genetic algorithm and analytic hierarchy process in selecting the optimum portfolio. In particular, the study aims to identify the efficient frontier using HGA and selecting the optimum portfolio according to AHP from HGA's EF.

Importance of the study

A review of literature on the optimum portfolio selection using a hybrid genetic algorithm and analytic hierarchy process indicates that most of the studies on this topic have been conducted in developed countries. This is probably attributed to the fact that the conduct of such studies requires a relevant level of disclosure data and analysing techniques [6-10]. Thus, this study will contribute to the literature by producing answers regarding the selection of optimum portfolio using a hybrid genetic algorithm and analytic hierarchy process. HGA and AHP are new techniques, which are added to portfolio selection to reach EF and optimum portfolio. Therefore, the importance of this methodology is derived from the importance of optimum portfolio, which all investors wish to reach [11].

Structure of the study

The paper is organized as follows: In addition to introduction, section two presents the theoretical background of the study. Section three discusses literature review of the study, and what distinguish the current study. Section four describes the procedures used to gather data for the study, the methodology used. Section five presents the results of analysis HGA and AHP. Final section presents recommendations and policy implications of the study in question [12-16].

Theoretical Background

Theories developed to explain the portfolio selection and genetic algorithm will be discussed in the upcoming lines. The beginning of the modern investment theory traced back to 1952, when Harry Markowitz (1952) published an article titled "Portfolio Selection" [17]. He showed how to create a frontier of investment portfolios, so that everyone had the highest expected rate of return given its

level of risk or the minimum level of risk given its rate of return. The calculation technique was very complex, especially given the technology of the time. In the past, the optimization of Markowitz's portfolio was used, mostly in the asset allocation decision. The investor decides on the amounts to invest in certain basic classes such as stock, bonds and real estate assets. The computing power needed to optimize more than a few asset classes is only a small fraction of what is needed to optimize more than thousands of stocks. There is a need to utilize the available quantitative data to solve the optimization problem. Prior to the spread of portfolio theory in the real world, three scholars simultaneously and independently asked the following questions: Assume that everyone successfully makes his investments using portfolio theory and invested in portfolios on the frontier, how would this affect the price of securities? In response to this question, Sharpe (1964), Lintner (1965) and Mossion (1966) [1-3] developed the Capital Asset Pricing Model (CAPM), which is widely used in the real world to measure portfolio performance; securities value, make capital budgeting decisions, and even regulate utilities. However, the model was challenged by Richard Roll [4,5], who argued that the model should be discarded because it was impossible to verify empirically its single economic forecast. This controversial issue remains the subject of a lively debate today. At the same time, Steve Ross [6] developed an alternative model to CAPM. This model was called the Arbitrage Pricing Theory (APT) where expected return should be linked to risk so that no single investor could create unlimited return through arbitrage. The question of how to price option contracts has puzzled researchers in finance until a paper of Fisher Black and Myron Scholes was published in 1973 [7]. They argued that you could make a riskless hedged position with an option by taking a position in both the option and the stock it is written on. Although researchers in the finance science were trying to determine the nature of the price structure in the securities markets, the issue of how efficient the market is in pricing to its structure was called into question. Fama (1970) stated that a market is efficient if security prices immediately and fully reflect all available relevant information [8]. Fama, [9] divided the overall EMH and the empirical test of the hypothesis into three sub-hypotheses depending on the information set including: Weak-Form of Efficiency, Semi Strong-Form of Efficiency, and Strong-Form of Efficiency [10].

The Markowitz theory is now known as the Modern Portfolio Theory (MPT). It is an investment theory that aims to maximize the expected return of the portfolio for a given level of portfolio risk, or in an equivalent way, to minimize the risk for a given expected return, by carefully choosing the proportions of different asset levels [18]. Although MPT is widely used in practice in the financial sector in recent years, the basic assumptions of the same have been largely questioned. MPT, as an improvement over

traditional investment models, is an important mathematical modelling for the advancement of finance [19].

Optimal Portfolio Selection

Portfolio theory assumes that investors are essentially risk averse, meaning that given the choice between two assets with equal rates of return, they would select the asset with the lowest risk [11]. This combination of risk preference and risk aversion can be explained by an attitude to risk that depends on the amount of money involved. Although diversity of attitudes is recognized, the basic assumption is that most investors who use large sums of money to develop an investment portfolio are averse to risk. As a result, there is a positive relationship between expected return and expected risk in the optimal selection process [12]. The optimal portfolio is a combination of investment, each of which has desirable individual risk-return characteristics that are also adjusted according to their correlations [13]. The optimal portfolio is the efficient frontier portfolio that has the greatest value for a given investor. It is at the point of tangency between the curve of the efficient frontier and the curve with the highest potential utility [20].

To expand Markowitz's model portfolio and the assumptions of EMH the risk-free rate of return should be considered. Correlation and covariance of any asset with a risk-free asset is zero. So any combination of an asset or portfolio in the risk-free asset generates a linear return and risk function. As a result, the combination of the risk-free asset with a risky asset in the Markowitz efficient frontier gives rise to linear portfolio opportunities while the dominant line is that which tangents to the efficient frontier. This dominant line is known as the Capital Market Line (CML) [10]. Because all investors want to invest in the portfolio that is risky at the point of tangency, this portfolio - known market portfolio must contain all risky assets pro rata to their relative market values. In addition, the investment decision and financing decision can be separated because, although everyone wants to invest in the market portfolio, investors make different financing decisions on loans or borrow depending on their preferences of individual risk [14]. Given the CML and the predominance of the market portfolio, the relevant risk measure for an active individual risk is its covariance with the market portfolio. That is, it is systematic risk, when the covariance for the market portfolio is standardized, a known beta measure of the systematic risk and market line, which relates to the expected or required rate of return of an asset with its beta version [15]. Individual securities and portfolios are represented in the Security Market Line (SML) to determine the expected return because of a systematic risk (beta). Alternatively, assuming that markets are not always fully effective, one can identify undervalued and overvalued securities by comparing the estimated rate of return of an investment to its expected rate of return. The systematic

variable risk (beta) for an active individual risk is calculated using a regression model that generates an equation referred to as the assets characteristic line [16].

Data and Methodology

The main objective of this section is to investigate the ability of a hybrid genetic algorithm (HGA) and analytic hierarchy process (AHP) in selecting the optimum portfolio, in ASE. This will be done, focusing on the methodology, data collection and sources, the study sample selection rules and empirical background of the study variables. Along with their operational definitions, measurements and hypotheses of the study. Finally, it presents the empirical models to be used to test the study hypotheses [21].

Population and sample

The study population consists of all companies listed in Amman Stock Exchange (ASE) during the period of January 1, 2015 to December 31, 2015. At the end of 2015, the total number of companies listed in was 228 companies. The current study sample been selected according to the criteria of continuous data availability. Therefore, the sample is limited to companies that met the following criteria:

1. Companies should have complete data availability during the study period.
2. Companies should have been established before 15 years.
3. Companies should not have been engaged in acquisitions or merger during the study period.

By applying these criteria, the final sample of this study is restricted to 60 companies.

Sources of data

The study depends on secondary data that have been published in annual reports issued by Jordanian companies listed on Amman Stock Exchange (ASE), reports and trading data issued by ASE, statistical databases issued by the Central Bank of Jordan (CBJ), books and references, studies, previous researches that are related to the subject matter, and network (internet) publications.

Variables of the study

To achieve the purposes of the study, the following variables will be used: return, risk, beta, liquidity ratio, Sharpe ratio, Treynor's ratio and Jensen's alpha.

Mathematical Model

The current study aims to test the ability of HGA and AHP in selecting the optimum portfolio of shares listed in Amman Stock Exchange. The model used in this study is based on the Solimanpur et al. [17] models.

Notations

The following notations are used to formulate portfolio selection problem:

- I: Index for stocks.
- N: total number of stocks.
- r_i : Return of stock i .
- R_p : Return of portfolio.
- σ_p : Risk of portfolio
- x_j : Percentage of stock j in portfolio.

Objective functions

The attempted mathematical model includes two objective functions. Namely, return and risk of portfolio. These objective functions are formulated in the following:

1. Return of Portfolio: The portfolio return is defined as the weighted average returns of the portfolio shares and is expressed as follows [10]:

$$R_p = \sum_{i=1}^N r_i x_i \quad (1)$$

2. Risk of portfolio: Risk of a stock is defined as the tolerance of stock's return from the mean. The risk of a portfolio is expressed as follows [10]:

$$\sigma_p = \sqrt{\sum_{i=1}^N \sum_{j=1}^N r_i x_j \text{cov}_{ij}} \quad (2)$$

Where:

cov_{ij} : The covariance between returns of stock i and j expressed as $\text{cov}_{ij} = r_{ij} \sigma_i \sigma_j$

r_{ij} : is the correlation coefficient between i and j .

3. Constraints:

Constraints of the attempted model are as follows:

$$\sum_{i=1}^N x_i = 1; \text{ if } x_i \geq 0, i = 1, 2, \dots, N$$

It should be noted that the non-negativity of the decision variables is used to prevent short selling.

4. Proposed Hybrid Genetic Algorithm (HGA):

Portfolio display

In the proposed HGA, any portfolio is represented by $N \times \text{num_bits}$ genes in which N is the number of companies and num_bits is the number of binary bits used for representing the share of each company in the portfolio. The share of company i in the portfolio can be calculated by Solimanpur [17]:

$$x_i = \frac{v_i}{\sum_{i=1}^N v_i} \quad (3)$$

Where:

v_i : The decimal value of the binary code dedicated for company i .

This formula ensures that, after the execution of genetic operations, the constraints of mathematical models will be satisfied.

Fitness function

Fitness of any chromosome in the proposed HGA is calculated in K directions. Fitness of chromosome S in direction k is computed by Solimanpur [17]:

$$\text{fit}_k(S) = w_{k1} R'_p(S) + w_{k2} \sigma'_p(S) \quad (4)$$

Where:

$\text{fit}_k(S)$: is the fitness value of chromosome S in direction k .

$R'_p(S)$: is the normalized value of return of chromosome S .

$\sigma'_p(S)$: is the normalized value of risk of chromosome S .

w_{k1} : is the weight of return in direction k .

w_{k2} : is the weight of risk in direction k .

The normalized values of return and risk of portfolio S in the population are defined as follows Solimanpur [17]: Ω

$$R'_p(S) = \frac{R_S}{\max_{U \in \Omega}(R_U)} \quad (5)$$

$$\sigma'_p(S) = \frac{\min_{U \in \Omega}(R_U)}{\sigma_S} \quad (6)$$

Selection

In the proposed HGA, selection probability of chromosome S in direction k it is proportional to the quality of this chromosome in the direction k , i.e $\text{fit}_k(S)$. In other words, the higher the fitness of a chromosome, the higher should be selection probability [22].

Portfolio Selection via Analytic Hierarchy Process (AHP)

The method proposed an evaluation procedure comprising the following steps [18]:

Step 1: Identify all the criteria to be taken into account in the choice of the portfolio (evaluation) and to build a hierarchy of decision-making.

Step 2: Calculate weights of criteria using AHP method.

These steps are performed in the following subsections.

Evaluation criteria

Selecting the appropriate portfolio of performance measures to provide the necessary information to investors to assess the effectiveness with which they can invest their money is a vital issue. The performance appraisal is primarily related to the determination of how a particular investment portfolio has performed compared to a benchmark comparison. To effectively manage portfolio selection, it is necessary to consider the critical factors that reflect investor behaviour and the state of the financial market. The different stakeholders within a decision process can be relatively diverse, with different objectives and conflicts of value systems [23-29]. In this respect, a key concept is the relationship between risk and return. Each performance index provides a different perspective on the balance between the level of return and the exposure to risk. Therefore, in this study, seven

measures, return, risk, beta, liquidity ratio, Sharpe ratio, Treynor's Ratio (TR) and Jensen's alpha (Alpha ratio) have been identified as criteria, which affect decision of investors in portfolio selection. Other criteria are discussed in the following subsections.

Beta

Beta measures the price volatility of a share compared to the rest of the market. This measure of risk is defined as [10]:

$$\beta_i = \frac{Cov_{r_i, r_M}}{Var(r_M)} \quad (7)$$

Where:

Cov_{r_i, r_M} : The covariance between the rates of return of shares i and the return of the market or index.

R_M : The rate of return of the market portfolio.

Liquidity ratio

This ratio indicates the annual liquidity of a stock. Haugen can then calculate liquidity of a portfolio [10]:

$$L_p = \sum_{i=1}^N L_i X_i \quad (8)$$

Where:

L_i : The liquidity of share i .

X_i : The percentage of share i in the portfolio.

Sharpe ratio

Sharpe ratio or RVAR measures the return into the portfolio risk (standard deviation of return) [20]. It can be defined as this ratio:

$$RVAR = \frac{\overline{TR_p} - \overline{RF}}{SD_p} \quad (9)$$

Where:

$\overline{TR_p}$: The mean of portfolio return in a certain period.

\overline{RF} : The mean of non-risk return rate in time set.

SD_p : The standard deviation of portfolio return in time set.

$\overline{TR_p} - \overline{RF}$: The surplus return of portfolio.

This ratio measures the surplus return of portfolio versus a risk unit. The higher the RVAR of a portfolio, the higher will be its desirability.

Treynor's ratio (TR)

TR measures the proportion of extra return on beta [20]. Treynor defines this ratio:

$$TR = \frac{\overline{TR_p} - \overline{RF}}{\beta_p} \quad (10)$$

Where:

β_p : The systematic risk of portfolio (beta ratio).

This ratio measures the excess return of portfolio versus portfolio's systematic risk unit.

Alpha ratio

Alpha ratio is linked to Treynor's ratio and, hence, it provides a classification that is close to the performance of a portfolio [21]. This ratio is defined as:

$$\overline{\alpha_p} = \overline{R_p} - [\overline{RF} + (RM - \overline{RF})\beta_p] \quad (11)$$

The weakness and strength of a portfolio's performance has two sources: The first is the portfolio manager's ability to select appropriate shares, and the second is their ability to make appropriate decisions over time and to assess threats and market opportunities. Obviously, the manager who considers these aspects will have a better performance in managing the portfolio. The benefit of using this ratio is the possibility to measure $\overline{\alpha_p}$ and β_p at the same time [10].

The analytic hierarchy process methodology

AHP includes six basic steps, as summarized in the following [18,22]:

Step 1: AHP breaks down a complex multi-criteria decision-making problem to several small sub-problems each with a single criterion. Thus, the first step is to break the problem of decision in a hierarchy with a goal at the top, criteria and sub-criteria at levels and sub-levels and alternative decisions at the bottom of the hierarchy.

Step 2: The decision matrix, which is based on Saaty's nine-point scale, is constructed. The decision-maker uses the fundamental scale defined by Saaty to assess the priority score.

Step3: It involves pairwise comparison of the elements of the constructed hierarchy. The aim is to set their relative priorities with respect to each of the elements at the next higher level.

Step 4: AHP also calculates an inconsistency index to reflect the consistency of decision maker's judgments during the evaluation phase. The inconsistency index in both the decision matrix and in pairwise comparison matrices can be calculated by the following equation [22]:

$$CI = \frac{\lambda_{max} - N}{N - 1} \quad (12)$$

The closer the inconsistency index to zero; the greater will be the consistency of decision-maker's judgments. The consistency of the assessments is ensured if the equality $a_{ji} \cdot a_{jk} = a_{ik}$ holds for all criteria. The relevant index should be lower than 0.10 to accept the AHP results as consistent. If this is not the case, the decision-maker should go back to Steps 2 and 3 and redo the assessments and comparisons.

Step 5: Before any calculation, the comparison matrix has to be normalized. To normalize, each column is divided by the sum of entries of the corresponding column. In that way, a normalized matrix is obtained in which the sum of the elements of each column is 1.

Step 6: The relative values obtained in the third step should satisfy:

$$AW = \lambda_{max} W, \quad (13)$$

Where:

A: Represents the pairwise comparison matrix.

λ_{max} : The highest eigenvalue.

W: Weight of the vector or elements.

If there are elements at the higher levels of the hierarchy, the obtained weight vector is multiplied by the weights of the elements at the higher levels, until the top of the hierarchy is reached. The alternative with the highest weight is finally considered as the best alternative.

Questions of the Study

The study questions are:

- Can HGA and AHP be used in selection of optimum portfolio?
- Can Efficient Frontier (EF) be identified using HGA?
- Can AHP select the optimum portfolio among the portfolios obtained by HGA?

The study provides the population, sample, sources of data and variables of the study. Mathematical model and the methodology of the HGA and AHP are also reviewed and in the last hypotheses of the study. The following section presents data analysis, process and implementation of the HGA, AHP and their results.

Data Analysis

This section shows the descriptive statistics of the study variables, and then examines the ability of the HGA and AHP to select the optimum portfolio. In particular this section aims to answer the questions and test the hypotheses of this study.

Hybrid genetic algorithm process and implementation

This section is going to discuss the implementation of the methodology of the HGA. To implement the previously discussed methodology, we created an application using Visual Studio 2010, and C#.NET as a programming language [30]. Visual studio 2010 is an integrated development environment used to create a variety of computer programs, you can select from many available programming languages such as C++,C# ,Visual Basic...etc. This study used C# language, C# is a general-purpose programming language used to create windows forms applications for any purpose. As Figure 1 show, the first step in this work is to read the stocks data. In this step, the program reads the stocks data and creates an object for each stock, and then loads the daily data return for each stock. After that it uses the stocks, data to create 10,000 portfolios, where by each portfolio consists of six stocks. After that it created the portfolios, it uses the daily data return of the assets to calculate the portfolios risk and return. Then it use the portfolios risk and return to evaluate the fitness of the portfolios and select the top 10 portfolios to be used in the HGA. After that, it uses the top 10 portfolios in the

crossover process to generate the second generation of portfolios. Finally, it selects the top 10 portfolios from the second-generation portfolios and draws the efficient frontier.

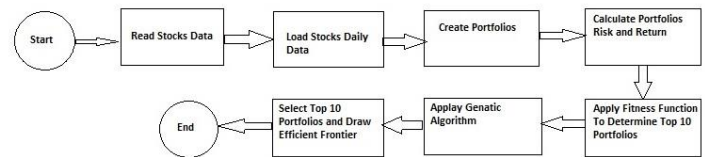


Figure 1: Flowchart of the Implementation Process.

Implementation details

The study gathered the stocks data from Amman Stock Exchange website for the year 2015 and used the data to create excel sheets and notepad files.

As Figure 2 shows, using the file menu and clicking the “Select Data” option, the program will open a select file dialog which allows us to select the stocks data from a notepad (.txt) file.



Figure 2: Creating a File and Selecting Data Using C#.

Then the program will read the file data and loads it to the system and create an object for each stock which consists of five characteristics, which are:

1. Stock ID: unique integer value varies between 60-120 which is used to identify each stock.
2. Stock Code: Company code as in Amman Stock Exchange.
3. Return: Stock expected return.
4. Risk: Stock risk.
5. Daily Data: The stock daily return.

We have assigned a unique ID for each stock which varies from 60-120. The ID will be used to calculate the weight of each stock in the portfolio, in a manner that the stock with the highest expected return has the highest ID and so on. It would be natural to choose the ID's from 1-60 but we chose to select the ID's from 60-120 for the following reason:

In the 1-60 ID's the share of each stock will dramatically vary between the following values:

- Highest weight would be the result of company with ID 60 to be assigned in a portfolio with the stocks of ID's 1,2,3,4 and 5. In this situation the Stock with ID 60 will have a weight of $60/(60+1+2+3+4+5) = 80\%$.

- Lowest weight would be the result of company with ID 1 to be assigned in a portfolio with the stocks ID's 60, 59, 58, 57 and 56. In this situation the Stock with the ID 1 will have a weight of $1/(60+59+58+57+56+1)=0.34\%$

However, in the situation of ID's 60-120 we had the weights vary between 9.2% and 27.9%. As we can see, the second approach gives a better chance for the assets to affect the portfolio return and risk, and eliminates the possibility of a single asset to dominate the portfolio. After we have loaded the stocks data, we can load the stocks daily data by clicking the "Load daily data" option as shown. By clicking, the "Load daily data" option the program will show a select file dialog box, which allows us to select notepad file (.txt) which contains each stock and its daily data. Thus, the program will load the daily data for each stock into the stocks objects, which is going to be used in the process of portfolios risk and return calculation [31-36].

Having the stocks are created and the daily data are loaded, we can create the portfolios, by clicking the "Create Portfolios" option shown in Figure 2. The program will use the stocks data to create 10,000 portfolios where each portfolio consists of six different stocks to be selected randomly from the previously created 60 stocks. When the portfolios creation is done, we can calculate the risk and return for each portfolio by clicking the "Portfolio Risk & Return" option from the Calculate menu, as in Figure 3.

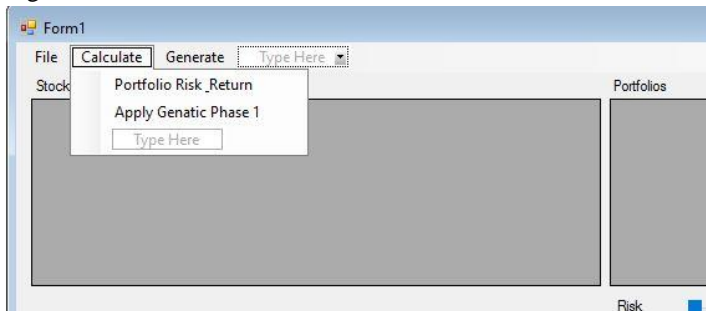


Figure 3: Calculate risk and return.

By clicking the "Portfolio Risk and Return" option the program will calculate the risk and return for each portfolio as follows. Portfolio return will be calculated according to the following formula [10]:

$$\text{Return} = w_1 * R_1 + w_2 * R_2 + w_3 * R_3 + w_4 * R_4 + w_5 * R_5 + w_6 * R_6 \quad (14)$$

Where:

W: The weight of each stock in the portfolio.

R: The rate of return of each stock in the portfolio.

The weight for each stock in the portfolio is calculated by dividing the asset ID over the summation of the total assets IDs of the portfolio.

The 6 asset portfolio risk will be calculated according to the following formula [10]:

$$\begin{aligned} \sigma^2 \text{por} = & w_1^2 \sigma_1^2 + w_2^2 \sigma_2^2 + w_3^2 \sigma_3^2 + w_4^2 \sigma_4^2 + w_5^2 \sigma_5^2 + w_6^2 \sigma_6^2 + \\ & 2w_1w_2 \text{cov}(1,2) + 2w_1w_3 \text{cov}(1,3) + 2w_1w_4 \text{cov}(1,4) + \\ & 2w_1w_5 \text{cov}(1,5) + 2w_1w_6 \text{cov}(1,6) + 2w_2w_3 \text{cov}(2,3) + \\ & 2w_2w_4 \text{cov}(2,4) + 2w_2w_5 \text{cov}(2,5) + 2w_2w_6 \text{cov}(2,6) + \\ & 2w_3w_4 \text{cov}(3,4) + 2w_3w_5 \text{cov}(3,5) + 2w_3w_6 \text{cov}(3,6) + \\ & 2w_4w_5 \text{cov}(4,5) + 2w_4w_6 \text{cov}(4,6) + 2w_5w_6 \text{cov}(5,6) \end{aligned} \quad (16)$$

And $\sigma \text{por} = \sqrt{\sigma^2 \text{por}}$

To calculate the covariance we used the stocks daily data to calculate the mean and variance for each stock and followed the following formula to calculate the covariance between each two stocks in any portfolio:

$$\text{Cov} = \frac{\sum (x - \bar{x})(y - \bar{y})}{n} \quad (17)$$

Where:

\bar{x} and \bar{y} : are the sample means return averages.

n: is the sample size.

After we calculate the risk and return for each portfolio, we apply the following fitness function to calculate the fitness factor for each portfolio [17]:

$$\text{fit}_k(S) = w_{k1} R'_p(S) + w_{k2} \sigma'_p(S) \quad (18)$$

Where:

$\text{fit}_k(S)$: fitness value of chromosome S in direction k.

$R'_p(S)$: normalized value of return of chromosome S.

$\sigma'_p(S)$: normalized value of risk of chromosome S.

w_{k1} : weight of return in direction k.

w_{k2} : weight of risk in direction k.

To produce our results we used the following weights for risk and return as shown in (Table 1). After we calculate the fitness value for each portfolio, we select the top 10 portfolios based on the highest fitness value.

Table 1: Fitness function weights.

Weight of risk	Weight of return
50%	50%
40%	60%
80%	20%
20%	80%
60%	40%

Now after we have determined the top 10 portfolios it is the time to use HGA. In the HGA, each portfolio is treated as a gene, where each gene consists of six chromosomes which represent the assets of the portfolio. A chromosome for a portfolio consists of the following assets {114, 120, 86, 74, 62, 91} would be [1110010 1111000 1010110 1001010 0111110 1011011] where each seven digits represent the chromosome for each asset which

are the result of the ID conversion from decimal to binary. The purpose of the HGA is to swap chromosomes between the best genes to generate better chromosomes. To do so, we separated the assets of the top 10 portfolios and used these assets to create the second generation of the portfolios using HGA [37]. After we separated the genes from which the chromosomes are created, we used these genes as the basis to create the new portfolios, which is known as the Crossover operation. We have created portfolios six times the number of the successful genes. The created portfolios are referred to as the second-generation portfolios [38-41]. After we created the second-generation portfolios, we selected the top 10 portfolios from the second-generation portfolios and created the efficient frontier graph, as Figure 3.

Coding Section of Preparing the HGA

So far, we have discussed the workflow of the program in general. In this section, we are going to discuss the work in more technical details and we are going to describe the code we used to create the application. Figure 4 shows the interface of the program we created. As we can see, the interface contains three menus, five grid views and a graph space to draw the efficient frontier and a navigation bar which is used to select the weight of the risk and return that used in the fitness function.

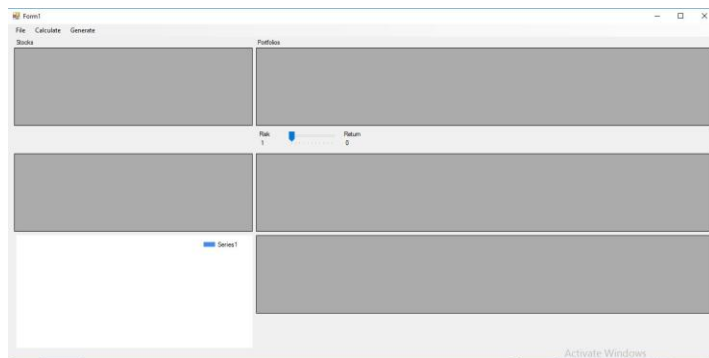


Figure 4: Program interface.

The left grid views are used to list the stocks used in creation of the portfolios. Where the right grid views are used to list the portfolios created in the different stages of the program run. The top left grid view is used to list all the available stocks which we created from the data we extracted from Amman Stock Exchange website. The bottom left grid view is used to list the stocks which are used to create the optimum portfolios to be used in the HGA to create the second generation portfolios [42]. The right top grid view is used to display the 10000 portfolios we created using the stocks listed in the top left grid view, where each portfolio consists of six different stocks. The right middle grid view is used to list the optimum portfolios which consist of the stocks listed in the bottom left grid view. The right bottom grid view is used to list the second generation

portfolios which are the result of the genetic algorithm and the crossover operation. As Figure 5 shows we have five different options which work as follows:

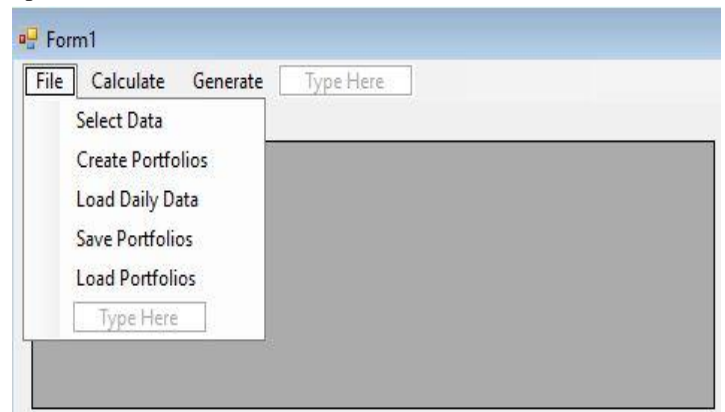


Figure 5: File Menu.

Figure 6 shows the code that runs when the “Select Data” option is clicked from the file menu. The first line of code defines a list of strings named lines to store the lines. The second line of code defines and creates an open file dialog box which is shown when the select data option is clicked. The dialog box allows us to select the file which contains the stocks data. The file should be in .txt format. The “IF statement” block is used to make sure a file is selected and then add the file lines into the list of lines which we defined in the first line.

```
private void selectDataToolStripMenuItem_Click(object sender, EventArgs e)
{
    List<string> lines = new List<string>();
    OpenFileDialog of = new OpenFileDialog();
    if (of.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
        StreamReader reader = new StreamReader(of.FileName);
        while (!reader.EndOfStream)
        {
            lines.Add(reader.ReadLine());
        }
        for (int i = 0; i < lines.Count; i++)
        {
            Stock temp = new Stock();
            string[] values = lines[i].Split('\t');
            temp.Code = Convert.ToInt32(values[0]);
            temp.Return = Convert.ToDouble(values[1])/100;
            temp.ID = Convert.ToInt32(values[2]);
            temp.Risk = Convert.ToDouble(values[3]);
            Stocks.Add(temp);
        }
        DataTable table = ConvertListToDataTable(Stocks);
        dataGridView1.DataSource = table;
    }
}
```

Figure 6: Select data code.

Figure 7 shows the stocks data file. Each line in the file represents a stock where the entries represent the stock code, expected return, ID and risk respectively. The entries are separated by tabs. The “For loop” block of code is used to create a stock for each line and assign the code, expected return, ID and risk for each stock and then add the stock to a list of stocks. The last two lines

of code are used to list the data on the application interface grid view.

Figure 8 shows the code which runs when the “Create portfolios” option is selected from the file menu. As we can see, the function contains a for-loop statement which is used to repeat the code within the block 10,000 times. In the first line of code we define a temporary portfolio, and then we define a temporary stock. After that, we keep repeating the code within the while-loop block until the number of stocks the portfolio contains is six. In the while loop, we create random number generator. The random number generator is defined as a random number that is bigger or equal to its lower limit, in our case 60, and less than its maximum limit, in our case 121, which gives us a random number from 60-120.

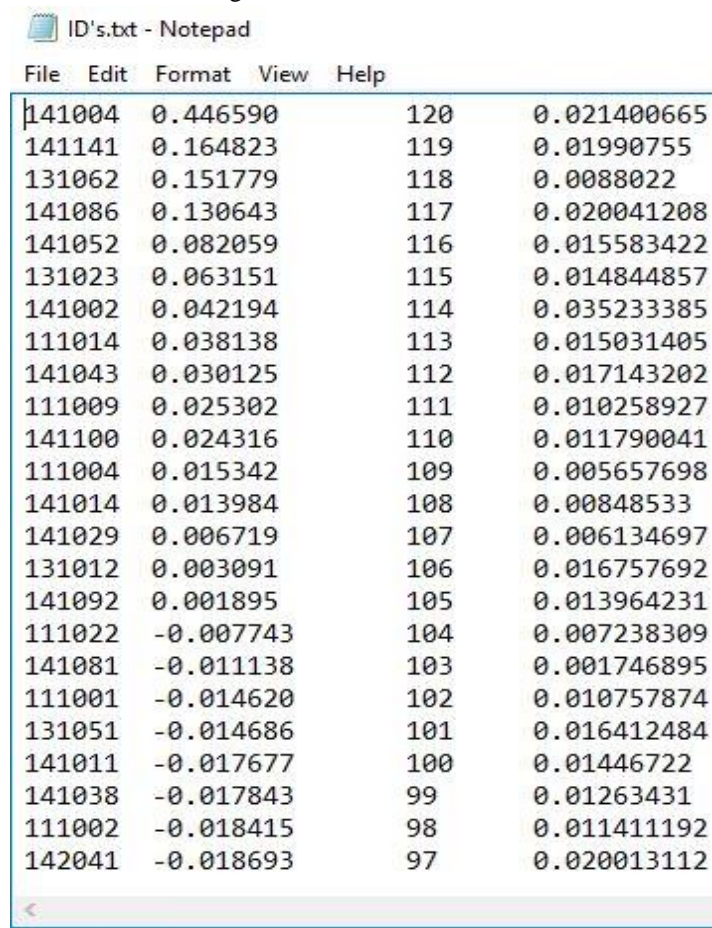


Figure 7: Stocks data file.

Then, we use the random number and search for the stock that has the same number as its ID and assign its value to the temporary stock that we defined earlier.

After that, we make sure that the portfolio does not contain this stock to make sure a portfolio does not have the same stock more than one time.

When the portfolio number of stocks is six the while-loop stops from adding more stocks to the portfolio. Then, we add the

portfolio to a list of portfolios and proceed to create another portfolio until the predefined number, in our case 10,000 is created and the loop stops [43-49].

```
private void createPortfoliosToolStripMenuItem_Click(object sender, EventArgs e)
{
    for (int i = 0; i < 10000; i++)
    {
        Portfolio temp_portfolio = new Portfolio();
        Stock temp_stock = new Stock();

        while(temp_portfolio.Stocks.Count<6)
        {
            Random rnd = new Random();
            temp_stock = Find_ID(rnd.Next(60, 121));
            if (!temp_portfolio.Stocks.Contains(temp_stock))
                temp_portfolio.Stocks.Add(temp_stock);
        }
        Portfolios.Add(temp_portfolio);
    }
}
```

Figure 8: Creating portfolios code.

Figure 9 shows the code which runs when “Load Daily data” option is clicked from the file menu. The first line is used to define a list of string values named lines. The second line is used to create an open file dialog box which is used to select a file that contains the daily data of the stocks.

```

List<string> lines = new List<string>();
OpenFileDialog of = new OpenFileDialog();
if (of.ShowDialog() == System.Windows.Forms.DialogResult.OK)
{
    StreamReader reader = new StreamReader(of.FileName);
    while (!reader.EndOfStream)
        lines.Add(reader.ReadLine());
}
for(int i=0;i<lines.Count;i++)
{
    string[] daily = lines[i].Split('\t');
    Stock s = Find_Code(Convert.ToInt32(daily[0]));
    for (int j = 1; j < daily.Length; j++)
        s.Daily.Add(Convert.ToDouble(daily[j]));
}

```

Figure 9: Load daily data code.

The If-statement block is used to make sure a file is selected and then it reads the content of the file and adds them into the array of lines.

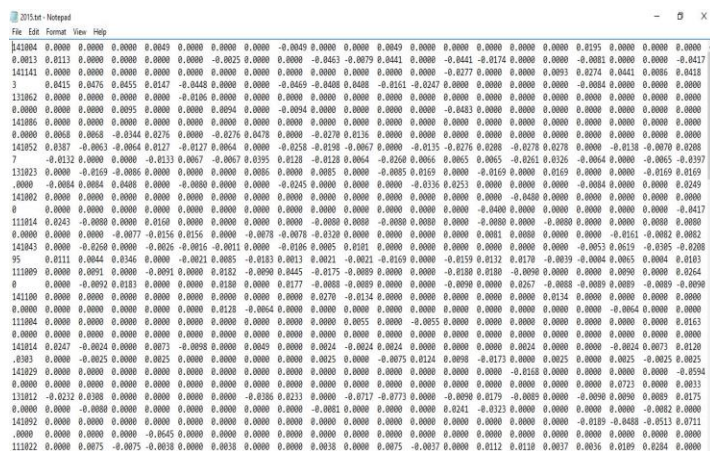


Figure 10: Daily data file.

The first line in the for-loop block splits each line based on the tab spaces between each value and assign them into an array named daily. The second line finds the stock that has the same code as found in the selected data file. Then, in the inner for-loop we add the daily data of the stock into the stock.

```
private void savePortfoliosToolStripMenuItem_Click(object sender, EventArgs e)
{
    string p = "";
    for (int i = 0; i < Portfolios.Count; i++)
    {
        for (int j = 0; j < 6; j++)
        {
            p += Portfolios[i].Stocks[j].ID + " ";
        }
        p += '\n';
    }
    StreamWriter sw = new StreamWriter("portfolios.txt");
    sw.Write(p);
    sw.Close();
    sw.Dispose();
}
```

Figure 11: Save Portfolios Code.

Figure 10 shows the daily data file. The file contains a line for each stock, where each line starts by the stock code, and then it contains the daily return of each stock. The values are separated by tabs.

Figure 11 shows the code which runs when “Save portfolios” option from the file menu is clicked. Since the process of creation of 10,000 seems to be a time consuming process, where we need to repeat the process of creation, random an ID number, search for the stock, check if the portfolio contains the stock or not, adding the stock and repeating this operation for 10,000 times, we figured that we can create the portfolios and save them, then we can load the portfolios later to assure the results consistency. The function saves the ID’s of the stocks which the portfolio contains as a string then writes the string to a file named “portfolios.txt” [50].

```
private void loadPortfoliosToolStripMenuItem_Click(object sender, EventArgs e)
{
    StreamReader sr = new StreamReader("portfolios.txt");
    string p = sr.ReadToEnd();
    string[] lines = p.Split('\n');
    for (int i = 0; i < lines.Length-1; i++)
    {
        Portfolio temp_port = new Portfolio();
        for (int k = 0; k < 6; k++)
        {
            Stock temp_stock = new Stock();
            string[] data = lines[i].Split(' ');
            temp_stock = Find_ID(Convert.ToInt32(data[k]));
            temp_port.Stocks.Add(temp_stock);
        }
        Portfolios.Add(temp_port);
    }
}
```

Figure 12: Load Portfolios Code.

Figure 12 shows the code which runs when the “Load portfolios” option from the file menu is clicked. As we can see the code, creates a file reader and read the contents of the “portfolios.txt” file and use the saved ID’s to create new portfolios using these ID’s.

Figure 13 shows the options of the calculate menu, the menu contains two options as shown. The “Portfolio Risk Return”

option is used to calculate the Risk and Return of the portfolios, while the “Apply Genetic Phase 1” is used to create the second generation portfolios.

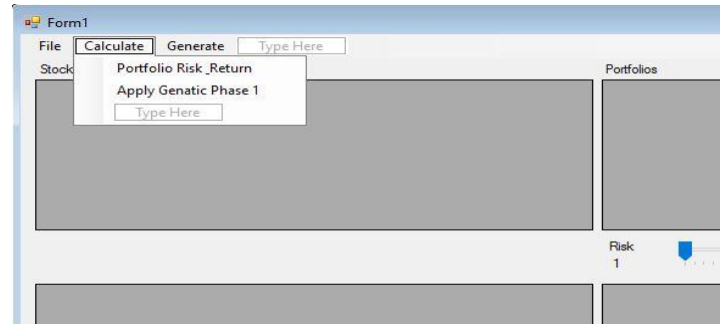


Figure 13: Calculate menu options.

Figure 14 shows the code which is used to calculate the risk and return for each portfolio, as we can see in the figure, the code calls the functions from the portfolio class to calculate the risk and return.

```
private void portfolioRiskReturnToolStripMenuItem_Click(object sender, EventArgs e)
{
    for (int i = 0; i < Portfolios.Count; i++)
    {
        Portfolios[i].Calculate_Return();
        Portfolios[i].Calculate_Risk();
    }
    DataTable table = ConvertPortToDataTable(Portfolios);
    dataGridView2.DataSource = table;
}
```

Figure 14: Calculate risk and return code.

Figure 15 shows the code used to calculate the portfolio return. At first we defined a dummy variable named sum then we calculated the sum of the stocks ID’s which is used to calculate the weight for each stock. Where the weight for any stock is the stock ID divided by the sum of stocks ID’s. Then in the second for-loop we calculate the value of each stock return multiplied by its weight and add it to the variable named “re”. Finally the function returns the Return value to be assigned as the portfolio expected return.

```
public void Calculate_Return()
{
    int sum = 0;
    double re = 0.0;
    for (int i = 0; i < Stocks.Count; i++)
        sum += Stocks[i].ID;
    for (int i = 0; i < Stocks.Count; i++)
    {
        re += Stocks[i].Return * Stocks[i].ID / sum;
    }
    Return = re;
}
```

Figure 15: Calculate Return Code.

Figure 16 shows the code used to calculate the portfolio risk. To simplify the risk equation we divided the equation into three loops, in the first for-loop we calculate the weights of the stocks

and add them into a list for an easier access in the later loops. In the second loop we calculated the terms which contain the square value of the weight multiplied by the square value of the stock risk then add them into the variable named “variance”.

```
public void Calculate_Risk()
{
    List<double> weight = new List<double>();
    double variance = 0.0;
    double sum = Stocks[0].ID + Stocks[1].ID + Stocks[2].ID + Stocks[3].ID + Stocks[4].ID + Stocks[5].ID;
    for (int i = 0; i < 6; i++)
    {
        double temp = 0.0;
        temp = Convert.ToDouble(Stocks[i].ID) / sum;
        weight.Add(temp);
    }
    for (int i = 0; i < 6; i++)
    {
        variance += weight[i] * weight[i] * Stocks[i].Risk * Stocks[i].Risk;
    }
    for (int i = 0; i < 5; i++)
    {
        for (int j = i+1; j < 6; j++)
        {
            variance += 2 * weight[i] * weight[j] * Covariance(i, j);
        }
    }
    Risk = Math.Pow (variance,0.5);
}
```

Figure 16: Calculate risk code.

The last for-loop is used to calculate the terms where the covariance is required, as we can see the loop calls a function named “Covariance (i,j)” which is used to calculate the covariance between the stocks referred to as “i” and “j”.

```
private double Covariance(int i, int j)
{
    double cov = 0.0;
    for (int k = 0; k < Stocks[i].Daily.Count; k++)
    {
        cov += (Stocks[i].Daily[k] - Stocks[i].Return) * (Stocks[j].Daily[k] - Stocks[j].Return);
    }
    return cov/Stocks[i].Daily.Count;
}
```

Figure 17: Covariance Code.

Figure 17 shows the code used to calculate the covariance between two stocks. The line inside the for-loop block is where the work is done, to calculate the risk we subtract the expected return of the stock from its daily value then multiply the result by the difference between the second stock daily value and its expected return then add the result to the variable named “cov”. This process is repeated for the number of available daily data for the stocks. Finally the function divides the “cov” value by the number of available daily data and returns the result as the covariance between the stocks “i” and “j”.

```
private void applyGeneticPhase1ToolStripMenuItem_Click(object sender, EventArgs e)
{
    double ret = (Convert.ToDouble(trackBar1.Value) / 10);
    double risk = 1 - Convert.ToDouble(trackBar1.Value) / 10;
    for (int i = 0; i < Portfolios.Count; i++)
        Portfolios[i].Calculate_Fitness(ret,risk);
    Select_Top_10(GP1_Portfolios);
    DataTable table = ConvertPortToDataTable(GP1_Portfolios);
    dataGridView4.DataSource = table;
    Extract_GP1_Stocks();
    CrossOver();
    DataTable table2 = ConvertPortToDataTable(GP2_Portfolios);
    dataGridView5.DataSource = table;
}
```

Figure 18: Apply Genetic Phase1 Code.

Figure 18 shows the code to apply HGA and the crossover operation. The first line reads the return weight selected from the track bar shown in the application interface, while the second line calculates the weight of the risk weight from the same track bar. Then the for-loop is used to calculate the fitness for each portfolio given the risk and return weights as selected in the track bar.

```
public void Calculate_Fitness(double w1, double w2)
{
    Fitness = w1 * Return + w2 * Risk;
}
```

Figure 19: Calculate Fitness Code.

Figure 19 shows the fitness calculation code which is simply the result of multiplying the weight of risk with the expected risk of the portfolio added to the result of multiplying the risk weight by the expected portfolio risk. Then, we call the function “Select_Top_10” passing the list of portfolios named “GP1_Portfolios” which is going to contain the top 10 portfolios.

```
private void Select_Top_10(List<Portfolio> p)
{
    for (int i = 0; i < 10; i++)
        p.Add(Portfolios[i]);
    for (int i = 10; i < Portfolios.Count; i++)
    {
        for (int j = 0; j < 10; j++)
        {
            if (Portfolios[i].Fitness > p[j].Fitness)
            {
                Insert_Port(Portfolios[i],j);
                break;
            }
        }
    }
}
```

Figure 20: Select Top 10 Portfolios Code.

Figure 20 shows the code to select the top 10 portfolios. In the first for-loop we add the first 10 portfolios to the list, then in the second loop we compare each portfolio fitness with the fitness of the selected 10 portfolios and if the fitness of any portfolio is greater than the fitness of one of the selected portfolios, we replace the selected portfolio with the one with the greater fitness and break from the comparison loop to prevent replacing more than one portfolio with the same portfolio. The two lines of code after the top 10 selection are used to format and view the data in the interface grid view. Then we use a function called “Extract_GP1_Stocks” followed by a function called “CrossOver”.

```
private void Extract_GP1_Stocks()
{
    for (int i = 0; i < GP1_Portfolios.Count; i++)
    {
        for (int j = 0; j < 6; j++)
        {
            if (!GP1_Stocks.Contains(GP1_Portfolios[i].Stocks[j]))
                GP1_Stocks.Add(GP1_Portfolios[i].Stocks[j]);
        }
    }
    DataTable table = ConvertListToDataTable(GP1_Stocks);
    dataGridView3.DataSource = table;
}
```

Figure 21: Extract GP1 Stocks Code.

Figure 21 shows the code used to extract the GP1 stocks. GP1 is the name of the list that contains the top 10 selected portfolios based on the highest fitness. This function extracts the stocks that are used to create these top 10 portfolios, which are the genes to be used in the crossover operation later on. In our application, we list those genes in the bottom left grid view. The function checks if the list names “GP1_Stocks” contains the stocks of each portfolio and if it does not contain that stock, it adds the stock into the list, then it updates the grid view shown in the application interface.

```
private void Crossover()
{
    for (int i = 0; i < GP1_Stocks.Count*6; i++)
    {
        Portfolio temp_portfolio = new Portfolio();
        Stock temp_stock = new Stock();

        while (temp_portfolio.Stocks.Count < 6)
        {
            Random rnd = new Random();
            temp_stock = GP1_Stocks[rnd.Next(0, GP1_Stocks.Count)];
            if (!temp_portfolio.Stocks.Contains(temp_stock))
                temp_portfolio.Stocks.Add(temp_stock);
        }
        Insert_Port2(temp_portfolio);
    }
}
```

Figure 22: Crossover Code.

Figure 22 shows the code used to create the crossover operation which is the core function of HGA. The crossover operation is the operation in which the genes of the optimum selected chromosomes are swapped between chromosomes in promises of generating a more optimum chromosome. Since the crossover is basically a genes swapping operation, instead of swapping a single gene at a time, we extracted the genes and created a pool of genes using the “Extract_GP1_Stocks” function. In this function, we use those genes to create a new generation of chromosomes “portfolios”. At first, we create a temporary portfolio and temporary stock. Then, we random a number between 0 and the number of extracted stocks. Then, we add that stock to the portfolio after checking if the portfolio already contains that stock or not, then we make sure that the portfolio we have created does not match another portfolio that have been created earlier. Then, we repeat this operation to generate six times the number of

extracted stocks, and then apply genetic phase2 and select the top 10 portfolio from the portfolios.

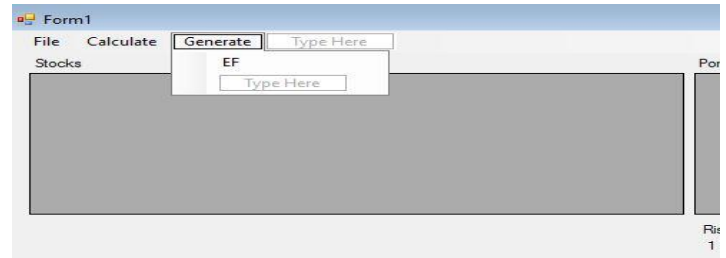


Figure 23: Generate Menu.

Figure 23 shows the generate menu which contains only one option used to generate the efficient frontier graph, which is shown in the bottom left interface of the application. A point worth mentioning here that, we have to normalize the risk and return values for the portfolios so we can generate an efficient frontier graph.

Figure 24 shows the code used to normalize the risk and return values for each portfolio. The normalized value of the return is the result of dividing the portfolio return by the maximum return in the portfolios. While the normalized risk is the result of dividing the portfolios minimum risk by the portfolio expected risk. To do this, we started by finding the maximum risk, minimum risk, maximum return and minimum return. Then we shifted the intervals to a positive interval in order to prevent the divide by 0 mathematical errors and calculated the normalized values.

```
public void Normalize(List<Portfolio> p)
{
    double MaxReturn = double.MinValue;
    double MinReturn = double.MaxValue;
    double MaxRisk = double.MinValue;
    double MinRisk = double.MaxValue;
    for (int i = 0; i < p.Count; i++)
    {
        p[i].Calculate_Risk();
        p[i].Calculate_Return();
    }
    for (int i = 0; i < p.Count; i++)
    {
        if (p[i].Risk < MinRisk)
            MinRisk = p[i].Risk;
        if (p[i].Risk > MaxRisk)
            MaxRisk = p[i].Risk;
        if (p[i].Return < MinReturn)
            MinReturn = p[i].Return;
        if (p[i].Return > MaxReturn)
            MaxReturn = p[i].Return;
    }
    MaxReturn = MaxReturn + Math.Abs(MinReturn);
    MinRisk = 1;
    this.Normalized_Return = (Return+ Math.Abs(MinReturn)) / MaxReturn;
    this.Normalized_Risk = MinRisk / (Risk+ Math.Abs(MinRisk)+1);
}
```

Figure 24: Normalization Code.

Results Obtained using the Hybrid Genetic Algorithm

After we have discussed how the code works, in this section we are going to run the program to get the results and discuss them.

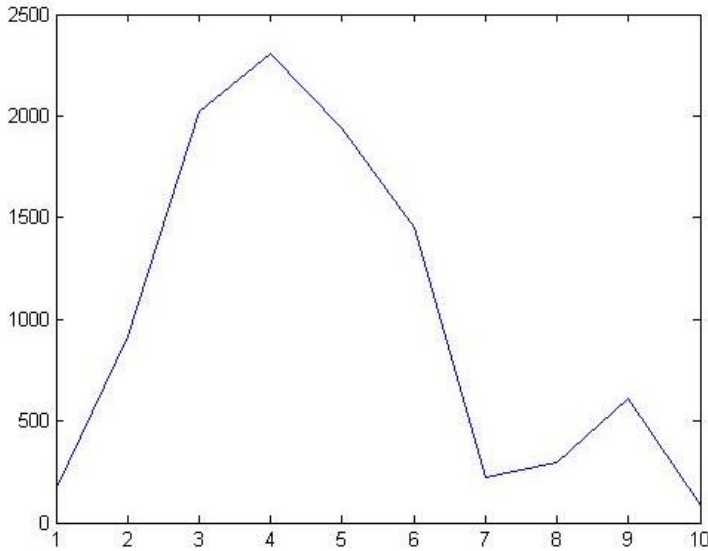


Figure 25: Return Histogram.

Figure 25 shows the 10,000 portfolios expected return histogram, where the Y-axis represents the frequency and the X-axis represents the expected return multiplied by 100. The histogram is a plot represents each portfolio expected return versus its frequency [51-57].

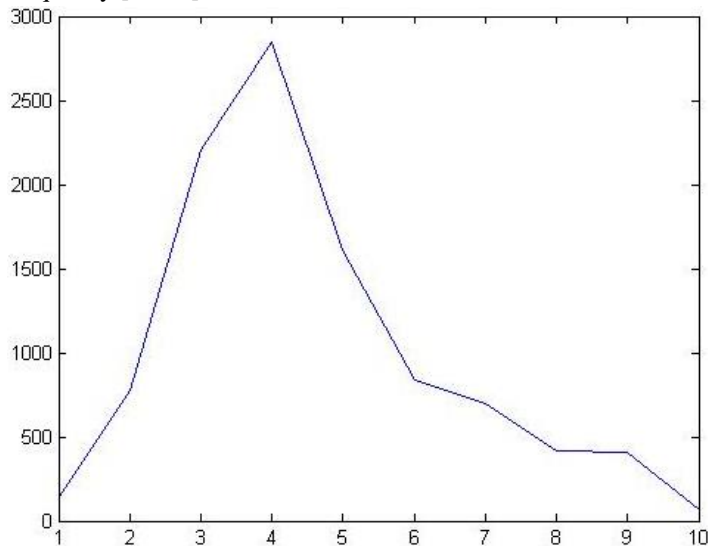


Figure 26: Return Histogram.

Figure 26 shows the 10,000 portfolios risk histogram, where the Y-axis represents the frequency and the X-axis represents the risk multiplied by 100. The histogram is a plot represents each portfolio risk versus its frequency.

To obtain our results, we applied five different tests. In each test, we selected different weights for the risk and return for portfolio fitness calculation. The weight 0.5-0.5 will display here and other weights in appendix 1.

0.5-0.5 Risk & Return weights

In this test we selected a weight of 0.5 for each of the risk and the return in the fitness function. Using these weights we obtained the following top 10 portfolios selected from the created 10,000 portfolios, the selection was based on the higher fitness [58-60].

1	Top 10 portfolios selected from the 10000 portfolios:									
2	1010001	1111000	1000101	1101101	1110111	1100001	0.00771894662218939	0.0946752554621849		
3	1111000	1000101	1101101	1110111	1100001	1001011	0.00837236649310477	0.0932574499151104		
4	1100110	1010001	1111000	1000101	1101100	1110110	0.00624016282231191	0.0916182892976589		
5	1100111	1010001	1111000	1000101	1101100	1110110	0.00564187490618866	0.0920396711185309		
6	1010001	1111000	1000101	1101100	1110110	1100000	0.00599002959296613	0.0914739476351351		
7	1010001	1111000	1000101	1101100	1110110	1100001	0.00692759672921441	0.0918478178752108		
8	1100111	1010001	1111000	1000101	1101101	1110110	0.00561666820085502	0.0921562816666667		
9	1111000	1000101	1101100	1110110	1100000	1001011	0.00654011026950142	0.0900161058020478		
10	1111000	1000101	1101100	1110110	1100000	1001010	0.00638947117316662	0.0901506017094017		
11	1100111	1010001	1111000	1000101	1101101	1110111	0.00657791258693496	0.0948382429284526		
12										

Figure 27: Top 10 Portfolios Selected from the 10,000 Portfolios.

Figure 27 shows the portfolios that are selected as the top 10 portfolios from the 10,000 portfolios we created based on the fitness function. The first six segments represent the chromosomes of the portfolio, where each chromosome represents a stock. The last two segments are the risk and the return respectively. Each chromosome is converted into the equivalent decimal number which represents the stock ID, and then the ID is used to calculate the stock weight in the portfolio.

The portfolio chromosome is [1010001 1111000 1000101 1101101 1110111 1100001], this chromosome is composed of six different genes, when each gene is converted to its equivalent decimal number the portfolio stocks IDs are: [81 120 69 109 119 97]. Therefore, the weights for each stock are calculated as follows:

$$v_1 = 1010001 \rightarrow v_1 = 81$$

$$v_2 = 1111000 \rightarrow v_2 = 120$$

$$v_3 = 1000101 \rightarrow v_3 = 69$$

$$v_4 = 1101101 \rightarrow v_4 = 109$$

$$v_5 = 1110111 \rightarrow v_5 = 119$$

$$v_6 = 1100001 \rightarrow v_6 = 97$$

The share of each company in this portfolio is obtained as follows:

$$x_1 = \frac{81}{81 + 120 + 69 + 109 + 119 + 97} = 13.6\%$$

$$x_2 = \frac{120}{81 + 120 + 69 + 109 + 119 + 97} = 20.2\%$$

$$x_3 = \frac{69}{81 + 120 + 69 + 109 + 119 + 97} = 11.6\%$$

$$x_4 = \frac{109}{81 + 120 + 69 + 109 + 119 + 97} = 18.3\%$$

$$x_5 = \frac{119}{81 + 120 + 69 + 109 + 119 + 97} = 20.0\%$$

$$x_6 = \frac{97}{81 + 120 + 69 + 109 + 119 + 97} = 16.3\%$$

Then we use the top 10 portfolios to extract the genes which we are going to use in the crossover operation.

Top 10 portfolios stocks

131003	81	1010001
141004	120	1111000
141019	69	1000101
111004	109	1101101
141141	119	1110111
142041	97	1100001
141059	75	1001011
111001	102	1100110
141014	108	1101100
131062	118	1110110
141081	103	1100111
111033	96	1100000
141042	74	1001010

Figure 28: Genes and Stocks used to Create Top 10 Portfolios.

Figure 28 shows the genes and the stocks that were extracted from the top 10 portfolios. After extracting the genes we used them in the crossover operation to create the second generation of the portfolios, and then we selected the top 10 portfolios of the second generation based on the fitness function.

Figure 29 shows the top 10 portfolios of the crossover results. These portfolios are used to draw the efficient frontier graph.

The HGA proposed in was programmed in C# and applied for the data collected from Amman Stock Exchange. The data adopted from this market for the year 2015. Using HGA to solve the Markowitz' portfolio optimization problem, 10 portfolios are obtained.

Crossover top 10 portfolios

1110111	1110110	1100110	1100000	1101101	1111000	0.00646079742249487	0.134291109939759
1110111	1110110	1100110	1001010	1101101	1111000	0.00675520227248665	0.129273693146417
1000101	1100110	1110111	1110110	1111000	1001010	0.0074520919325518	0.118732436877076
1111000	1001011	1110111	1110110	1010001	1100000	0.00731863679208056	0.12119884729064
1100001	1110110	1111000	1001010	1110111	1000101	0.00817775246464712	0.119187514237856
1010001	1110111	1110110	1100110	1001010	1111000	0.00707252922642634	0.121227885993485
1110111	1110110	1001011	1100000	1101101	1111000	0.00695421613566113	0.12930908477237
1100110	1100000	1110110	1111000	1000001	1101101	0.00678583435607438	0.10551735046729
1100110	1100001	1110110	1111000	1001010	1110111	0.00787473981489722	0.126203511111111
1001010	1110111	1110110	1100110	1100000	1111000	0.00712075374560816	0.125906252782194

Figure 29: Top 10 second generation portfolios.

Figure 30 illustrates the generated efficient frontier by the proposed HGA. Overall, the annual return of optimal portfolios

varies between 10.552% and 13.429%, while their risk changes between 0.646% and 0.818%. A decision-making technique based on AHP is proposed in the next section which helps decision-maker to select the most suitable portfolio from among 10 portfolios.

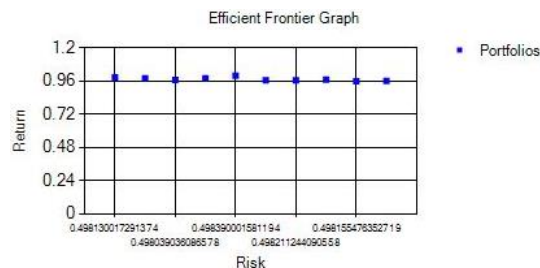


Figure 30: Efficient frontier graph.

Portfolio Selection via Analytic Hierarchy Process

The proposed hierarchical structure of the optimum portfolio selection problem along with the alternatives obtained by HGA and the identified criteria are depicted in Figure 31.

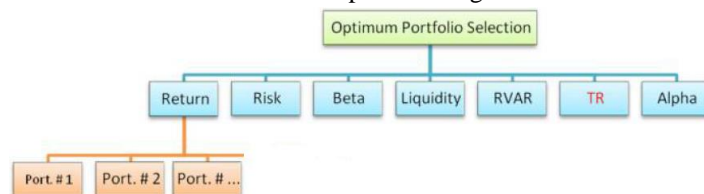


Figure 31: Proposed Hierarchy of the Portfolio Selection Problem.

As seen, the decision hierarchy consists of three levels. The optimum portfolio selection is the prime objective of the problem and takes place at the topmost level (Level 1) of the hierarchy. The seven criteria: return, risk, beta ratio, liquidity, RVAR, TR and alpha ratio take place at the second level. Finally, the 10 portfolios identified by HGA take place at the most bottom level (Level 3) as decision alternatives [61-64] (Table 2).

Table 2: Portfolios and stocks.

Stocks				Portfolios							
Code	Return	ID	Risk	Stock 1	Stock 2	Stock 3	Stock 4	Stock 5	Stock 6	Return	Risk
141004	0.0044659	120	0.021400665	108	86	76	99	105	96	-0.00028080817...	0.005031406138...
141141	0.00164823	119	0.01990755	110	92	116	60	109	98	-0.0022431459...	0.007156914128...
131062	0.00151779	118	0.0088022	86	103	69	67	84	72	-0.00087605773...	0.006311575609...
141086	0.00130643	117	0.020041208	100	88	115	81	69	96	-0.00034561690...	0.007644281441...
141052	0.00082059	116	0.015583422	85	73	100	66	115	81	-0.00051415842...	0.006700806926...
				99	97	114	80	68	95	-0.00034924678...	0.009581333335...
Code	Return	ID	Risk	Stock 1	Stock 2	Stock 3	Stock 4	Stock 5	Stock 6	Return	Risk
141002	0.00042194	114	0.035233385	114	120	86	74	62	91	0.000359496398...	0.011134194209...
141004	0.0044659	120	0.021400665	114	92	70	80	120	68	0.000501477389...	0.00973373696...
111007	-0.0006873	86	0.013549322	114	62	102	112	90	68	-0.00052579711...	0.011069403727...
141042	-0.00112189	74	0.017433938	114	93	71	81	120	69	0.000521492536...	0.010132882089...
141055	-0.00325519	62	0.029532228	60	70	110	120	98	76	0.000290736029...	0.010011481324...
				75	114	92	102	80	120	0.00053039622...	0.00963866694...

Table 3 shows the top 10 portfolios that resulted from the (GA) operation and their criteria which mentioned before, and it should

be noted that the Sharpe and Tryner for the market portfolio was calculated and they were (-8.646) and (-0.038049125) respectively.

Table 3: Seven criteria's portfolios.

Por.#	Risk	Return	liquidity	Sharpe	Beta	TR	Alpha
1	0.646%	13.429%	1.702	14.904	0.153	0.631	0.102
2	0.676%	12.927%	1.815	13.512	0.191	0.478	0.099
3	0.745%	11.873%	2.159	10.834	0.197	0.410	0.088
4	0.732%	12.120%	2.462	11.368	0.295	0.282	0.094
5	0.818%	11.919%	2.301	9.928	0.146	0.554	0.087
6	0.707%	12.123%	2.018	11.768	0.281	0.296	0.094
7	0.695%	12.931%	1.913	13.130	0.211	0.434	0.099
8	0.679%	10.552%	0.860	9.950	0.133	0.507	0.073
9	0.787%	12.620%	1.966	11.201	0.335	0.263	0.101
10	0.712%	12.591%	1.843	12.345	0.166	0.529	0.094

Table 4: Portfolios optimality.

Portfolio #	Optimality Order
1	2
2	7
3	7
4	6
5	3
6	5
7	7
8	1
9	7
10	4

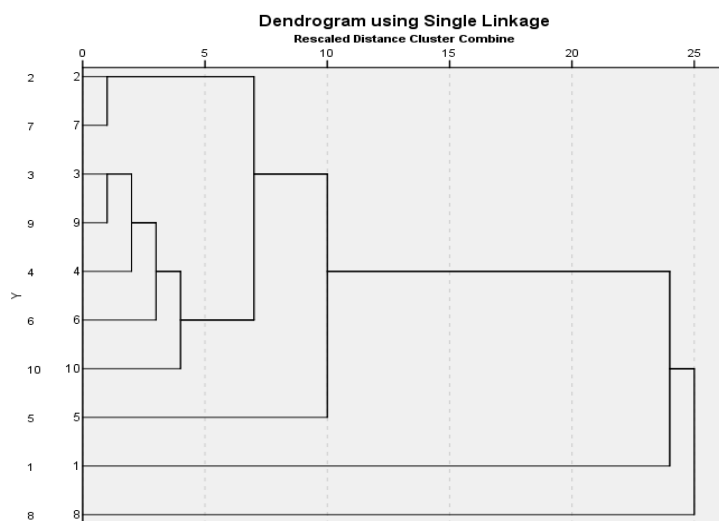


Figure 32: Portfolio hierarchy analysis.

Figure 32 shows the result of the hierarchy analysis process, where the (Y) axis represents the portfolio number and the (X) axis represents the relative weight of the portfolio in terms of preference. As illustrated by the figure the optimum portfolio is portfolio number (8).

Therefore, the portfolios can be displayed according to its optimality as in the following (Table 4):

The study provides the process and implementation of HGA, results obtained using the HGA and portfolios selection via AHP. The following section presents conclusion, and policy implications of the study.

Conclusion and Policy Implications

The main objective of this study is to investigate the ability of a Hybrid Genetic Algorithm and Analytic Hierarchy Process in selecting the optimum portfolio. The study creates portfolios using the daily return of the companies listed in Amman Stock Exchange, during the period from January 1, 2015 to December 31, 2015. With reference to the findings of the analysis, the findings might be list down as follows [65]:

1. Hybrid Genetic Algorithm can identify the portfolios that being in the efficient frontier by using Jordanian trading data.
2. Hybrid Genetic Algorithm does not have any restriction as far as the number of assets, is concerned.
3. Hybrid Genetic Algorithms have advantage over problems for the portfolio selection cases which scale of the problem or the nonlinear constraints of the problem unable us to use linear or quadratic models.

4. Analytic Hierarchy Process can select the optimum portfolio among the portfolios obtained by HGA.
5. Finally, the selected optimal portfolio achieve a return of 10.552% with a risk of 0.679%, where the liquidity of this portfolio is 0.860, Sharpe ratio 9.950, Beta 0.133, Trynor ratio 0.507 and Alpha ratio 0.073.

Policy Implications

The results discussed above can lend support the following recommendations:

1. To make portfolio selection and optimization problems more accurate, we can present other main parameters such as taxes and transaction costs in order to make more perfect decisions.
2. AHP in this thesis includes seven criteria. This hierarchy can be more completed by adding other quantitative or qualitative criteria not used here.
3. Individuals, Investors and governments can employ this method to construct optimized portfolio and modify their portfolios investment depending on their Investment strategy.

Declaration of Interest

The author has no relevant affiliations or financial involvement with a financial interest in or financial with the subject matter or materials discussed in the manuscript.

Conflicts of Interest

There is no conflict of interest.

References

1. Sharpe W. Capital Asset Prices: A Theory of market equilibrium under conditions of risk. *J Finance*. 1964; 19: 425-442.
2. Lintner J. The Valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. *Rev Econ Stat*. 1965; 1: 13-37.
3. Mossin J. Equilibrium in a capital asset market. *Econometrica J Econom Soc*. 1966; 768-783.
4. Roll R. A Critique of the asset pricing theory's tests. part i: on the past and potential testability of the theory. *J Financial Econ*. 1977; 4: 129-176.
5. Roll R. Ambiguity When performance is measured by the security market line. *J Finance*. 1978; 26: 569-585.
6. Ross S. The arbitrage theory of capital asset pricing. *J Econ Theory*. 1976; 13: 341-360.
7. Black F, Scholes M. The pricing of options and corporate liabilities. *J Political Economy*. 1973; 81: 637-654.
8. Fama E. Efficient Capital Markets: A Review of theory and empirical work. *J Finance*. 1970; 25: 383-417.
9. Fama E, Malkiel BG. Efficient markets: a review of theory and empirical work. *J Business*. 1970; 25: 383-417.
10. Haugen AS. Modern investment theory 5th edition prentice-hall inc. Upper Saddle River New Jersey. 2001; 11: 07458.
11. Maginn JL, Tuttle DL, McLeavey DW, Pinto JE. Managing investment portfolios: a dynamic process. 2nd Edn Warren Gorham Lamont. 1990.
12. Peavy J. Cases in portfolio management charlottesville va. Association for Investm Manage Res. 1990.
13. Desai RT, Lele T, Ens F. A Monte-Carlo Method for portfolio optimization under partially observed stochastic volatility. *Proceed Computat Intellig Finan Engine Compe Solutions Online Finan*. 2003.
14. Davis M, Norman A. Portfolio Selection with Transaction Costs. *Mathem Operations Res*. 1990; 15: 673-713.
15. Lee MC, Su LE. Capital market line based on efficient frontier of portfolio with borrowing and lending rate. *Universal J Accoun Finance*. 2014; 2: 69-76.
16. Hong G, Sarkar S. Equity systematic risk (beta) and its determinants. *contemporary accounting Res*. 2007; 24: 423-466.
17. Solimanpur M, Mansourfar G, Ghayour F. Optimum portfolio selection using a hybrid genetic algorithm and analytic hierarchy process. *Studies Econ Finance*. 2015; 32: 379-394.
18. Isiklar G, Büyükzkan G. Using a multi-criteria decision making approach to evaluate mobile phone alternatives. *Computer Stand Inter*. 2007; 29: 265-274.
19. Sharpe W. Mutual fund performance. *J Bus*. 1966; 39: 119-139.
20. Treynor J. How to Rate management of investment funds. *Harvard Busi Review*. 1965; 43: 63-75
21. Jensen M. The performance of mutual funds in the period 1945-1964. *J Finan*. 1968; 23: 389-416.
22. Saaty TL. The analytic hierarchy process. McGraw Hill Company NY. 1980.
23. Akcay Y, Yalcin A. Optimal portfolio selection with a shortfall probability constraint: evidence from alternative distribution functions. *J Finan Res*. 2010; 33: 77-102.
24. Alrabadi DWH. Portfolio optimization using the generalized reduced gradient nonlinear algorithm: an application to amman stock exchange. *Internal J Islamic Middle Eastern Finan Manage*. 2016; 9: 570-582.
25. Brown KC, Reilly FK. Analysis of Investments and Management of Portfolios 9th edition McGraw-Hill Irwin. 2008.
26. Carrasco M, Noumon N. Optimal portfolio selection using regularization. *Discussion paper*. 2011.

27. Chyi L, Robinson J, Reed R. Downside beta and the cross-sectional determinants of listed property trust returns. *J Real Estate Portfolio Manage.* 2007; 14: 49-62.
28. De Bondt WF, Thaler R. Does the stock market overreact. *The J Finan.* 1985; 40: 793-805.
29. DeMiguel V, Plyakha Y, Uppal R, Vilkov G. Improving portfolio selection using option-implied volatility and skewness. *J Finan Quantita Anal.* 2013; 48: 1813-1845.
30. Edirisinghe NCP. Index-tracking optimal portfolio selection. *Quantita Finan Lett.* 2013; 1: 16-20.
31. Edwin E, Martin J. Modern portfolio theory 1950 to date. *J Banking Finan.* 1997; 21: 1743-1759.
32. Ehrgott M, Klamroth K, Schwehm C. An MCDM approach to portfolio optimization. *Euro J Oper Res.* 2004; 155: 752-770.
33. Elton J, Gruber M, Brown S, Goetzmann W. Modern portfolio theory and investment analysis. 5th ed NYJohn Wiley Sons Inc.1995.
34. Mihoub T, Hopgood A, Nolle L, Battersby A. Hybrid genetic algorithms: a review engineering letters. 2016; 13: 124-137.
35. Fama E. The behavior of stock prices. *J Business.* 1965; 38: 34-105.
36. Fard D. Portfolio selection and optimization with genetic algorithm: the case of alborz insurance company. Allame Tabatabai University Iran. 1965.
37. Nasab G, Ahari S, Makui A. A Portfolio selection using fuzzy analytic hierarchy process: a Case study of iranian pharmaceutical industry. *Intern J Indust Engine Comput.* 2011; 2: 225-236.
38. Goldberg D, Deb K, Korb B. Messy genetic algorithms: motivation analysis First Results. *Complex systems.* 1989; 1: 493-530.
39. Hagstrom RG Jr. The Essential Buffett: Timeless principles for the new economy. John Wiley Sons NY. 2001.
40. Harry MM. Autobiography the nobel prizes 1990 editor tore frangsmyr. Nobel Foundation Stockholm. 1991; 56-70.
41. Jang SH. A Study on performance maximization portfolio analysis technique using ahp. *Internal Rev Manage Bus Res.* 2016; 5: 225.
42. Kamal J. Optimal Portfolio selection in ex ante stock price bubble and furthermore bubble burst scenario from dhaka stock exchange with relevance to sharpe's single index model. 2012.
43. Kan R, Wang X, Zhou G. Optimal portfolio selection with and without risk-free asset. 2016.
44. Levinson M. Guide to financial markets, london: The Economist. 2006; 145-146.
45. Manganelli R. Solving the corporate value enigma: a system to unlock stockholder Value. AMACOM Division of Ameri Manage Assoc. 2002.
46. Markowitz H. Portfolio selection: efficient diversification of investments basil blackwall. NY. 1959.
47. Markowitz HM. Foundations of portfolio theory. *J Finan.* 1991; 46: 469-477.
48. Markowitz HM. Portfolio Selection - 2nd Edition. John Wiley Sons, NY. 1991.
49. Markowitz HM. Portfolio Selection. *J Finance.*1952; 7: 77-91.
50. Papahristodoulou C, Dotzauer E. Optimal portfolios using linear programming models. *J Opera Res Society.* 2004; 55: 1169-1177.
51. Pappas D, Kiriakopoulos K, Kaimakamis G. Optimal portfolio selection with singular covariance Matrix, *Internal Mathem Forum.* 2010; 5: 2305-2318.
52. Rahmani N, alebpour A, Ahmadi T. Developing amulti criteria model for stochastic it portfolio selection by ahp method. *Procedia Social Behavioral Scien.* 2012; 62: 1041-1045.
53. Reilly F, Brown K. Investment analysis and portfolio management tomson. South Western Pub Mason. 2006.
54. Rytchkov O. Time-varying margin requirements and optimal portfolio choice. *J Finan Quantita Analy.* 2016; 51: 655-683.
55. Saaty R. The Analytic hierarchy process-what it is and how it is used. *Mathem modelling.* 1987; 9:161-176.
56. Sahu R, Jain M, Garg G. Optimal portfolio allocation using portfolio theory and heuristics driven evolutionary technique. *J Adv Manage Res.* 2006; 3: 81-87.
57. Scholz H. Refinements to the Sharpe ratio: Comparing alternatives for bear markets. *J Asset Manag.* 2007; 7: 347-357.
58. Taleb N. Black swans and the domains of statistics. *The Am Stat.* 2007; 61: 198-200.
59. Vargas L. An overview of the analytic hierarchy process and its applications. *Euro J Operat Res.* 1990; 48: 2-8.
60. Wan S. Optimal portfolio model under compound jump processes. *Kybernetes.* 2009; 38: 522-532.
61. Yao X, Liu Y. Fast evolutionary programming. *Evolutionary Programming.* 1996; 3: 451-460.
62. Yi B, Viens F, Law B, Li Z. Dynamic portfolio selection with mispricing and model ambiguity. *Annal Fin.* 2015; 11: 37-75.
63. www.ase.com.jo.
64. www.jsc.gov.jo.
65. www.sdc.com.jo.