Genetic Algorithms in Multi-Stage Portfolio Optimization System

Man-Chung CHAN¹, Chi-Cheong WONG¹, Bernard K-S Cheung², Gordon Y-N Tang³

¹Department of Computing, The Hong Kong Polytechnic University, Hong Kong
²GERAD and Ecole Polytechnique de Montreal, Canada
³Dept of Finance and Decision Sciences, Hong Kong Baptist University, Hong Kong

e-mail: csccwong@comp.polyu.edu.hk
Tel: + (852) 2766 4900

Abstract

Portfolio optimization problem decides the percentage of the overall portfolio value allocated to each portfolio component with specified risk-return characteristics. A multi-stage stochastic optimization manages portfolio in constantly changing financial markets by periodically rebalancing the asset portfolio to achieve return maximization and/or risk minimization. This paper presents a decision-making process that incorporates Genetic Algorithms into multi-stage portfolio optimization system. The objective function is to maximize one’s economic utility or end-of-period wealth. The performance of our system is demonstrated by optimizing the allocation of cash and various stocks in Shenzhen market of China. Experiments are conducted to compare performance of the portfolios optimized by different objective functions in terms of expected return and standard derivation.

Keywords: Genetic Algorithms, multi-stage stochastic optimization, asset allocation

1. Introduction

Financial planning involves asset allocation and risk management. Asset allocation problem decides the percentage of the overall portfolio value allocated to each portfolio component. Risk management measures the risk of different investment instruments and creates or maintains portfolios with the specified risk-return characteristics. A multi-stage stochastic optimization is a quantitative model that integrates asset allocation strategies and saving strategies in a comprehensive fashion. It manages portfolio in constantly changing financial markets by periodically rebalancing the asset portfolio to achieve return maximization and risk minimization. The multi-stage optimization technique captures dynamic aspects of the problem, leading to optimal portfolio.

Optimization of asset allocation is complex and NP-hard. It is non-linear with many local optima. To solve the asset allocation problem, one may employ linear programming solvers such as CPLEX and OSL. The nonlinear terms in the objective function can be piecewise linearized (see [Carino(1995)]). The interior-point algorithms are well suited to the scenario structure of multi-stage stochastic programs. Searching the global solution by these methods is computationally expensive and ineffectively. Since time is a constraint for financial problems, a trade-off should be made between the performance and the computational time. Heuristic methods provide some appropriate ways to find optimal asset allocation. Berger ([Berger(1995)], [Berger(1996)]) applied
Tabu Search, an adaptive memory programming, to the problem. Their method improves computational performance by over 200 times as compared with interior-point methods for solving the problem.

Tabu Search is systematic and strategic towards the problem. We use a less problem-dependent heuristic method – Genetic Algorithms (GA) as our self-learning portfolio optimizer to optimize one’s asset allocation. GA reassures a higher chance of reaching a global optimum by starting with multiple random search points and considering several candidate solutions simultaneously. The unique crossover operator in GA offers the possibility of exchanging attributes among potential solutions.

In next section, the multi-stage portfolio optimization model is described. In section 3, we mention how GA applies to the portfolio problem. The system development is described in section 4. The experimental results are described in section 5. Finally, conclusions are drawn in section 6.

2. Multi-Stage Portfolio Optimization Model

Single period portfolio optimization model possesses several drawbacks. For examples, the risk is inconsistent over time. The multi-stage stochastic model [Mulvey(1997)] captures dynamic aspects of asset allocation problem. It manages portfolio in constantly changing financial markets by periodically rebalancing the asset portfolio to achieve return maximization and/or risk minimization, leading to optimal portfolio.

The stochastic nature incorporates scenario analysis into the model. Each scenario depicts a single path over a multi-stage planning period, sharing the same history. In our system, scenarios are defined by the changes of market index. For example, we can simply set two scenarios as: (1) the market index has been dropped and (2) the market index has been raised.

Suppose we have to optimize $A$ assets, with 1 denoting cash and the others may represent any investment instruments such as bonds, funds, futures and stocks. Let the entire planning horizon $T$ be divided into a number of periods as $t = \{1,2,3,\ldots,T\}$. Investment decisions are made at each of the period. Each period may have different scenarios. A graphical scenario tree can be constructed to visualize the optimal dynamic balanced investment strategy for asset allocation. Figure 1 depicts a scenario tree with two scenarios and three time periods.
The mathematical formulation of the portfolio optimization is described as follows:

**Parameters:**

\[ r_{t,i}^s = 1 + \rho_{t,i}^s, \text{ where } \rho_{t,i}^s \text{ is the return percentage of asset } i \text{ at time } t \text{ under scenario } s \]

\[ \pi_s \text{ is probability that scenario } s \text{ occurs, thus } \sum_s \pi_s = 1 \]

\[ w_0 \text{ is the wealth at the beginning of time } 0 \]

\[ w_t^s \text{ is the wealth at the beginning of time } t \text{ under scenario } s \]

\[ v_{t,i}^s \text{ is the amount of money in asset } i \text{ at the beginning of time } t \text{ under scenario } s \text{ before rebalancing} \]

\[ T \text{ is the time steps considered in investment decision-making} \]

**Decision variables:**

\[ x_{t,i}^s \text{ is the amount of money allocated to asset } i \text{ at time } t \text{ under scenario } s \text{ after rebalancing} \]

\[ p_{t,i}^s \text{ is the amount of asset } i \text{ purchased for rebalancing at time } t \text{ under scenario } s \]

\[ d_{t,i}^s \text{ is the amount of asset } i \text{ sold for rebalancing at time } t \text{ under scenario } s \]

\[
\begin{align*}
\text{Max } Z = & \sum_s \pi_s \text{Measure}_t^s \\
\text{subject to } & \sum_i x_{t,0}^s = w_0 \quad \forall s \quad \text{..........................................................(1)} \\
& \sum_i x_{t,i}^s = w_t^s \quad \forall s, \quad t=1,\ldots,T \quad \text{..........................................................(2)} \\
& v_{t,i}^s = r_{t-1,i}^s x_{t-1,i}^s \quad \forall s, \quad \forall i \in A, \quad t=1,\ldots,T \quad \text{..........................................................(3)}
\end{align*}
\]
\[
x_{i,t}^s = v_{i,t}^s + \sum_{j=1}^{d_{i,t}} d_{i,t}^s - \sum_{j=1}^{p_{i,t}} p_{i,t}^s \quad \forall s, \quad t=1,\ldots,T, \quad s \neq s' \quad (4)
\]

\[
x_{i,t}^s = v_{i,t}^s + p_{i,t}^s - d_{i,t}^s \quad \forall s, \quad t=1,\ldots,T, \quad i \neq 1 \quad (5)
\]

\[
x_{i,t}^s = x_{i,t}^s \quad \text{for all scenarios } s \text{ and } s' \text{ with identical past up to time } t \quad (6)
\]

where \( \text{Measure}_t^s \) is performance measure value under scenario \( s \) at time period \( T \).

Constraints (1) and (2) state the initial total wealth and the total wealth under scenario \( s \) at the beginning of time period \( t \) respectively. Constraint (3) states the wealth accumulated at the end of \( t \)-th period under scenario \( s \) before rebalancing in asset \( i \). Constraints (4) and (5) are the balance constraint for cash and other asset categories respectively. Constraint (6) is the non-anticipativity conditions stipulate that decision variables are equal to each other whenever they share a common historical past up to time \( t \) in the planning horizon \( \{0,1,\ldots,T\} \).

The objective function is performance measure weighted by occurrence probability of scenario. The probability-weighted optimization means that the optimization focuses more on the scenarios with higher occurrence probability than those with less occurrence probability. The probability of the occurrence of each scenario can be generated by historical statistics or any forecasting system. If one favours unbiased optimization, we may assign the same weights to all scenarios. In this paper, we adopt historical statistics to generate the occurrence probability of each scenario. The performance measures we use are described in the next section.

The portfolio management process of our system is graphically represented in the figure 2.

![Figure 2: Portfolio Management Process](image-url)

Investor’s risk aversion level and assets selected are the inputs of our asset allocation optimizer which optimizes the allocation of the selected assets so as to maximize economic utility or end-of-period wealth. The optimized portfolio composition and its performance in terms of its average return and variance are displayed to the user. Based on the portfolio performance, the user may review risk aversion level or the selected asset if necessary.
3. Genetic Algorithms as Asset Allocation Optimizer
3.1 Genetic Algorithm Approach

Genetic algorithms (GA) are search algorithms inspired by natural evolution that mimics operation in natural genetics to search the optimal solution in a solution space. Genetics provides the chromosomal representation to encode the solution space of the problem. GAs are theoretically and empirically proven to provide robust search in complex space effectively (see [Goldberg(1989)]). Their evolutionary procedures based on the survival-of-the-fittest fashion by gradually manipulating the potential problem solutions to obtain the more superior solutions in population.

GA starts with a population of randomly generated solutions called chromosomes to explore the solution space of a problem. Then GA searches the improvement of solutions through a number of iterations called generations. The performance of each solution is evaluated by a fitness function, which always contains the objective function. In each generation, relatively good solutions have a higher chance to be selected for reproduction of offspring by genetic operators – crossover and mutation. Crossover combines materials from parents to produce their children. Crossover provides pressure for improvement or exploitation while mutation makes small local changes of feasible solutions to provide the variability of the population. The reproduction cycle goes on until the maximum number of iterations is run or there is no further improvement in consecutive generations.

The exploration of feasible solutions made by random initialization depends on the population size. Small population size provides an insufficient sample size, causing premature performance while a large population size requires more time to converge the population. Krishnakumar [Krishnakumar(1989)] develops Micro-genetic algorithm (µGA) which runs with small population size to shorten the computational time. The key to success of µGA with small population size is in bringing new strings into population by random generation of new chromosomes when no convergence occurs for a number of generations. The "start and restart" procedure of µGA infuses new schema. So µGA helps to avoid premature convergence and is always looking for better strings.

Other variants of GA are used as well. Firstly, fitness of each chromosome is scaled before selection process to regulate the level of competition among members of the population so that the extraordinary individuals cannot take over a significant proportion of the finite population in a generation, leading to genetic similarity of their offspring. Furthermore, elitism is used to preserve the best chromosome in each generation so as to increase the speed of the search.

The skeleton of µGA is depicted as follows.
1. Initialize a small population randomly
2. Evaluate each chromosome.
3. Applying elitist selection, carry the best individual to the next generation.
4. Select chromosomes for reproduction.
5. Apply crossover and mutation to reproduce the next generation.
6. Evaluate the new chromosomes.
7. If the termination condition is satisfied, return the best solution; if not, carry the elite to the next generation.
8. If the reshuffling condition for population is reached, randomly generate the remaining individuals and then go to 6; if not, go to 4.

The skeleton of $\mu$GA is depicted in the following flow chart (figure 3).

![Flowchart of Micro-Genetic Algorithm](image)

Figure 3  Flowchart of Micro-Genetic Algorithm

The population is reshuffled when it has been converged which can be indicated by the following conditions:

1. The population has not been improved in specified consecutive generations after last restart procedure
2. The percentage of the total number of the different genes in the same position of chromosomes is less than a specified GA parameter -- *heterogenetic threshold*.

   GA is terminated under either of the following conditions:
1. The population has not been improved in specified consecutive generations for termination
2. The maximum number of generations has been passed.

Micro-Genetic Algorithm is used as self-learning portfolio optimizer to optimize one’s asset allocation under different scenarios over different time periods within the planning horizon in terms of performance measures.

### 3.2 Chromosomal Representation

Our decision variables are the allocation of various selected assets under different scenarios over the planning horizon. These decision variables are encoded in a chromosome for GA implementation. Firstly, the scenario tree is represented sequentially into an array. For example, we index each node of the scenario tree with two scenarios and three time periods as follows:

![Figure 4](image)

Figure 4  A scenario tree with two scenarios and three time periods

The sequential representation of this scenario tree is:

![Figure 5](image)

Figure 5  Sequential representation of scenario tree in figure 4

Each node of the array contains the allocation proportions of all assets under the corresponding scenario. The array can be interpreted as a chromosome as follows.
Each allocation proportion values are stored as 7 bits in a chromosome, representing from 0 to 127. Hence, the total number of bits in a chromosome is \( 7A(\sum_{i=0}^{T} S^i) \), where \( A \) is the number of assets to be optimized and \( S \) is the total number of scenarios in a time step.

For each scenario \( s \) at time \( t \), the total asset allocation proportion must be equal to 100%. Hence, each of asset allocation proportion \( a_{it}^s \) under the same scenario is normalized by \( a_{it}^{s'} = \frac{a_{it}^s}{\sum a_{it}^s} \) where \( a_{it}^{s'} \) is the normalized asset allocation proportion.

### 3.3 Fitness Function as Portfolio Evaluation

Although mean-variance model is popular in evaluating a portfolio, [Dahl(1993)] and [Elton(1995)] state that it is descriptive of investors with a quadratic utility function which has unrealistic properties. For example, absolute risk aversion increases with quadratic utility functions.

Economic utility is a more realistic one in evaluating a portfolio. The power law utility function or called von Neumann-Morgenstern utility function [Mulvey(1993)] is used since it has constant relative risk aversion for any value of risk-aversion value \( \gamma \) (i.e. the percentage invested in risky assets remains unchanged as wealth increases). The corresponding performance measure function is

\[
Measure = \begin{cases} 
\ln(w) & \text{if } \gamma = 0 \\
\frac{(w)^T}{\gamma} & \text{elsewhere}
\end{cases}
\]

The advantage of using utility function is that it can generate investment decisions for a wide range of risk-bearing attitudes: Utilities with \( \gamma > 1 \) describe risk-seeking or “thrill-seeking” behaviour and those with \( \gamma < 1 \) describe risk-averse behaviour. The smaller the values of \( \gamma \), the more sensitive to the loss the investor is. It can be seen when \( \gamma = 1 \), the utility function is exactly end-of-period wealth and risk is not taken into account in this evaluation. So it is suitable for risk-neutral investor. It is noted that the optimal growth strategy results when \( \gamma = 0 \).

Power utility function is used to evaluating the quality of a portfolio. Hence, our fitness function is
\[ Z = \begin{cases} \sum_{s} \pi_s \ln(w^*_s) & \text{if } \gamma = 0 \\ \sum_{s} \pi_s (w^*_s)^\gamma & \text{elsewhere} \end{cases} \]

4. Portfolio Optimization System

A portfolio optimization system is developed according to the model described in the previous sections. Figure 7 is the architecture of our portfolio optimization system.

![Figure 7 Architecture of Portfolio Optimization System](image)

Financial data are stored in Oracle database. The daily closing price of stocks, that of the market index and the interest rate are collected for our portfolio optimization.

Our portfolio optimization system is developed on the internet environment. Genetic algorithm parameters and other portfolio optimization parameters can be set by the administrative users on the screens shown in figures 8-10. The end users can select their favourite assets and input their risk aversion level on the screen shown in figure 11 where the names of stocks for selection are written in Chinese. Once the end user clicks the “Optimize Asset Allocation” button, the portfolio optimization process proceeds by means of Genetic Algorithms. The optimization result is then shown on the screen as figure 12-14. The scenario tree is shown in figure 12. When the node of a scenario is clicked, the screen will display the precondition of the scenario, the expected return and the variance of the precondition and the optimal asset allocation under the scenario (see figure 13). Finally, the average return and variance of the optimized portfolio under different scenarios and overall scenario in the training phase and the testing phase are displayed as well (see figure 14).

If the optimization process is performed when an ad hoc portfolio optimization is requested, each chromosome evaluates every historical event at each iteration, leading to long optimization time and response time to the system. The end-users may lack patience to wait for the optimal portfolio found by the optimizer. In order to speed up the optimization process for the internet environment, preprocessing is necessary for shortening the optimization to increase the efficiency of the system.

Since there is enormous number of possible asset combinations for the end users to select for asset allocation optimization, it is impossible to optimize all possible asset combinations beforehand. Instead, the data required in the optimization process is preprocessed from the raw data and the optimization is performed with this preprocessed data when an ad hoc demand is requested.
Figure 8 User interface for setting financial parameters

Figure 9 User interface defining scenarios

Figure 10 User interface for setting GA parameters

Figure 11 User interface for setting assets to be optimized and risk aversion level
Once the financial parameters such as the number of scenarios, the range of each scenario, time switch, training period and testing period have been set, the expected
return of each asset, variance of each asset and the covariance between assets are estimated based on the historical data over the training period and then stored in the database for ad hoc portfolio optimization. When an end user selects his or her favourite assets, our portfolio optimizer shortened its optimization time by evaluating the candidate portfolio with these summarized preprocessed data, instead of raw data. In this way, an optimal portfolio can be obtained effectively.

5. Experiments and Results

In order to evaluate the system, experiments were carried out. Daily closing prices of Shenzhen Composite Index and Shenzhen stocks were collected. Eighteen stocks shown in figure 11 and cash were selected to be optimized. The entire planning horizon was divided into three periods. Each period lasts a month and has two scenarios: rise and drop. The training period was from 1 January, 1994 to 1 December, 1998. The testing period follows until 31 December, 2000.

The parameters of μGA are determined empirically. To find out the suitable value of a parameter, this parameter is varied while the other parameters are fixed. Select the value that has the best average performance in ten runs. As a result, the following μGA parameters were used:

- Population size: 4
- Crossover rate: 0.2
- Mutation rate: 0.01
- Maximum number of non-improving generations for reshuffling: 20
- Heterogenetic threshold: 0.01
- Maximum number of non-improving generations for termination: 71
- Maximum number of generations: 10000 (which is sufficient large enough so that GA is rarely terminated by this criterion)

The risk aversion parameter was varied to test the performance of the system in terms of expected return, variance and utility value weighted by the probability of occurrence of scenarios. The performance of market index, equal weighted portfolio and GA-optimize portfolio with various risk aversion parameter in the testing phase are shown in figures 15-17.

In comparison of GA-optimized portfolio and equal weighted portfolio, GA-optimized portfolio has a higher expected return and utility value for any risk aversion parameter. For risk aversion parameter –1 and 0, GA-optimized portfolio has a lower variance value. However, for risk aversion parameter 1, it has a higher value than equal weighted portfolio. Since risk aversion parameter of 1 is used by the risk-neutral investors, the variance metrics should not be considered in this case.
Figure 15: Performance of market index, equal weighted portfolio and GA-optimized portfolio with $\gamma = -1$

Figure 16: Performance of market index, equal weighted portfolio and GA-optimized portfolio with $\gamma = 0$

Figure 17: Performance of market index, equal weighted portfolio and GA-optimized portfolio with $\gamma = 1$
Compared with market index, GA-optimized portfolio has a higher expected return and utility value but a larger variance. Investors can select their investment between them according to their attitude toward expected return and risk.

6. Conclusions & Discussions

In this paper, we use micro-genetic algorithm to optimize a multi-stage portfolio. Among market index, equal weighted portfolio and GA-optimized portfolio, GA-optimized portfolio has the highest expected return and utility value for any risk aversion parameter. For non-risk neutral investor, GA-optimized portfolio has a lower variance value than equal weighted portfolio. GA-optimized portfolio has a larger variance than market index. It provides an alternative for investors.

Future direction of the research should investigate new strategies for improving the performance. Research can be focused on developing a hybrid form of algorithms, which combine the good features of different algorithms, says Genetic Algorithms and Tabu Search. Saving computational time is very important. Time can be saved from parallel or distributed processing.
References


