

# Algorithm of construction of optimum portfolio of stocks using genetic algorithm

Pankaj Sinha · Abhishek Chandwani ·  
Tanmay Sinha

Received: 4 January 2014 / Revised: 3 September 2014 / Published online: 20 September 2014

© The Society for Reliability Engineering, Quality and Operations Management (SREQOM), India and The Division of Operation and Maintenance, Lulea University of Technology, Sweden 2014

**Abstract** The objective of this paper is to develop an algorithm to create an optimum portfolio from a large pool of stocks listed in a single market index SPX 500 Index: USA (for example) using genetic algorithm. The algorithm selects stocks on the basis of a priority index function designed on company fundamentals, and then genetically assigns optimum weights to the selected stocks by finding a genetically suitable combination of return and risk on the basis of historical data. The effect of genetic evolution on portfolio optimization has been demonstrated by developing a MATLAB code to implement the genetic application of reproduction, crossover and mutation operators. The effectiveness of the obtained portfolio has been successfully tested by running its performance over a 6 month holding period. It is found that genetic algorithm is successful in providing the optimum weights to stocks which were initially screened through a predetermined priority index function. The constructed portfolio beats the market for the considered holding period by a significant margin.

**Keywords** Optimum portfolio · Genetic algorithm · Portfolio construction · MATLAB

---

P. Sinha (✉)  
Faculty of Management Studies, University of Delhi, New Delhi,  
India  
e-mail: pankaj-sinha@fms.edu

A. Chandwani  
Indian Institute of Technology Kharagpur, Kharagpur, India  
e-mail: chandwani.abhishek@iitkgp.ac.in

T. Sinha  
Jaypee Institute of Information Technology, Noida, India  
e-mail: tanmay3011@gmail.com

## 1 Introduction

### 1.1 Multiobjective portfolio construction

Multiobjective portfolio optimization problem as discussed in this paper is the construction of a portfolio of stocks selected from a large pool of stocks which provide maximum return at minimum risk. Modern portfolio theory provides a well-developed paradigm to form such a portfolio; however, constructing an optimum portfolio with limited available capital is a challenge when a large pool of stocks is taken into account. In the capital market, there are thousands of equities and depending on the financial characteristics of the equity, the risk and return of the investment are dissimilar. In order to construct an optimum and profitable portfolio of stocks, risk and return should also be simultaneously considered and hence, portfolio optimization is a complex multiobjective problem of multistage decision-based. In this paper, the multistage decision-based genetic algorithm is proposed for the multiobjective portfolio optimization problem. On the basis of the application of the algorithm on S&P US 500, we shall show the effectiveness of the proposed algorithm is validated for solving this problem. The Sect. 1.2 shall explain about the Markowitz theory of portfolio optimization, considered as the cornerstone of the portfolio theory.

### 1.2 Markowitz theory of portfolio optimization

The multiobjective portfolio optimization problems have been an area of research since 1952. Markowitz (1952), a creator of modern portfolio theory, originally formulated the fundamental theorem of mean–variance portfolio framework, which explains the trade-off between mean and

**Table 1** Generation of reproduction pool

| Chromosome (CH) | Fitness value | Evaluation function | Cumulative probability | Random # generated | Chromosome (CH) selected | # of copies of CH received |
|-----------------|---------------|---------------------|------------------------|--------------------|--------------------------|----------------------------|
| CH1             | 1.79          | 0.23                | 0.23                   | 0.36               | CH2                      | 1                          |
| CH2             | 1.95          | 0.25                | 0.48                   | 0.16               | CH1                      | 1                          |
| CH3             | 2.10          | 0.27                | 0.72                   | 0.57               | CH3                      | 2                          |
| CH4             | 1.87          | 0.25                | 1.00                   | 0.63               | CH3                      | 0                          |

variance each representing expected returns and risk of a portfolio, respectively. The mean–variance approach proposed by (Markowitz 1952) was to deal with the portfolio selection problem which can determine the weights to be allotted to each equity on the basis of minimum required rate of return.

Mean variance theorem:

The formulation of the mean–variance method can be described as follows:

$$\min \sigma_p = \min \sum_{i=1}^n \sum_{j=1}^m \sigma_{ij} x_i x_j \quad (1)$$

$$\text{s.t. } \sum_{i=1}^n u_i x_i \geq E, \sum_{i=1}^n x_i = 1, x_i \geq 0$$

$$i = 1, 2, \dots, n$$

where  $\sigma_p$  denotes the portfolio risk,  $\sigma_{ij}$  denotes the covariance between the return of the  $i$ th security and the  $j$ th security,  $\mu_i$  denotes the expected return rate of the  $i$ th security,  $E$  denotes the acceptable least rate of the expected return,  $x_i$  denotes the investment portion in the  $i$ th securities.

For a given specific return rate, one can derive the minimum investment risk by minimizing the variance of a portfolio; or for a given risk level which the investor can tolerate, one can derive the maximum returns by maximizing the expected returns of a portfolio. The main input data of the Markowitz mean–variance model are expected returns and variance of expected returns of these securities as presented by Ehrgott et al (2004). Although Markowitz's theory uses only mean and variance to describe the characteristics of return, his theory about the structures of a portfolio became a cornerstone of modern portfolio theory. However as the complexities of financial markets increased, we have realized the model has some practical limitations as explained in the Sect. 1.3

### 1.3 Limitations of mean–variance model

The mean variance model requires either the acceptable least rate of return or the variance of the portfolio for which the expected return shall be computed. In the paper, we have developed a genetic approach which eliminates the need to estimate the above two parameters.

Mean variance model is not practical for a large number of securities because of the computational complexity involved which increases solving time, a key parameter for algorithmic trading. Hence we have proposed a genetic algorithm based solution to the problem of construction of multiobjective optimum portfolio as explained in the Sect. 1.4.

The mean variance model which involves minimizing risk as shown in Eq. 1 is based on the assumption that the returns are normally distributed. It considers simple linear correlations of returns which is equivalent of assuming that the 'joint' distribution of asset returns is (multivariate) normal. But it is found that returns of many assets are not normally distributed in today's financial markets due to influence of political factors, market regulations, economic policies, and nonsystematic risk. Which violates the covariance conditions of mean–variance model. Hence the solution produced by the mean variance theorem is skewed, thus creating the need for an alternative approach. The paper proposes genetic algorithm as the solution to the problem as it is not based on the assumption of normal returns.

### 1.4 Decision to use genetic algorithm

The work presented in this paper is motivated by the need to develop an optimization algorithm which is efficient in managing portfolios consisting of a large number of stocks.

The algorithms earlier used require user inputs on either expected return or acceptable risk which is difficult to estimate for large portfolios.

This limits the solution to the present value of either return or risk as received from the user. In order to eliminate this predicament, the authors have presented in this paper the multistage decision-based genetic algorithm approach for dealing with the multiobjective portfolio optimization problem. Firstly, we select the short list of the stocks by the priority index (as explained in section) and then genetic algorithm is applied to decide the investment weight of the stocks.

The approach reduces the computational complexity and generates the most optimum portfolio implementing genetic theory.

Genetic algorithm uses an objective function, explained in Sect. 1.7.1 to calculate the fitness value. The stocks are optimized using the genetic evolution process determined by these fitness values. This eliminates the need to solve for minimum risk under a constraint of expected return. Thus genetic algorithm is valid for non normal return data as well.

### 1.5 Advantages of GA

Evolutionary algorithms offer a number of advantages over the traditional optimization methods. They can be applied to problems with a non-differentiable or discontinuous objective function, to which gradient-based methods such as Gauss–Newton would not be applicable. They are also useful when the objective function has several local optima. The basic feature of genetic algorithms is the multiple directional and global searches, in which a population of potential solutions is maintained from generation to generation as discussed by Gen M et al (1997).

The so-called schema theorem shows that a genetic algorithm automatically allocates an exponentially increasing number of trials to the best observed schemata (see Lee (2011)).

The population-to-population approach is beneficial in the exploration of the securities optimal portfolio selecting solutions. It is this property which allows the algorithm to find a global solution without the constraints of a minimum acceptable return thus providing the best and most acceptable solution.

Another useful feature of GA is to handle multiobjective function optimization. The multistage decision based process is explained in Sect. 1.6.

### 1.6 The proposed solution

The process of genetic evolution which has been verified by the laws of nature since the beginning of earth is proven to be the most intricate and beautiful optimization technique. The key elements of genetic algorithm are: creation of chromosomes, initiating parent species, creation of initial population, reproduction pool on the basis of fitness selection, genetic operators: crossover and mutation operation.

This study applies the very concepts of above genetic evolution in constructing an optimum portfolio of stocks selected from a large pool of stocks listed in the single market index. We have put in our utmost efforts to contribute significant analytical conclusion to the application of genetic algorithm on the issue of optimization of weights

of stocks selected in the portfolio. The algorithm for portfolio construction involves two stages—selection of stocks by using a priority index function and optimization of the weights of the selected stocks. The process initially selects stocks for the portfolio on the basis of fundamentals of the company using a priority index function and then it optimizes the weights of the selected stocks by a genetic approach, where the selected stocks were allowed to genetically evolve towards the fittest population taking into account both risk and return of stocks. A systematic and computational mode of Darwinian evolution has been applied in this research paper. The technique has been explained in Sect. 1.7. The optimization of stocks will be adjusted as per the stocks undergo a genetic evolution through the process of reproduction and mating using crossover and mutation.

The formulation of genetic algorithm is represented in Fig. 1. The elaborated explanation is presented in Sect. 1.7.

### 1.7 Genetic algorithm

Holland (1975) provides the theoretical foundation of genetic algorithms. The entire process of optimization by genetic algorithm is explained in the following steps:

The process was initiated by starting a random solution called ‘Population’ which consists of chromosomes. Each chromosome represents a solution of the problem with genes representing the weight given to each selected stock. The genes of the chromosome are string (not necessarily

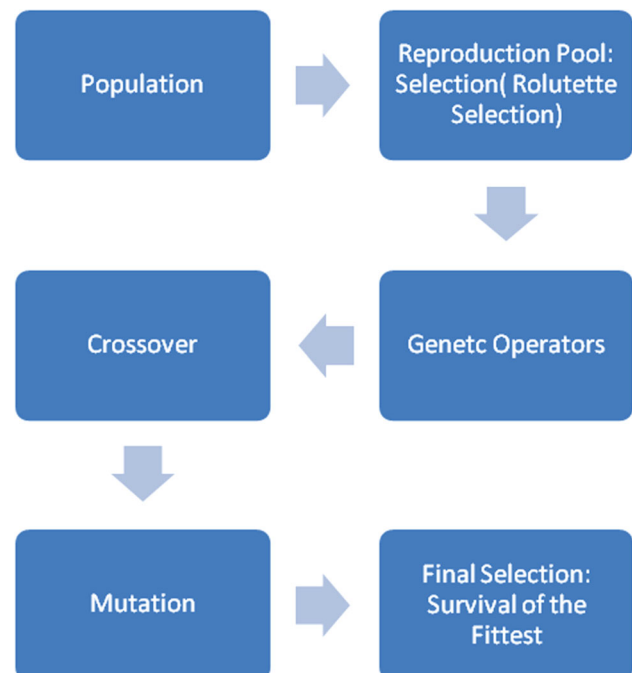


Fig. 1 Genetic evolution mechanism

binary) of symbols, in this case it is representing the randomly assigned weights of selected stocks. These chromosomes are then made to evolve genetically through iterations giving rise to newer populations. We compare the fitness values of evolving chromosomes with the existing chromosomes, after each iteration. The objective function used to calculate fitness is described in the following section. In this study we evaluate the objective function over a large population size such that the objective function attains the maximum possible value.

After all the iterations are over, we calculate the fitness value of each of the chromosomes and select the chromosome which attains the maximum fitness value, in accordance with Darwin's (1859) theory of survival of the fittest.

This selected chromosome shall represent the optimization of the weights of the stocks giving us genetically suitable return and risk.

This genetic procedure is designed to maximize the fitness value under the constraint that the sum of the weights is 1 and that all the weights are  $<1$  and  $>0$ . The genetic algorithms are typically implemented as follows

Step 1: Randomly generate an initial population of chromosomes (solutions).

Step 2: Evaluate each chromosome in the population.

Step 3: Create new chromosomes by mating current chromosomes and apply mutation and recombination when the parent chromosomes mate.

Step 4: Delete members of the population to make room for the new chromosomes.

Step 5: Evaluate the new chromosomes, and insert them into the population.

Step 6: If a stopping criterion is satisfied, then stop and output the best chromosome (solution); otherwise, go to step 3.

Figure 1 explains the above genetic evolution mechanism: The following various functions are used in the algorithm:

### 1.7.1 Objective function

The objective function is used to calculate the fitness value of the chromosomes which is used to find the fittest species.

The objective function is:

Maximize return and simultaneously minimize risk:

$$\min \sigma_p = \min \sum_{i=1}^n \sum_{j=1}^m \sigma_{ij} x_i x_j$$

$$\text{s.t. } \sum_{i=1}^n u_i x_i \geq E, \sum_{i=1}^n x_i = 1, x_i \geq 0$$

$$i = 1, 2, \dots, n$$

where  $x_i$  is the weight of  $i$ th stock,  $u_i$  is the average daily return of  $i$ th stock, Average daily return ( $r_i$ ) is the  $\{(1 + r_1/100)(1 + r_2/100)(1 + r_3/100)\dots(1 + r_n/100)\}^{1/252} - 1$ ,  $r_1$

is the daily return as on day 1 of the period,  $r_2$  is the daily return as on day 2,  $r_n$  is the daily return as on day  $n$  (read from Table 3)

$$f(\text{chromosome}) = \frac{\text{portfolio return}}{\text{portfolio standard deviation}},$$

where  $f(\text{chromosome})$  is the fitness value of the chromosome,  $\sigma(i,j)$  is the covariance of returns between the  $i$ th and  $j$ th stocks.

The fitness value is then used to identify the fitness of the population and the process is explained in the Sect. 1.7.2.

### 1.7.2 Evaluation function

This function shall be used to create the reproduction pool from the parent population. The evaluation function is similar to the probability which decides the fitness of the species. It is the ratio of the fitness value of the  $i$ th chromosome to the sum of fitness values of all the chromosomes.

$$f(\text{evaluation } i) = \frac{f(\text{chromosome } i)}{\sum f(\text{chromosome } i)}$$

The value of the evaluation function is then used to determine the cumulative probability of that chromosome. The cumulative probabilities are used to designate the fittest chromosomes as shown in the algorithm mentioned in the Sect. 1.7.5. In Sect. 1.7.3 the paper discusses the initiation of a population of chromosomes in order to begin the genetic evolution. The process is demonstrated in Table 1.

### 1.7.3 Population initiation

A population shall consist of chromosomes which shall be made up of genes. A gene is representative of values which shall return weights of the selected stocks. The gene is constructed by random allocation of values using the normal probability distribution.

These obtained random values ( $V_i$ ) are now used to calculate weights ( $w_i$ ) of the selected stocks by calculating  $V_i / (\sum V_i)$ . These weights are the stock wise percentages of the total investment which the portfolio manager shall recommend to the user. A parent population of 10 random chromosomes representing 10 random initial solutions to the objective function is created. A reproduction pool is then created from the initial population.

### 1.7.4 Reproduction pool: the method of selection

The method shall decide the reproduction pool that is the chromosomes which shall undergo mating. There are

several available techniques like: tournament selection, roulette wheel selection, stochastic based selection and reward based selection. In this paper we have used the roulette wheel selection (also known as the fitness proportionate selection) in order to select the species on the basis of their fitness values mainly.

### 1.7.5 Roulette selection

Each chromosome is evaluated on the basis of the fitness function (as mentioned above). A random number ( $r_i$ ) is generated from normal distribution and is compared with the chromosome's cumulative probability. The chromosome  $i$  having cumulative probability such that  $p_{(i-1)} < r_i < p_i$  selected into the reproduction pool (Table 1). The reproduction pool shall consist of species which have been selected by the roulette selection method defined above. The species in the reproduction pool which then undergo genetic mating via the processes of crossover and mutation as explained in Sect. 1.7.6.

### 1.7.6 Genetic mating

**1.7.6.1 Crossover** Crossover can be performed by a variety of methods:

- (a) Single point crossover
- (b) Two point crossover
- (c) Uniform crossover
- (d) Heuristic crossover

In our research we have used arithmetic crossover due to the proven accuracy of the same.

$$\text{Offspring 1} = a * \text{CH1} + (1 - a) * \text{CH2}$$

$$\text{Offspring 2} = (1 - a) * \text{CH1} + a * \text{CH2}$$

where,  $a$  is the Any random number belonging to  $(0, 1)$

$$V'' = a * V + (1 - a) * v'$$

$$V''' = (1 - a) * v + a * v'$$

The new values ( $v''$  and  $v'''$ ) can be calculated by the formula given above, where in  $a$  is random number between 0 and 1. Figure 2 above represents arithmetic point crossover and describes how two parent chromosomes mate by crossover to give the two off springs. Each of these values shall now give different weights to the stocks 1,2,3,...,  $n$ , Hence allowing genetic evolution by crossover.

Mutation is explained in Fig. 3. As shown, the gene at the mutation point 1 shall move to the mutation point 2 while the others shall shift a position each towards the initial position of mutation point 1.

Parent Chromosome 1 (CH1):



Parent Chromosome 2(CH2):



Chromosome 1 X Chromosome 2

Offspring 1



Offspring 2



Fig. 2 Arithmetic crossover

**1.7.6.2 Decision of crossover versus mutation** This is a question which has seldom been addressed when dealing with a large data set, the problem of how to understand which species shall undergo cross-over and which shall decide to mutate.

We considered the following points before deciding on the approach to tackle this issue:

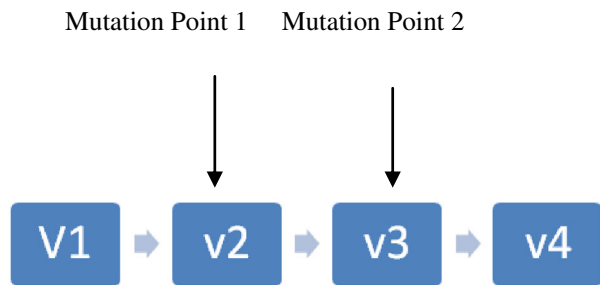
The solution depends on the nature of problem to a large extent. The solution should be oriented in such a way that the genesis contributes to evolution of the species selecting the fittest species which is decided on the basis of fitness function.

Crossover is a primarily explorative procedure, it accommodates the features of both the parents and creates a chromosome somewhere in between the parents.

Mutation is exploitative; it only creates a slight diversion with the parent and hence alters the feature locally. Only crossover can combine information from two parents where as mutation shall introduce new information into the offspring. Hence we need a lucky mutation for a perfect genesis.

**Mutation:**

Parent Chromosome



Offspring

**Fig. 3** Mutation

The paper has found on the basis of probabilistic approach that a mutation probability of  $P_m = 0.4$  and a crossover probability of  $P_c = 0.6$  shall yield the optimum solution. In case of identical chromosomes being selected in the reproduction pool, the stocks shall mutate in order to obtain a genetic advantage for the next population.

**2 Literature review**

Markovitz (1952), the father of modern portfolio theory, has established the role of combining different assets to minimize risk of the portfolio constructed via his publication. We have incorporated his conclusions in our research by keeping diversity a factor while constructing the portfolio. We achieved this by dividing the initial pool of stocks into sectors allowing the representation of each sector in the portfolio thus reducing the risk.

Melanie (1998) and Gen and Cheng (1997) explains the process of genetic algorithm in detail. They have presented various mathematical models for applying evolutionary genetic technique.

Lin and Gen (2007) stresses on the importance of taking risk as well as return into consideration while portfolio selection. The paper suggested that the multi stage genetic algorithm can be used for the portfolio optimization.

Pereira (2000) has suggested that genetic algorithms are a valid approach to optimization problems in finance. Yang (2006) advocates that genetic algorithms can be used to improve the efficiency of the portfolio. Bakhtyar et al. (2012) discussed that the choice of crossover and mutation influence the genetic algorithm performance. Seflane and Benbouziane (2012) shows that arithmetic crossover is genetically better than single point and two point crossover using an example of five stocks, while considering both return and risk in the objective function.

Sinha and Goyal (2012) have developed an algorithm for the portfolio construction from a large pool of stocks listed in a single market index SP CNX 500 using MATLAB code.

**2.1 Motivation**

From review of the earlier research done in this area, it is concluded that the portfolio optimization was not done on an exhaustive genetic scale. An application of a detailed genetic evolutionary algorithm, for constructing an optimum portfolio of stocks selected from a large pool of stocks, is demonstrated in this study. A priority index function is designed to select the stocks on the basis of company fundamentals, and then genetic algorithm is used to optimize the weights of the selected stocks. We also test the effectiveness of the genetic algorithm for portfolio construction. Steps have been taken to address the practical issues faced during the detailed implementation of genetic algorithm. An effective MATLAB code has been developed for this purpose. The motivation of the work was to apply genetic programming in the field of portfolio optimization for a large number of stocks. The paper has created a novel multi stage solution by creating and applying an algorithm for solution of portfolio optimization problem by genetic algorithm for the first time.

**3 Portfolio construction: algorithm**

The two stage decision based algorithm to construct the optimum portfolio is explained in this section. The section consists of two parts: the explanation of the priority index in Sect. 3.1 and the Optimization of selected stocks as explained in Sect. 3.2. The MATLAB code for the entire process of portfolio construction, as described below, is given in the appendix of this paper.

**3.1 Stock selection procedure**

The stocks have been selected on the basis of a lot of factors which have been explained in this section:

### 3.1.1 Diversification of portfolio

The most vital logic behind our portfolio selection is the inclusion of Diversity in our portfolio. We have achieved this by dividing the pool of stocks into sectors namely IT, Telecom, Automobile et al. Now the stocks will be evaluated on the parameters mentioned below using the criteria of maximum and minimum of that sector. This implies that stocks will get selected if their performance score is high in that sector, and thus only best performers from each sector would comprise of our portfolio resulting in a diversified portfolio. The parameters for selection of stocks into the portfolio are specified below:

#### 3.1.2 Parameters for selection

The fundamental factors are those which shall allow the user to construct a portfolio based on business performance principle rather than focusing merely on market sentiments. The following factors are essential for determining the stocks in the final portfolio:

#### 3.1.3 Price/earnings (P/E)

The P/E ratio is an important parameter for understanding the earnings per money invested. A P/E ratio of  $x$  shall imply an investment of  $x$  units of money for unity profit. Generally, an investor shall prefer to choose stocks which have a lower P/E ratio.

#### 3.1.4 Earnings/share (EPS ratio)

Earnings/Share is defined as the portion of company's profit allotted to each outstanding share of common stock. From an investor's perspective, a higher EPS is desirable.

#### 3.1.5 Wealth creation

Wealth creation is defined as the difference between return on invested capital and the weighted cost of average capital. The investor shall choose stocks of those companies having a higher Wealth creation for investing into stocks.

#### 3.1.6 Undervaluation

Undervaluation is defined as the situation when the stocks of the company are priced such that the market price is lower than the fair price. An investor shall always look to 'pick up' undervalued stocks. This is measured by the market capitalization to revenue ratio. If the value of this ratio is less than 1, the stock is considered to be undervalued.

### 3.1.7 (Price is to earning)/growth (PEG ratio)

We have used this parameter to represent a significant comparison between companies having different price/earnings ratios and growth percentages.

An investor shall always want to invest into businesses having lower PEG values.

The stocks are selected on the basis of their performance in all these parameters on the basis of historical data. The priority function is such designed that stocks having highest score shall be selected into the pool of stocks comprising the portfolio.

#### 3.1.8 Calculation of priority index

The priority index function uses the factors mentioned above as the benchmark, and the stocks are ranked on the basis of their cumulative scores as generated by the priority index function. Equities are ranked and allotted a score as per linear interpolation which can be expressed as follows for values whose maximum is desired:

$$S_{ij} = 100(X_{ij} - \text{Min}) / (\text{Max} - \text{Min})$$

where  $S_{ij}$  is the score of  $i$ th stock on  $j$ th parameter,  $X_{ij}$  is the functional value of  $i$ th stock on  $j$ th function,  $\text{Min}$  is the minimum value of  $i$ th stock on  $j$ th function,  $\text{Max}$  is the maximum value of  $i$ th stock on  $j$ th function

The formula for values whose minimum is desired is:

$$S_{ij} = 100(X_{ij} - \text{Max}) / (\text{Min} - \text{Max})$$

The priority index ( $PI_i$ ) is calculated by the summation of all scores each stock has earned on all the parameters.

$$PI_i = \sum S_{ij}$$

The stock selection is based on the  $PI_i$  and the algorithm shall create the portfolio from only those stocks which have a Priority Index greater than 3.8 on a 5 point scale.

## 3.2 Portfolio optimization

The stocks selected on the basis of  $PI_i$  are added to the portfolio and their weights are operated genetically for optimization. The selected stocks comprise of the genes of a chromosome. Each chromosome has genes represented by stocks which are further allotted random values (Fig. 4). These weights are calculated by the formula  $(v_i / \sum v_i)$  which represent the proportion in which the stock  $i$  is to be invested in. An initial generation of ten such chromosomes is created by the random function. The next step is the creation of the reproduction pool which can be understood by the explanation given in the earlier section. A tabular representation is shown in Table 1.



**Fig. 4** A chromosome with gene representation

Generation Pool



Reproduction Pool



**Fig. 5** Example to show reproduction pool

The reproduction pool shall now consist of the chromosomes: CH1, CH2, CH3, and CH3 as shown in Fig. 5.

After the selection of the reproduction pool, the chromosomes are operated upon by the crossover and mutation operators. This gives rise to the second generation such that each generation has the same number of chromosomes which have been modified genetically. For example, after running 100 generations of ten chromosomes each, we get a population size of 1,000 chromosomes. The chromosome with the best fitness value is selected and the composition shall be the solution to our portfolio optimization problem.

#### 4 Applied example of genetic algorithm on SP 500 Index: US

In this section we will illustrate the applied example of the multi stage portfolio construction algorithm on the SP500 US Index. The algorithm is illustrated in two steps: the

input required from the user and the output generated. We shall analyze the output in Sect. 5.

##### 4.1 Input

The input shall consist of the stocks from SP 500 Index: US comprising of the daily closing prices, EPS ratios, PEG values, weighted average cost of capital, market capitalization/revenue and return on invested capital in the US market from the duration December 2011–December 2012. The input was taken from Bloomberg. The data should be divided sector wise as shown in the column ‘Sectors’ of the Table 2. A sample input data file obtained from BLOOMBERG is shown in Table 2.

And the data set shall continue in the same order for all the input stocks (SP500 INDEX: US in our study).

A second input file is a column of the daily market returns for the same duration for the SP 500 Index: US. It is taken as the return of the market portfolio for the calculation of raw Beta of each stock of the index. Adjusted Beta is calculated by using the following relation:

$$\text{Adjusted beta} = 0.67 * \text{raw beta} + 0.33 * 1$$

A third input file (Table 3) containing the daily returns of all the stocks listed in the single market index (SP500 INDEX: US) for the period December 2011–December 2012 is shown below:

And the data set shall continue in the same order for all the input stocks (SP500 INDEX: US in our study).

##### 4.2 Output

The portfolio constructed for a threshold of 3.8 on a 5 point scale is shown in the Table 4 below along with the performance analytics. This threshold value has given us a portfolio of 25 stocks selected by the priority index function and optimized by genetic algorithm. The lower the threshold, the higher will be the number of stocks in the portfolio and vice versa.

## 5 Analysis of portfolio

We shall now analyze the constructed portfolio on the basis of the following parameters (Table 5):

##### 5.1 Average annual return

The average annual return is calculated from the average daily return as per the following formula:



**Table 2** Input data file for MATLAB code as obtained from BLOOMBERG

| Stocks         | Sectors | EPS    | PEG   | ROI   | WACC equity | P/E    | WACC   | Market cap/revenue |
|----------------|---------|--------|-------|-------|-------------|--------|--------|--------------------|
| AAPL US Equity | 7       | 59.14  | 0.50  | 39.41 | 10.48       | 11.62  | 28.93  | 3.10               |
| ABC US Equity  | 5       | 9.80   | 1.15  | 14.04 | 7.18        | 15.08  | 6.86   | 0.13               |
| ABT US Equity  | 5       | 1.68   | 1.28  | 15.31 | 6.53        | 12.99  | 8.78   | 2.61               |
| ACE US Equity  | 4       | -48.85 | 1.10  | 4.11  | 8.10        | 9.79   | -3.99  | 1.53               |
| ACN US Equity  | 7       | 12.46  | 1.28  | 81.44 | 11.48       | 16.83  | 69.96  | 1.54               |
| ADBE US Equity | 7       | 0.60   | 2.63  | 10.99 | 9.84        | 22.91  | 1.15   | 4.19               |
| ADM US Equity  | 2       | -41.96 | 1.15  | 5.25  | 7.45        | 13.60  | -2.20  | 0.20               |
| ADP US Equity  | 7       | 12.20  | 2.03  | 3.97  | 9.77        | 20.76  | -5.81  | 2.56               |
| ADSK US Equity | 7       | 34.41  | 1.04  | 17.47 | 15.48       | 30.83  | 1.99   | 3.43               |
| AEE US Equity  | 10      | 270.69 | -3.15 | 4.64  | 4.75        | 12.69  | -0.11  | 1.08               |
| AEP US Equity  | 10      | 28.46  | 3.13  | 4.88  | 4.40        | 14.31  | 0.47   | 1.40               |
| AET US Equity  | 5       | 25.41  | 0.86  | 15.52 | 8.63        | 9.02   | 6.89   | 0.44               |
| AIZ US Equity  | 4       | 124.21 | 0.64  | 3.95  | 7.22        | 5.15   | -3.27  | 0.32               |
| ALL US Equity  | 4       | -12.21 | 1.21  | -3.32 | 7.69        | 7.70   | -11.01 | 0.59               |
| ALTR US Equity | 7       | -6.27  | 2.17  | 23.07 | 12.86       | 19.22  | 10.21  | 6.11               |
| AMAT US Equity | 7       | -93.84 | 2.34  | 3.63  | 11.35       | 15.19  | -7.72  | 1.56               |
| AMGN US Equity | 5       | -15.56 | 1.45  | 10.31 | 5.96        | 14.80  | 4.35   | 3.93               |
| AMZN US Equity | 1       | -46.12 | 4.31  | 6.33  | 12.00       | 469.12 | -5.68  | 1.96               |

$$AAR = (1 + ADR)^{252} - 1$$

This shall give us the annual average return for the portfolio allowing the investor to make a decision. The AAR of the portfolio is 26.01 % on the basis of the historical data (Dec’ 2011–Dec’ 2012).The constructed portfolio beats the market return by a significant margin.

### 5.2 Beta of portfolio ( $\beta_p$ )

The Beta of portfolio shall be an indicator of the correlated volatility of the asset in relation with the volatility of the index on which the stocks are benchmarked. The Beta of portfolio is 0.87 which is less than Beta of the market portfolio.

### 5.3 Treynor’s ratio

The return which would be earned in excess as compared to a risk free environment is referred to as the Treynor’s ratio. This shall be analyzed for the above portfolio. The mathematical expression for Treynor’s ratio is given as:

$$\text{Treynor’s ratio} = \text{excess return}/\text{beta}$$

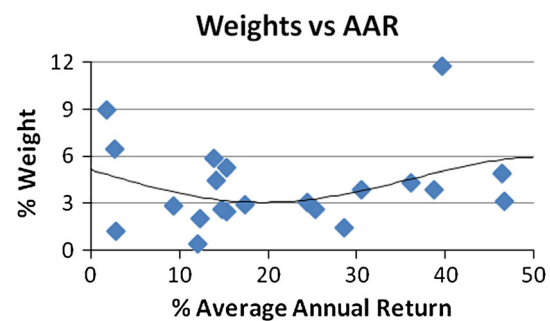
The Treynor’s ratio is 0.2976. This means that the portfolio provides a 29.76 % return per unit of risk. The Beta of 0.87 states that it is only 87 % as volatile as that of the SP 500

thus establishing that not only does the portfolio has maximized return but it has also managed the risk of the portfolio.

### 5.4 Jensen’s alpha

It is used as an indicator to determine the unusual return on the portfolio as compared to the return on the market indicating the effectiveness of the selection and optimization of the portfolio algorithm. Jensen’s alpha of the portfolio constructed is 14.56 % which indicates that the portfolio constructed by the algorithm is very effective.

$$\text{Jensen’s alpha} = (R_p - R_f) - \beta_p(R_m - R_f)$$



**Fig. 6** Plot of weights of selected stocks against AAR

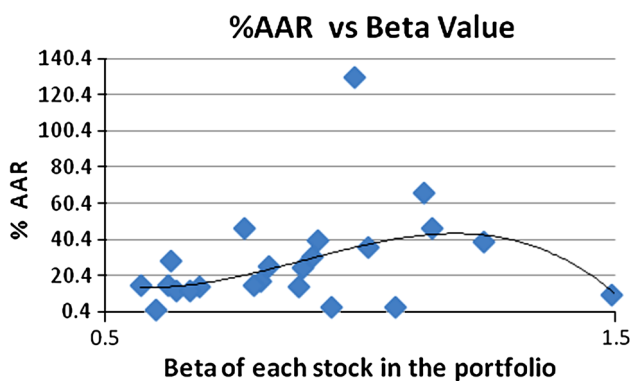


Fig. 7 Plot of AAR against Beta Value of each selected stock

Risk free return in the US market is taken as the yield on a US Treasury Bill for 1 year = 0.13 %.

5.5 Graphical analysis

5.5.1 Graph 1: the relation between stock weights assigned and %AAR (Fig. 6)

The stock weights assigned show an interesting trend, the genetic algorithm allotted weights which were initially decreasing with %AAR and then began increasing after a certain amount. This technique gives insight to the method of optimization of genetic algorithm.

5.5.2 Inference

The reader can infer that the algorithm developed is a multi decision multi stage algorithm where the weights assigned are multi function of return and risk. Hence we do not see a direct correlation between weights and AAR.

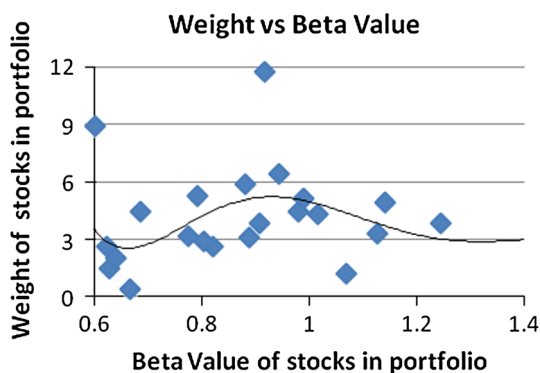


Fig. 8 Plot of weight of selected stock against Beta Value

5.5.3 Graph 2: variation of %AAR with Beta value (Fig. 7)

The annual return shows an increasing trend with Beta initially and then begins to decrease. The stocks have been chosen at such a manner that we get values with higher AAR and lower Beta values.

5.5.4 Inference

The reader can infer that the algorithm developed is a multi decision multi stage algorithm where the weights assigned are multi function of return and risk. Hence we do not see a direct correlation between weights and

5.5.5 Graph 3: weights versus beta (Fig. 8)

The weights increase with Beta value for lower Beta values but then begin to decrease with higher Beta values. This shows the optimization technique of the genetic algorithm.

5.5.6 Inference

It can infer that the returns are low for stocks with a low Beta value. Equities with higher Beta values have higher returns indicating that higher the risk, higher the return. However, the returns decrease for very high Beta values. Genetic algorithm optimizes these variations to yield the most optimum portfolio.

6 Performance of the portfolio over a 6 month holding period

We have analyzed the constructed portfolio on the basis of live data also. Please note that the portfolio was constructed using historical data (December 2011–December 2012). We analyze the constructed portfolio by using the futuristic data (January’13–June’13). The analysis shows that the portfolio has performed remarkably with an %AAR as high as 15.98 % which was calculated from the 6 month yield rate of 7.69 %. During this period market shows a %AAR if 10.14 %, hence the portfolio constructed above using genetic algorithm beats the market. This is presented in Table 6.

**Table 3** Input data file for MATLAB code as obtained from BLOOMBERG for daily returns of all the stocks for the period December 2011 to December 2012

| Stock   | Daily return |            |            |            |            |            |            |
|---------|--------------|------------|------------|------------|------------|------------|------------|
|         | 02-12-2011   | 05-12-2011 | 06-12-2011 | 07-12-2011 | 08-12-2011 | 09-12-2011 | 12-12-2011 |
| AAPL UW | 0.46         | 0.85       | -0.52      | -0.48      | 0.4        | 0.76       | -0.45      |
| ABC UN  | -0.67        | -1.08      | -0.46      | 0.41       | -0.85      | 0.25       | -1.46      |
| ABT UN  | -0.68        | 0.3        | 0.42       | 0.29       | -0.95      | 0.72       | -0.55      |
| ACE UN  | -0.32        | 1.18       | 1.52       | 0.07       | -2.29      | 1          | -1.38      |
| ACN UN  | 0.71         | 1.61       | -0.25      | -1.06      | -3.19      | 3.6        | -0.83      |
| ADBE UW | -0.11        | 2.95       | 0.18       | -0.36      | -1.72      | 2.41       | -2.32      |
| ADM UN  | -0.76        | -0.1       | -1.2       | -0.17      | -2.74      | 1.32       | -2.68      |
| ADP UW  | -0.39        | 1.54       | 0.1        | 0.82       | -1.05      | 1.67       | -1.35      |
| ADSK UW | -0.88        | 0.35       | 2.02       | 0.06       | -2.47      | 1.33       | -2.76      |
| AEE UN  | -3.65        | -0.8       | 1.57       | -0.59      | -1.12      | 1.79       | -1.36      |
| AEP UN  | -0.93        | 0.56       | 0.68       | 0          | -1.13      | 1.12       | -1.03      |
| AET UN  | -1.61        | -0.39      | 1.22       | 0.87       | -3.5       | 1.64       | -2.18      |
| AIZ UN  | 0.1          | 1.36       | 0.76       | 1.4        | -3.48      | 2.25       | -2.38      |
| ALL UN  | 0.45         | 1.24       | 0.41       | 0.55       | -2.53      | 1.32       | -2.23      |
| ALTR UW | -1.07        | 1.24       | -2.05      | -1.28      | -2.39      | 1.1        | -0.86      |
| AMAT UW | 0.56         | 1.48       | 0.91       | 1.44       | -2.93      | 2.56       | -6.07      |
| AMGN UW | 0.17         | -1.03      | 0.68       | 0.78       | 0.12       | 0.31       | -2.44      |
| AMZN UW | -0.56        | 0.11       | -2.17      | 1.73       | -2.48      | 1.34       | -1.82      |

**Table 4** Output file showing weights of selected stocks in the optimum portfolio

| Equity           | % Annual return | Weights (%) |
|------------------|-----------------|-------------|
| 'ADBE US equity' | 6.435194        | 5.82        |
| 'AKAM US equity' | 3.14256         | 0.56        |
| 'AMZN US equity' | 2.841326        | 0.09        |
| 'AVB US equity'  | 1.456976        | 0.62        |
| 'BEAM US equity' | 11.7248         | 1.85        |
| 'BXP US equity'  | 1.237292        | 6.57        |
| 'CCL US equity'  | 4.894944        | 1.29        |
| 'CTL US equity'  | 4.456662        | 5.17        |
| 'EFX US equity'  | 3.865197        | 9.52        |
| 'EXPE US equity' | 5.850266        | 6.92        |
| 'GAS US equity'  | 5.276695        | 0.85        |
| 'HCN US equity'  | 3.299015        | 6.55        |
| 'JNPR US equity' | 3.874859        | 9.13        |
| 'KIM US equity'  | 3.078482        | 0.21        |
| 'KO US equity'   | 2.502491        | 1.63        |
| 'LRCX US equity' | 2.624299        | 8.92        |
| 'NBL US equity'  | 0.398242        | 1.62        |
| 'NTRS US equity' | 2.002546        | 4.75        |
| 'PKI US equity'  | 8.936224        | 0.06        |
| 'PNR US equity'  | 4.292124        | 4.11        |
| 'PXD US equity'  | 4.428024        | 5.09        |
| 'RHT US equity'  | 2.723802        | 8.59        |
| 'RRC US equity'  | 2.623591        | 3.37        |

**Table 4** continued

| Equity           | % Annual return | Weights (%) |
|------------------|-----------------|-------------|
| 'SRCL US equity' | 5.10981         | 3.56        |
| 'URBN US equity' | 2.924574        | 3.14        |

## 6.1 Performance parameters

The success of the proposed solution is gauged by comparison of the portfolio performance against the benchmark. The paper has analyzed the performance by gauging how strongly the portfolio performance against the benchmark of SP500 US. The performance on a 6 month holding period is analyzed as follows:

## 6.2 Average annual return

The analysis of the portfolio shows an %AAR of 26.01 % from the historical data (December' 2011 to December' 2012). Now when tested for a 6 month holding period (Jan' 13–June' 13), the portfolio performed reasonably well giving an %AAR of 15.98 % whereas the market return for the same period was 10.14 %. Thus the above two stage technique involving genetic algorithm is highly objective, useful and effective for portfolio construction and optimization.

**Table 5** Performance parameters of the optimum portfolio constructed for the period (Dec' 2011–Dec' 2012)

| Performance parameters             |         |
|------------------------------------|---------|
| Priority index threshold           | 3.5/5   |
| No of stocks selected              | 25      |
| Average Annual return of portfolio | 26.01 % |
| Beta of portfolio                  | 0.87    |
| Treynor's ratio                    | 0.2976  |
| Jensen's alpha                     | 14.56 % |

**Table 6** The performance of portfolio on 6 month holding period

| Name of equity | % Return as per 6 month holding period |
|----------------|--|
| ADBE UW equity | 20.91333188                            |
| AKAM UW equity | 4.008895325                            |
| AMZN UW equity | 10.57193525                            |
| AVB UN equity  | -0.501705071                           |
| BEAM UN equity | 3.306263904                            |
| BXP UN equity  | -0.321246069                           |
| CCL UN equity  | -6.744575008                           |
| CTL UN equity  | -9.636814951                           |
| EFX UN equity  | 8.887625119                            |
| EXPE UW equity | -2.115684313                           |
| GAS UN equity  | 7.230467441                            |
| HCN UN equity  | 9.365386173                            |
| JNPR UN equity | -1.830422237                           |
| KIM UN equity  | 10.92169129                            |
| KO UN equity   | 10.64765461                            |
| LRCX UW equity | 22.72229139                            |
| NBL UN equity  | 18.0266024                             |
| NTRS UW equity | 15.43065818                            |
| PKI UN equity  | 2.393997109                            |
| PNR UN equity  | 17.37519993                            |
| PXD UN equity  | 35.80015595                            |
| RHT UN equity  | -9.705376896                           |
| RRC UN equity  | 23.06221326                            |
| SRCL UW equity | 18.3978598                             |
| URBN UW equity | 2.185068441                            |

### 6.3 Beta of portfolio ( $\beta_p$ )

The Beta of portfolio shall be an indicator of the correlated volatility of the asset in relation with the volatility of the

index on which the stocks are benchmarked. The Beta of portfolio is 0.87 which is less than beta of the market portfolio.

### 6.4 Treynor's ratio

The Treynor's ratio is 0.1819. This means that the portfolio provides a 18.19 % return per unit of risk. The Beta of 0.87 states that it is only 87 % as volatile as that of the SP 500 thus establishing that not only does the portfolio has maximized return but it has also managed the risk of the portfolio.

## 7 Conclusion

The study has shown the wide implications of the above two stage process used in portfolio construction:

Selection of stocks on the basis of business fundamentals rather than market sentiments is an important lesson for acquiring long term benefits at lower risks. Our portfolio is based on creating a genetically suitable return and risk at the same time. We have successfully reduced the risk by a selection procedure based upon company fundamentals. Thus the portfolio is said to be optimum giving considerable return by taking a calculated risk.

The priority index allows users to select the threshold above which the stocks shall enter the portfolio. Higher is the threshold, fewer are the stocks and vice versa. This indicates that as the threshold goes up, the decision making becomes stern and we chose stocks which are stronger in business fundamentals. This might compromise on the return but shall give us long term stability.

The research has aptly demonstrated the application of genetic algorithm for optimization of portfolio. This is a novel example of application of genetic algorithms in the field of portfolio optimization in finance.

The portfolio for a threshold of 3.8 on a five point scale gives an annual average return of 15.98 % with respect to a risk free return of 0.13 %.

## Appendix

The MATLAB code for the procedure adopted in the optimum portfolio construction is given below:

The MATLAB code for the procedure adopted in the optimum portfolio construction is given below:

```

%DEFINITION AND NAMES OF VARIABLES
USED

%ndata= numerical data related to stock (EPS, PE,
PEG, Wealth Creation, MC/RE)
%text= name of all stocks
%value= daily adjusted closing price values of all
the stocks
%ntms= no of chromosomes to be generated
%l= number of stocks
%m= number of days of which data is available
%value1= daily return of market index(in
percentage)
%rtm= daily return of stocks
%marketrtm= daily return on market
%k= array used for priority function (i.e. to sort
and select stocks)
%z= array used for counting stocks in each sector
%stk= sorted out stocks by priority function
(portfolio)
%stkn= number of stocks selected by priority
function (portfolio)
%stkrn= return of each stock selected by priority
function
%rbs= average daily return of stocks
%rb1= average annual return of stocks
%rb= arithmetic mean of return
%cv= covariance of return of selected stock
%cvr= covariance of return and market return of
each stock
%va= variance of market return
%fitness= fitness value of each chromosome
%ro= return of each stock in portfolio
%rtr1= total return of portfolio
%ann= annual return of the portfolio
%betaf= beta of individual stocks
%bta= beta of the portfolio
%rpool= reproduction pool containing chromosome
details (weight,value,random value, selected
chromosome for crossover or mutation)
%chrv= random value for stocks in a chromosome
%chrw= weight of the stocks in a chromosome
%rc= counter used for number of chromosomes
%treynors= treynors ratio of portfolio
%alpha= alpha of the portfolio
%annmtrn= annual market return
%prfn= threshold value of the Priority Function

%Main Program which is needed to run [Save as:
Main.m]

clc;
close all;
clear all;

prfn=3.8;
ntms=[500 , 1000 , 1500 , 2000 ,2500];
[la,lb]=size(ntms);
for ntimes=1:lb
[ndata, text, alldata] =
xlsread('Stock_details.xlsx');
[l,n]=size(ndata);

%data=transpose(ndata);
v=n;
n=n+1;

%adding row number
for i=1:l
ndata(i,n)=i;
end

%count of all stocks in each sector
for i=1:10
x=1;
for j=1:l
if(ndata(j,1)==i)
z(i)=x;
x=x+1;
end
end
end

%PE
k=sortrows(ndata,6);
for y=1:10
j(y)=1;
end
n=n+1;
[k,j,z]= sortp(k,l,z,j,n);

%PEG
k=sortrows(k,3);
for y=1:10
j(y)=1;
end
n=n+1;
[k,j,z]= sortp(k,l,z,j,n);

%EPS
k=sortrows(k,2);
for y=1:10
j(y)=z(y);
end
n=n+1;
[k,j,z]= sortm(k,l,z,j,n);

%Wealth Creation
k=sortrows(k,7);

```

```

for y=1:10
    j(y)=z(y);
end
n=n+1;
[k,j,z]= sortm(k,l,z,j,n);

%MC/RE
k=sortrows(k,8);
for y=1:10
    j(y)=1;
end
n=n+1;
[k,j,z]= sortp(k,l,z,j,n);

n=n+1;
for i=1:l
    k(i,n)=0;
    for j=v+2:n-1
        k(i,n)=k(i,n)+k(i,j);
    end
end
k=sortrows(k,n);

f=0;
for i=1:l
    if k(i,n)>=prfn;
        f=f+1;
        for j=1:n
            stk(f,j)=k(i,j);
        end
    end
end

k=sortrows(k,1);
stk=sortrows(stk,9);
[stkn,stk]=size(stk);

[value,name] = xlsread('Stock_return.xlsx');
[nn,m]=size(value);
for j=1:nn
    rtrn(j,1)=j;
end
for i=1:m
    rtrn(:,i+1)=value(:,i)/100;
    rtrn1(:,i)=rtrn(:,i+1);
end

[value1] = xlsread('Market_return.xlsx');
value1=transpose(value1);
for i=1:(m)
    marketrtrn(i)=value1(i)/100;
end

%no of stock screened
y=1;
for i=1:stkn
    for j=1:nn
        if stk(i,9)==rtrn(j,1);
            txt(y)=text(j);
            for x=1:m+1
                stkr(y,x)=rtrn(j,x);
                if x>1;
                    stkrn(y,x-1)=rtrn(j,x);
                end
            end
            y=y+1;
        end
    end
end

for i=1:stkn
    stddv(i)=std(stkrn(i,:));
    rbs(i)=1;
    for j=1:m
        rbs(i)=rbs(i)*(1+stkrn(i,j));
    end
    rb1(i)=(rbs(i)^(1/252))-1;
    rb(i)=mean(stkrn(i,:));
end

for i=1:stkn
    for j=1:stkn
        cv(i,j)=0;
        for d=1:252
            temp=(stkrn(i,d)-rb(i))*(stkrn(j,d)-rb(j));
            cv(i,j)=cv(i,j)+temp;
        end
        cv(i,j)=cv(i,j)/252;
    end
end

rbm=mean(marketrtrn);
va=var(marketrtrn);
for i=1:stkn
    cvr(i)=0;
    for d=1:252
        tmp4=(stkrn(i,d)-rb(i)-0.0013)*(marketrtrn(d)-rbm-0.0013);
        cvr(i)=cvr(i)+tmp4;
    end
    cvr(i)=cvr(i)/252;
    bt1(i)=0.67*(cvr(i)/va)+0.33;
end

rc=1;
[rpool,chr,chrw,fitness,rc]
=rpoola(stkn,rb1,cv,rc,stk);
for i=1:ntms(ntimes)
    [rpool,chr,chrw,fitness,rc]
=rpoolb(rpool,chr,chrw,fitness,stk,rb1,cv,rc,stk);
end

[mxxx,ind]=max(fitness);
[mx2,ind1]=max(mxxx);
index=((ind(ind1)-1)*10)+ind1;
rtr1(ntimes)=0;
if index >10

```

```

ind3=index-mod(index,10);
rc=ind3/10;
ind4=mod(index,10);
if ind4==0
    ind4=10;
end
else
    rc=1;
    ind4=index;
end

plt(ntimes,:)=chrw(rc,ind4,:);
bta(ntimes)=0;
for i=1:stkn
    ro(ntimes,i)=rb1(i)*chrw(rc,ind4,i);

rtr1(ntimes)=rtr1(ntimes)+rb1(i)*chrw(rc,ind4,i);
    betaf(ntimes,i)=bt1(i)*chrw(rc,ind4,i);
    bta(ntimes)=bta(ntimes)+betaf(ntimes,i);
end
rtr1(ntimes);
bta(ntimes);
ann(ntimes)=((1+rtr1(ntimes)).^252-1)
treynors(ntimes)=(ann(ntimes))/bta(ntimes);
annmtrn=(1+rbm)^252-1;
alpha(ntimes)=(ann(mtrn)-0.0013)-
bta(ntimes)*(annmtrn-0.0013);
end

[v1,ind]=max(ann);
xlswrite('output.xlsx',txt(1:stkn),'A1:A10000');
xlswrite('output.xlsx',bt1(1:stkn),'B1:B10000');
xlswrite('output.xlsx',plt(ind,:),'C1:C10000');
xlswrite('output.xlsx',rb1(1:stkn),'D1:D10000');
xlswrite('output.xlsx',alpha(1:lb),'E1:E10000');
xlswrite('output.xlsx',annmtrn,'F1');
xlswrite('output.xlsx',ann(ind),'G1');

%sortp function to allot priority grades in
descending order as per the higher value [Save as:
sortp.m]
function [k,j,z]= sortp(k,l,z,j,n)
for i=1:l
    if k(i,1)==1 ;
        k(i,n)=j(1)/z(1);
        j(1)=j(1)-1;

    elseif k(i,1)==2 ;
        k(i,n)=j(2)/z(2);
        j(2)=j(2)-1;

    elseif k(i,1)==3 ;
        k(i,n)=j(3)/z(3);
        j(3)=j(3)-1;

    elseif k(i,1)==4 ;
        k(i,n)=j(4)/z(4);
        j(4)=j(4)-1;

    elseif k(i,1)==5 ;
        k(i,n)=j(5)/z(5);
        j(5)=j(5)-1;

    elseif k(i,1)==6 ;
        k(i,n)=j(6)/z(6);
        j(6)=j(6)-1;

    elseif k(i,1)==7 ;
        k(i,n)=j(7)/z(7);
        j(7)=j(7)-1;

    elseif k(i,1)==8 ;
        k(i,n)=j(8)/z(8);
        j(8)=j(8)-1;

    elseif k(i,1)==9 ;
        k(i,n)=j(9)/z(9);
        j(9)=j(9)-1;

    else k(i,1)=10 ;
        k(i,n)=j(10)/z(10);
        j(10)=j(10)-1;
    end
end

%sortm function to allot priority grades in
ascending order as per the higher value [Save as:
sortm.m]
function [k,j,z]= sortm(k,l,z,j,n)
for i=1:l
    if k(i,1)==1 ;
        k(i,n)=j(1)/z(1);
        j(1)=j(1)+1;

    elseif k(i,1)==2 ;
        k(i,n)=j(2)/z(2);
        j(2)=j(2)+1;

    elseif k(i,1)==3 ;
        k(i,n)=j(3)/z(3);
        j(3)=j(3)+1;

    elseif k(i,1)==4 ;
        k(i,n)=j(4)/z(4);
        j(4)=j(4)+1;

    elseif k(i,1)==5 ;
        k(i,n)=j(5)/z(5);
        j(5)=j(5)+1;

    elseif k(i,1)==6 ;
        k(i,n)=j(6)/z(6);
        j(6)=j(6)+1;

    elseif k(i,1)==7 ;

```

```

k(i,n)=j(7)/z(7);
j(7)=j(7)+1;

elseif k(i,1)==8 ;
k(i,n)=j(8)/z(8);
j(8)=j(8)+1;

elseif k(i,1)==9 ;
k(i,n)=j(9)/z(9);
j(9)=j(9)+1;

else k(i,1)=10 ;
k(i,n)=j(10)/z(10);
j(10)=j(10)+1;
end
end

%rpoola function to generate the first reproduction
pool [Save as: rpoola.m]

function[rpool,chrV,chrW,fitness,rc]
=rpoola(stkn,rb1,cv,rc,stkcn)

for i=1:10
ss=0;
for j=1:stkn
chrV(rc,i,j)=randn();
if chrV(rc,i,j)<0;
chrV(rc,i,j)=chrV(rc,i,j)*(-1);
end
ss=ss+chrV(rc,i,j);
end

for j=1:stkn
chrW(rc,i,j)=(chrV(rc,i,j)/ss);
end
end

for y=1:10
sumchr(y)=0;
newv(y)=0;
for i=1:stkn
val=var(stkcn(i));
for j=1:stkn

sumchr(y)=((sumchr(y)+2*cv(i,j)*chrW(rc,y,i)*chr
w(rc,y,j)));
newv(y)=newv(y)+rb1(j)*chrW(rc,y,j);
end
sumchr(y)=sumchr(y)+chrW(rc,y,i).^2*val;
end
sumchr(y)=sqrt(sumchr(y));
fitness(rc,y)=newv(y)/sumchr(y);
end

k=y;
sum1=0;
for i=1:10
sum1=sum1+fitness(rc,i);
end
sfit(rc)=sum1;

pl=1;
rrp=0;
for i=1:10
rpool(rc,i,pl)=i;
rpool(rc,i,pl+1)=fitness(rc,i);
rpool(rc,i,pl+2)=rpool(rc,i,pl+1)/sfit(rc);
rrp=rrp+rpool(rc,i,pl+2);
rpool(rc,i,pl+3)=rrp;
rpool(rc,i,pl+4)=rand();
end

for i=1:10
j=1;
mnx=1;
while(rpool(rc,j,pl+3)<rpool(rc,i,pl+4))
mnx=j;
j=j+1;
end
rpool(rc,i,pl+5)=mnx;
end

for i=1:10
countp=histc(rpool(rc,.,6),i);
rpool(rc,i,pl+6)=countp;
if countp>0
rpool(rc,i,pl+7)=rpool(rc,i,pl+1);
else
rpool(rc,i,pl+7)=0;
end
end
k=0;

[chrV,fitness,k,stkcn,chrW,sumchr,newv,rb1,cv,rpool
,pl,rc,yyy] =
crossb(chrV,fitness,k,stkcn,chrW,sumchr,newv,rb1,c
v,rpool,pl,rc);
if yyy>0
[chrV,fitness,k,stkcn,chrW,sumchr,newv,rb1,cv,rpool
,pl,rc,yyy] =
crossb(chrV,fitness,k,stkcn,chrW,sumchr,newv,rb1,c
v,rpool,pl,rc);
end
if yyy>0
[chrV,fitness,k,stkcn,chrW,sumchr,newv,rb1,cv,rpool
,pl,rc,yyy] =
crossb(chrV,fitness,k,stkcn,chrW,sumchr,newv,rb1,c
v,rpool,pl,rc);
end

for i=1:10
while rpool(rc,i,8)>0

[chrV,fitness,k,stkcn,chrW,sumchr,newv,rb1,cv,rpool

```



```
,pl,rc) =
mutate(chrv,fitness,k,stkn,chrw,sumchr,neww,rb1,c
v,rpool,pl,rc);
end
end
rc=rc+1;
```

```
%crossb function for the crossover between two
chromosomes [Save as: crossb.m]
```

```
function
[chrw,fitness,k,stkn,chrw,sumchr,neww,rb1,cv,rpool
,pl,rc,yyy] =
crossb(chrv,fitness,k,stkn,chrw,sumchr,neww,rb1,c
v,rpool,pl,rc)
```

```
largest = rpool(rc,1,8);
x=1;
secondLargest = 0;
y=0;
for j=2:10
number = rpool(rc,j,8);
if number > largest
secondLargest = largest;
largest = number;
y=x;
x=j;
else
if number > secondLargest
secondLargest = number;
y=j;
end
end
end
yyy=y;
if y>0
rpool(rc,x,7)=rpool(rc,x,7)-1;
rpool(rc,y,7)=rpool(rc,y,7)-1;
for i=1:10
if rpool(rc,i,7)<1
rpool(rc,i,pl+7)=0;
end
end
end
```

```
i=k+1;
aa=rand();
for j=1:stkn
chrw(rc+1,i,j)=(1-
aa)*chrw(rc,x,j)+(aa*chrw(rc,y,j));
chrw(rc+1,i+1,j)=(aa)*chrw(rc,x,j)+((1-
aa)*chrw(rc,y,j));
end
```

```
for i=k+1:k+2
ss=0;
for j=1:stkn
ss=ss+chrw(rc+1,i,j);
end
for j=1:stkn
```

```
chrw(rc+1,i,j)=(chrw(rc+1,i,j)/ss);
```

```
end
end
for y=k+1:k+2
sumchr(y)=0;
neww(y)=0;
for i=1:stkn
for j=1:stkn
```

```
sumchr(y)=(sumchr(y)+cv(i,j)*chrw(rc+1,y,i)*chr
w(rc+1,y,j));
neww(y)=neww(y)+rb1(j)*chrw(rc+1,y,j);
end
end
sumchr(y)=sqrt(sumchr(y));
fitness(rc+1,y)=neww(y)/sumchr(y);
```

```
end
k=k+2;
end
```

```
%mutate function for the mutation of chromosome
[Save as: mutate.m]
```

```
function
[chrw,fitness,k,stkn,chrw,sumchr,neww,rb1,cv,rpool
,pl,rc] =
mutate(chrv,fitness,k,stkn,chrw,sumchr,neww,rb1,c
v,rpool,pl,rc)
```

```
xx=0;
i=1;
while i<=10 && xx==0
if rpool(rc,i,7)>0
rpool(rc,i,7)=rpool(rc,i,7)-1;
if rpool(rc,i,7)<1
rpool(rc,i,pl+7)=0;
end
xx=i;
end
i=i+1;
end
```

```
i=k+1;
for j=1:stkn
chrw(rc+1,i,j)=chrw(rc,xx,j);
end
```

```
j1=(17/27)*stkn;
j2=1;
while j2<j1
j2=j2+1;
end
```

```
j3=(14/27)*stkn;
j4=1;
while j4<j3
j4=j4+1;
```

```

end

temp=chrv(2,i,j2+1);
for j=j4:j2
    temp1(j)=chrv(rc+1,i,j);
end
for j=j4+1:j2+1
    chrv(rc+1,i,j)=temp1(j-1);
end
chrv(rc+1,i,13)=temp;

ss=0;
for j=1:stkn
    ss=ss+chrv(rc+1,i,j);
end

for j=1:stkn
    chrv(rc+1,i,j)=(chrv(rc+1,i,j)/ss);
end

y=k+1;
sumchr(y)=0;
newv(y)=0;
for i=1:stkn
    for j=1:stkn

sumchr(y)=(sumchr(y)+cv(i,j)*chrv(rc+1,y,i)*chr
w(rc+1,y,j));
        newv(y)=newv(y)+rb1(j)*chrv(rc+1,y,j);
    end
end
sumchr(y)=sqrt(sumchr(y));
fitness(rc+1,y)=newv(y)/sumchr(y);
k=k+1;

%rpoolb function to generate the remaining number
of reproduction pools [Save as: rpoolb.m]

function[rpool,chrw,chrw,fitness,rc]
=rpoolb(rpool,chrw,chrw,fitness,stkn,rb1,cv,rc,stkcn
)

for i=1:10
    ss=0;
    for j=1:stkn
        chrv(rc,i,j)=randn();
        if chrv(rc,i,j)<0;
            chrv(rc,i,j)=chrv(rc,i,j)*(-1);
        end
        ss=ss+chrv(rc,i,j);
    end

    for j=1:stkn
        chrv(rc,i,j)=(chrv(rc,i,j)/ss);
    end

    end

    for y=1:10
        sumchr(y)=0;
        newv(y)=0;
        for i=1:stkn
            val=var(stkcn(i));
            for j=1:stkn

sumchr(y)=(sumchr(y)+2*cv(i,j)*chrv(rc,y,i)*chr
w(rc,y,j));
                newv(y)=newv(y)+rb1(j)*chrv(rc,y,j);
            end
            sumchr(y)=sumchr(y)+chrv(rc,y,i).^2*val;
        end
        sumchr(y)=sqrt(sumchr(y));
        fitness(rc,y)=newv(y)/sumchr(y);
    end

    k=y;
    sum1=0;
    for i=1:10
        sum1=sum1+fitness(rc,i);
    end
    sfit(rc)=sum1;
    pl=1;
    rrp=0;
    for i=1:10
        rpool(rc,i,pl)=i;
        rpool(rc,i,pl+1)=fitness(rc,i);
        rpool(rc,i,pl+2)=rpool(rc,i,pl+1)/sfit(rc);
        rrp=rrp+rpool(rc,i,pl+2);
        rpool(rc,i,pl+3)=rrp;
        rpool(rc,i,pl+4)=rand();
    end

    for i=1:10
        j=1;
        mx=1;
        while(rpool(rc,j,pl+3)<rpool(rc,i,pl+4))
            mx=j;
            j=j+1;
        end
        rpool(rc,i,pl+5)=mx;
    end

    for i=1:10
        countrp=histc(rpool(rc,:,6),i);
        rpool(rc,i,pl+6)=countrp;
        if countrp>0
            rpool(rc,i,pl+7)=rpool(rc,i,pl+1);
        else
            rpool(rc,i,pl+7)=0;
        end
    end
    k=0;

```

```

[chr,fitness,k,stk,chrw,sumchr,newv,rb1,cv,rpool
,pl,rc,yyy] =
crossb(chrv,fitness,k,stk,chrw,sumchr,newv,rb1,c
v,rpool,pl,rc);
if yyy>0
[chr,fitness,k,stk,chrw,sumchr,newv,rb1,cv,rpool
,pl,rc,yyy] =
crossb(chrv,fitness,k,stk,chrw,sumchr,newv,rb1,c
v,rpool,pl,rc);
end
if yyy>0
[chr,fitness,k,stk,chrw,sumchr,newv,rb1,cv,rpool
,pl,rc,yyy] =
crossb(chrv,fitness,k,stk,chrw,sumchr,newv,rb1,c
v,rpool,pl,rc);
end

for i=1:10
    while rpool(rc,i,8)>0

[chr,fitness,k,stk,chrw,sumchr,newv,rb1,cv,rpool
,pl,rc] =
mutate(chrv,fitness,k,stk,chrw,sumchr,newv,rb1,c
v,rpool,pl,rc);
    end
end
rc=rc+1;

```

## References

- Bakhtyar R, Meraji SH, Barry DA, Yeganeh-Bakhtiary A, Li L (2012) An application of evolutionary optimization algorithms for determining concentration and velocity profiles in sheet flows and overlying. *J Offshore Mech Arct Eng* 134(2): 021802.1–021802.10
- Darwin Charles (1859) *On the origin of species*. John Mayer, London
- Erghott M, Klamroth K, Schwehm C (2004) An MCDM approach to portfolio optimization, *European J of Oper Res* 155(2004): 752–770
- Gen M, Cheng R (1997) *Genetic algorithms and engineering design*. Wiley, New York
- Holland JH (1975) *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, Ann Arbor
- Lee WC (2011) Redefinition of the KMV model's optimal default point based on genetic algorithms—evidence from Taiwan. *Expert Syst Appl* 38(8):10107–10113
- Lin CM, Gen M (2007) An effective decision based genetic algorithm approach to multi-objective portfolio optimization problem. *Appl Math Sci* 1(5):201–210
- Markowitz H (1952) Portfolio selection. *J Financ* 7(1):77–91
- Melanie M (1998) *An introduction to genetic algorithms*. MIT Press, Cambridge
- Pereira R (2000) *Genetic algorithm optimization for finance and investments*, MPRA 8610. Univeristy Library of Munich, Germany
- Seflane S, Benbouziane M (2012) Portfolio selection using genetic algorithm. *J Appl Financ Bank* 2(4):143–154
- Sinha P, Goyal L (2012) Algorithm for construction of portfolio of stocks using Treynor's ratio, MPRA paper 40134. University Library of Munich, Germany
- Yang X (2006) Improving portfolio efficiency: a genetic algorithm approach. *Comput Econ* 28(1):1–14