GENETIC ALGORITHMS FOR FINANCIAL

PORTFOLIO SELECTION

By

YI DENG

Bachelor of Computer Science

University of Science & Technology Beijing

Beijing, China

2002

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 2008

GENETIC ALGORITHMS FOR FINANCIAL

PORTFOILIO SELECTION

Thesis Approved:

Xiaolin Li

Thesis Adviser

John P. Chandler

Johnson Thomas

A. Gordon Emslie

Dean of the Graduate College

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Motivation

Portfolio management is a critical financial task, in which investors try to select an optimal portfolio to maximize their profits and minimize the risks. The conventional portfolio theory and models which originated from Markowitz have relied mostly on mathematic models and included only quantifiable objective variables. In Markowitz portfolio theory, he proposed “mean-variance criterion” for selecting an optimal portfolio. In other words, when we assume the risk is fixed, it maximizes the rate of return; or when we assume the rate of return is fixed, it minimizes the risk. Because of the time and cost requirements, the performance of this mode is not very attractive, and it is not broadly accepted by the market. The Langrangian objective function can apply differential to obtain weights of portfolio, but differential cannot handle inequality constraints. Another tool is quadratic programming method, which needs differential also and it can solve the effective and quantifiable problems. However, when the quadratic programming tries to seek an efficient portfolio, so many parameters with very wide range of values can yield unpractical results [1]. To overcome this shortness, Sharpe developed a single index model, which can simplify the portfolio variance of the quadratic programming method and determine the weights of efficient portfolio. This method has assumed that the interrelations of returns of stocks are subsequences of random factors and market factors, but in this model, the inappropriate selection of index will cause misunderstanding.

The obvious limitation of above conventional portfolio model includes [4]: 1. only linear portfolio problem can be solved. 2. Computer procedures and parameters are too complicated and too many assumptions needed. 3. The influence variables for portfolio cannot be used flexibly. The characteristic of time series and nonlinear plus unpredictable chao systems in the financial market cannot be simulated and achieved by any conventional analysis method. Consequently, the research work brings up the genetic algorithms to select portfolio.

1.2 Simple Genetic Algorithms

Genetic algorithm was first illustrated by Holland in 1975 [2]. It adopted the concept that only the strongest individual survives in an environment from Darwin's natural selection theory. Holland's algorithm was commonly called simple Genetic Algorithm, in which a string of characters is applied to simulate chromosome of living things. The fitness value of each chromosome is computed to evaluate its adaptability to the environment and determines how many offspring the chromosome will have in next generation. Certain operations will be applied on each generation to maintain the variety and the creativity of each generation. Thus, the working of the SGA can be summarized as Table 1.1:

```
Algorithm GA;
{
        Initialize population;
        Generation :=0;
          Repeat
               Generation = Generation +1;
              Selection (population);
              Crossover (population);
               Mutate (population);
               Until Termination_Criterion;
  }
```

Table 1.1 Simple Genetic Algorithms

This SGA has the following components [2]:

a population of binary strings,

a mechanism to encode the solutions as binary strings,

control parameters,

a fitness function,

a selection mechanism , and

genetic operators(cross over and mutation)

The classical GA processes as follows [2]:

1 a population of chromosomes (strings) is created.

2 the chromosomes are evaluated by a defined fitness function.

3 some of the chromosomes are selected for performing genetic operations.

4 crossover and mutation are performed according to probabilities.

Table 1.2 shows a generational cycle of the genetic algorithms with a population of four strings with 10 bits each. The fitness function performs a “count the number of 1 and divide by 10” to normalize the value to the range of (0, 1).

Population 1:		
	Individual	Fitness value
I1	1010010000	0.3
I2	1000011111	0.6
I3	0110101011	0.6
I4	1111111011	0.9
Population 2:		
	Individual	
I1	1000011111	0.6
I2	0110101011	0.6
I3	1111111011	0.9
I4	1111111011	0.9
Population 3:		
	Individual	
I1	1000011011	0.5
I2	0110101011	0.6
I3	1111111011	0.9
I4	1111111111	1.0
Population 4:		
	Individual	
I1	1000011011	0.5
I2	0110111011	0.7
I3	1111111011	0.9
I4	0111111111	0.9

Table 1.2 Operator of SGA [2]

In this generational cycle, from Population1, we obtain that the fourth individual I_4 can generate two offspring, and each of the I_2 and the I_3 can generate one respectively. Then, I_1 and I_4 form one pair, as well as I_2 and I_3 from the other pare to perform the crossover and mutation operations with the probability of 0.5 and 0.05 respectively, which means only one pair is crossovered and only two bits out of 40 are mutated. In population3, the pair of I_1 and I_4 has been actually crossed and in population4, the sixth bit of I_2 and the first bit of I_4 obtain the chance to mutate.

In a typical SGA, control parameters must be specified before its execution including the fixed number of generation which could be a stopping criterion of the algorithm.

1.3 Research Overview

There have been a great number of the studies that focused on the domain of portfolio management or investment strategy using GA related models. Most of these researches and models have made great improvements in different aspects of traditional genetic algorithms, and their performances in rate of return are tested with various stock markets. As different models obtained their investment return pictures within certain time periods from different stock markets, the capabilities and performances of these models and ideas cannot be decided and compared easily. To obtain the first hand experience of how the features and parameters influence the performance of GA, we decided to design and implement our own adaptive model to examine the rate of return and other features of the actual models carrying different alternations and parameters with uniform market data .

The overall goal of this study can be divided into two sections. 1. Review and classify some of the typical research achievements on GA applied in investment domain. 2. Design and implement a general adaptive GA model in C++, and use the uniform history market data to evaluate the performance of selected improvements that have been mentioned in literature review section.

1.4 Contributions

- Recent developments on GA applying on portfolio investment field are classified and compared.
- A conventional GA model facing financial problems has been implemented in object-oriented language (c++), features and parameters of GA can be easily

modified. And the model can be redeveloped by users for other financial problems, such as option pricing problems.

- Based on the conventional GA, an Adaptive model is designed and implemented for more suitable for the changing environment, and both of the models are tested on history market data of a 100 AMEX stock pool. The experimental results are analyzed and compared, and a conclusion has been drawn.

1.5 Outlines of the Thesis

The remainder of the thesis is organized as follows. Chapter two reviews and categories the applications of improved GA in portfolio selection and other financial domains. Chapter three describes the design and architecture of our research model, and the programming issues of the model. Finally, Chapter four draws the conclusion by studying the experimental results, and also discusses the probability for improvement in future work.

CHAPTER 2

REVIEW OF LITERATURE

2.1 Application of GA in Quantitative Finance

As one of the most popular methods in artificial intelligence area, Genetic algorithm has been applied to most of the financial, particularly quantitative financial domains depending on what the chromosome represents and how the string is coded. The applications of GA in this field could be divided into four categories: portfolio management [1, 4, 7, 22], investment strategy [8, 11, 13], option pricing [21, 22, 23] and financial distress prediction [9, 10, 29], shown in Figure 2.1. The following has briefly discussed the application of GA in these four sub-areas.

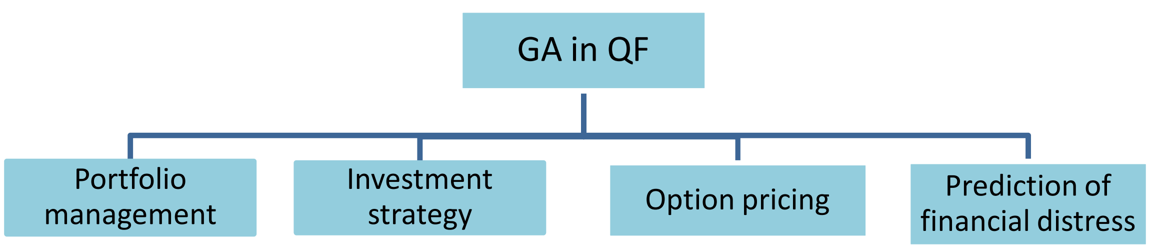


Figure 2.1Application of GA in QF

1) In portfolio management, the most common coding mechanism is to use one binary bit to represent the decision on one single stock. If '1' is to represent the 'select', then '0' is to represent 'not select'. There are also some researches which have apply

natural integers to represent the proportion of that stock in the portfolio, where if the value if larger, the amount purchased in bigger [4] [22]. As different chromosome stands for different portfolio, the fitness value of each individual will depends on the performance of the portfolio it represents.

2) Other than binary strings that represent the actual portfolio, the strings can also be coded as a sequence of trading rules or strategies. An idea of moving averages is frequently seen in theses coding methods [8] [13]. For example, If M_1, M_5, M_{25}, M_{50} and M_{100} represent one certain stock's average price of last one, five, twenty five, fifty, and one hundred days respectively, a map can be built from these five moving averages to five numbers, as:

$$\begin{array}{ccccc} M_1 & M_5 & M_{25} & M_{50} & M_{100} \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & 2 & 3 & 4 & 5 \end{array}$$

Then one tube of these five numbers, such as (3 4 2 1 5) in buy part can be interpreted as:

IF ($M_{25} > M_{50} > M_5 > M_1 > M_{100}$) , Then Buy;

While (5 2 1 4 3) in sell part can be explained as :

IF ($M_{100} > M_5 > M_1 > M_{50} > M_{25}$) , Then Sell;

After coding, a pair of tubes in buy part can be picked up from the population to perform any GA operator as crossover and mutation, so does pair in sell part.

3) Genetic algorithms are applied in the option pricing problems, which is one of the most complex financial issues. In these researches, the fitness functions of GA's simulation are derived from the Black- Scholes Option pricing model, and the results

from GA model are compared with the exact solution yielded by Black-scholes option pricing theory [21] [22].

4) Financial distress prediction is another aspect that the GA can be applied to. The researchers from British universities have formed the GA's chromosome by a list of financial ratios [9], such as Gross profit margin**,** Return on net assets (RONA), Current ratio, and so on. The aim is to select the most informative financial ratios that can effectively predict the financial distress. A certain number of failed or continuous companies are chosen to form a pool for testing. As the principle to select is to maximize the contained discriminatory information from these ratios and minimize their co-linearity, the fitness value of each chromosome (ratios list) reflects its discretionary cutting down its co-linearity [10]. In other researches, not only financial ratios are considered as the evaluating criteria of financial distress, but the corporate governance features are introduced into the GA chromosome coding [29].

2.2 Major Improvements in GA

Portfolio management is an important facet of financial management on which genetic algorithms and many other AI methods have been applied on it. Considering the limitation and deficiency of the simple genetic algorithm, recent research works have focused on the following directions of improvements. In this chapter, we will briefly discuss and category some of these studies and improvements.

- Improvement in crossover and mutation operators [3] [4].
- Combined GA with other methods [5] [6] [7].
- Develop User-oriented GA model [8].

2.2.1 Improved Crossover and Mutation Operators

Crossover and mutation are common used operators in GA to keep and increase variety to the next generation. In simple GA, the randomly picked pair of chromosomes will be subjected to crossover only if a randomly generated number in the range 0 to 1 is greater than p_c, the predetermined probability of crossover. Otherwise the pair of strings remains unchanged. Similar to crossover, mutation is another operator with the role of restoring lost genetic information. Also, a probability of mutation, denoted by $p_{m,}$ will give the probability that whether a bit will be flipped or not.

As the selected probabilities of crossover and mutation affect the efficiency and quality of the generation evolvement, improved crossover probability p_c and mutation probability p_m have been proposed as follows [4].

$$P_c = \begin{cases} k_1 \sin[(\pi/2)(f_{max} - f_h)/(f_{max} - f_{avg})], & f_h > f_{avg} \\ k_2, & f_h \leq f_{avg} \end{cases} \tag{2.1}$$

$$P_m = \begin{cases} k_3 \sin[(\pi/2)(f_{max} - f_h)/(f_{max} - f_{avg})], & f > f_{avg} \\ k_4, & f \leq f_{avg} \end{cases} \tag{2.2}$$

f_{avg} is the average fitness value in population; f_{max} is the highest fitness value of the population; f_h is higher fitness value between two individuals which will carry out the crossover operation ; f is the fitness value which will carry out the mutation operation.

In this new improved GA solution, the selected parents who have been randomly assigned into groups will form their offspring by generating the specific digit as following:

$$X_1(i) = cP_1(i) + (1-c)P_2(i) \tag{2.3}$$

$$X_2(i) = (1-c)P_1(i) + cP_2(i) \tag{2.4}$$

Where P_1(i), P_2(i) are *i*th digit of the two selected parents , X_1(i), X_2(i) are the offspring of these two parents , while c is a random number in(0,1). In this improvement, the crossover and mutation probabilities are changed based on the status of the entire population, the larger probabilities will be imposed on population whose fitness values are less diverse. The new solutions of selection, crossover and mutation can enhance the robustness of the new population.

2.2.2 GA Combined with Other Methods

(1) Genetic algorithms in multi-stage asset allocation

Single period asset allocation model possesses limitations because its risk is inconsistent from time to time. Thus, the multi-stage investment decision model is developed to capture dynamic aspects of asset allocation problem [8]. It manages portfolio in constantly changing financial markets by periodically relocating and rebalancing the portfolio leading to optimal portfolio.

In multi-stage asset allocation model, investment decisions are made at each of the periods as t = {1, 2, 3,…,T} of the entire planning horizon T. A graphical scenario tree can visualize the optimal dynamic balanced investment strategy for asset allocation. Two researches from Taiwan have depicted a scenario tree with two scenarios and tree time periods as Figure 2.1.

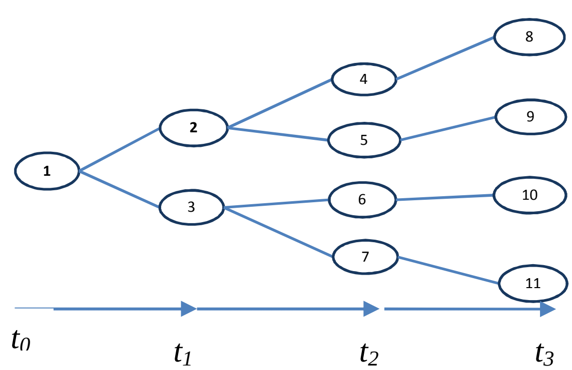


Figure 2.1A Scenario Tree [8]

In this model, the probability of the occurrence of each scenario can be generated by historical statistics or any forecasting system. The 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. Generic Algorithm is used as a portfolio optimizer to optimize asset allocation under different scenarios over different time periods.

(2)Combined GA model with fuzzy set

Recently, the traditional methods combining with intelligent methods (such as Genetic Algorithm, Genetic Programming.) have been applied to solve investment decision problem. Among those approaches, many combine the fuzzy decision with Genetic Algorithm, ([6], [7]). The processes of constructing those combining models usually include following steps. First, to earn the maximum return with minimum risk rate in portfolio selection, the multiple-objective optimum is described as:

Objective 1 : max E(r_p) = max E(r_p) = $X^T R$ $\quad r_p = \sum_{i=1}^{n} x_i r_i$ (2.5)

Objective 2 : min σ^2 (r_p) = $X^T \sum X$ (2.6)

$r = (r_1, r_1, \ldots r_n)^T$ is investment earning rate vector of each stock;

$R = (R_1, R_2, \ldots R_n)^T$ is expect vector of earning rate vector r;

$X = (x_1, x_2, \ldots x_n)^T$ is investment proportion vector of portfolio;

$\sum = (\sigma_{ij})_{n\times n}$ is the covariance of earning rate vector r,

where σ_{ij} = Cov(r_i,r_j), $i,j = 1,2,\ldots,n$;

Then, multiple-objective is converted into the fuzzy multiple-objective decision model:

$$\text{Min } F(\mu, X) = (\mu^2 \sigma^2 (r_p) - (1-\mu)^2 E(r_p)) \quad (2.7)$$

$$\text{s.t } E^T X = 1$$

$Xj \geq 0$, $j = 1,2,\ldots,n$

Where: $E(r_p) = X^T R$ and $\sigma^2 (r_p) = X^T \sum X$;

$0 \leq \mu \leq 1$; show that if μ is larger, investment is unwilling to receive risk

Finally, the optimal solution of Portfolio investment model is obtained by using Genetic Algorithm. Use fuzzy chromosome $V = (\tilde{x}1, \tilde{x}2, \ldots \tilde{x}n)$ for fuzzy decision, where each gene is a fuzzy set but not a visibility number, and the fuzzy multiple-object function can be taken to define the fitness function as (2.8):

$$Fit(F(\mu, X) = (\mu^2 \sigma^2 (r_p) - (1-\mu)^2 E(r_p) \quad (2.8)$$

At last, the final effective evaluation of each investment portfolio chromosome X_i can be described as follow:

$$e(X) = Fit(F(\mu, Xi)) / \sum Fit(F(\mu, Xi)) \quad (2.9)$$

Then, the general GA methods can continue to solve the multiple-objective optimum problem based on evaluation function shown above.

2.2.3 GA on User-oriented Model

Some studies focused on developing complex methods based on GA to better reflect the financial market, while another way to make these models more practical is to consider preference and requirement of investors. Along with this direction, the researchers from Hong Kong develop a user-oriented invest decision making model [6].

The major improvement of this user-oriented portfolio selection model is to make the portfolio decision according to the user's preference and recognition. As the fitness function serves and represents as the environment of generations evolvement, the researchers design a new fitness function, shown as (2.10) in which there are 5 influence variables, and 3), 4) represent the users' requirement [6]:

1) Sharpe index: Assesses investment combination performance according to the unit risk size of rate of return

2) EPS: Earnings per share

3) Industrial categories

4) Industrial finance capability

5) Numbers of stock combine

Combining those five influence index, the fitness value of this model is shown as:

Fitness function = a* Sharpe index + b*EPS + c* stock combination number +d*industrial categories + e* Industrial finance capability (2.10)

2.3 The Alternative of GA – GP

Genetic programming is a very close related AI method, which has been considered to belong to GA by some researchers. It has also been used in quantitative finance fields as what GA does, such as investment decision [13] [27], financial predictions [15] [16]. In normal GP models, the candidate solution is represented as decision trees, while it's represented as a binary or integer strings in GA usually. The principle of evolvement of both are very similar, which is reflects the idea that only the best fitted individuals (trees in GP) can survived to the next generations. Many applications of GP have been combined with neural network and some of which are called Genetic network programming. In those models, the GA's operator such as crossover and mutation are still performed, but the evolvement of the candidate solution is in the form of a network [26] [28] [32]. As GP or GNP is not our research target, we will not introduce them in detail.

CHAPTER 3

ADAPTIVE GA MODEL

3.1 Conventional GA Model

To study the capabilities of GA in solving real problems, we construct a stock selection system and compute the Rate of Return of the portfolio generated by GA model. Before the construction of the adaptive GA, the basic conventional GA has been developed and applied to the portfolio problem, and has set up the fundamental parts for the more complex adaptive GA in the next section.

In the conventional GA model, the initial population will be generated randomly, and the population should evolve towards increasing the fitness function value. Based on previous research work, one fitness function will be applied at first, and to yield the best returning rate, a few of potential fitness functions might be tested and compared later. Also, the length of chromosome and the size of the population will be flexible to adopt alternation. A stock pool which includes the history data of 100 major US stocks will be used to test the GA model for the portfolio selection. The design of this conventional GA model is shown in Figure 3.1.

Transfer the real problem into the form of GA involves the problems of how to encode, select parameters and create the evolve environment. Therefore the model of research and design component is stated in detail as follows:

a. Encoding: we use a binary string to represent a chromosome, and each bit represents whether the stock will be selected or not. 100 major companies traded in AMEX will be the candidate stocks to be selected, thus the length of the chromosome is 100.

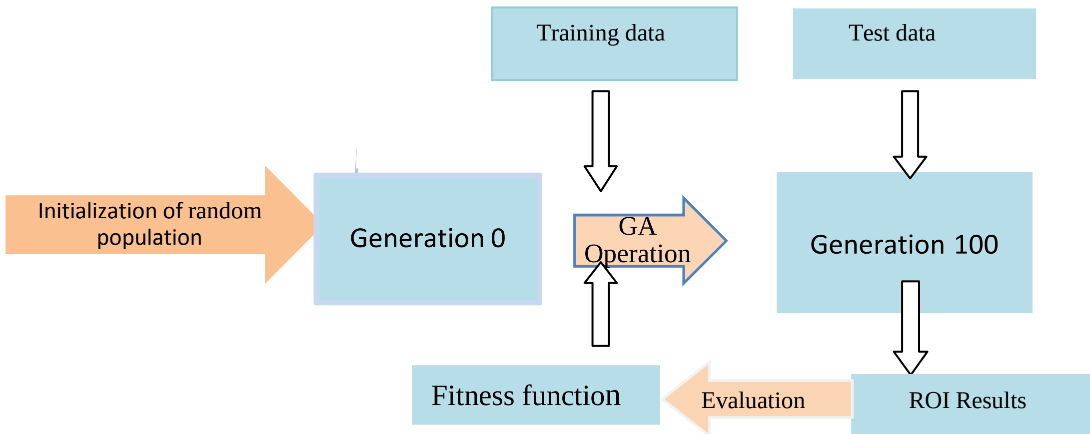


Figure 3.1 Design of Conventional GA Model

b. Selection Mechanism: As the GA population evolves, the chromosomes which have better fitness to the environment obtained higher chance to be selected to be in the next generation. Then, P_i is the probability of the ith chromosome to be selected as the parents in the selection process.

$$P_i = \frac{Fitness(C_i)}{\sum_{i=1}^{n} Finess(C_i)}, \quad i = 1,2,3 \ldots popsize$$

c. Crossover and Mutation: We use Two-point crossover to increase the diversity of the population. Both crossover and mutation are operated by a pre-determined rate.

d. Fitness function: As in this research, the goal is to find the optimal portfolio for investors, which should include the most valuable stocks for investment. According to the financial theory that the price of stock fluctuates around the intrinsic value of the stock, which can be simply simulated by the average price during certain past period, we use the ratio between this average price and the price of the purchasing day to represent the investment value. The larger this ratio is, the more investment value the stock possesses, because comparing to its potential value, the stock seems to be at a relatively low price. In this model, we consider this period as 256 days (around 1 calendar year), and the fitness function can be presented as following:

stock value ratio: $R_j = \frac{P_{AVEj}}{P_0} \times 100\%$ j = 0,1,2…N (chromosome length)

P_{AVEj} : the average price of past 256 days of stock j;

P_0 : the stock's current price ;

Then, the fitness function of *i*-th chromosome:

$$Fitness(i) = \left. \sum_{j=0}^{N'} \frac{P_{AVEj}^{i}}{P_0^{i}} \middle/ N' \right. \qquad i = 0,1,2,\ldots,Popsize$$

N: chromosome length, in this research target, the chromosome length is 100.

N' : number of selected stock, in fitness value, only the selected stock added.

The setting of testing parameters of the conventional GA is shown in Table 3.1.

Parameters	Numeral
Chromosome Length	100
Population size	60
Generation	100
Crossover rate	0.65
Mutation rate	0.05

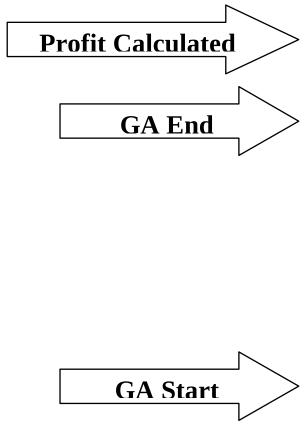
Table 3.1Conventional GA Parameters

3.2 Construct of Adaptive GA Model

In the conventional GA model, the selection of the portfolio will be highly influenced by two factors: 1) stock prices of the day that we make the portfolio; 2) the average stock prices of the last 256 days, which means that when the GA population evolves, it only evolves to fit one constant environment. As in the stock exchange and other financial markets, the circumstances are changing very fast and randomly. To place the working GA population into a continuously changing environment and select better portfolio, we design an adaptive GA model, or called sliding-window GA model.

3.2.1 Use of History Data

In this research, 100 stocks from AMEX will be used as the experimental data to test our models. In the conventional GA model, for each stock, only the purchase-day price, and the average price of 255 days before the purchase-day needs to be gathered to compute the fitness of this stock and this GA individual (one expecting portfolio), and also the price of the selling-day will be obtained to calculate the profit that this stock earns in a certain period of time. In adaptive GA, we start gathering the data one year before the purchase-day. For example, if we need to make portfolio on the date of Jan-01-2006, for conventional model, only the price of this day and the average price of the past 256 trading days (one calendar year) will be considered, and after 100 generations calculation and evolvement, the model will yield one best portfolio for this particular day (Jan-01-2006). While in the adaptive model, the first considered purchase day will be Jan-01-2005, one year before the actual purchase day, and a number of generations will be generated based on the price of this day and average price of the past 256 days. Then, the GA population will continue to evolve not based on the same data, but based on the price of the first day of each month, and the 256-day average price before that, as shown in the Figure 3.2.



Date	Price	other
…	…	…
1-Jul-06	5.17	
…	…	…
1-Jan-06	6.23	
..		
1-Dec-05	5.45	
1-Nov-05	7.34	
..		
..		
1-Feb-05	5.12	
..		
..		
1-Feb-04	4.54	
…		

Figure 3.2 Data Application of New GA

3.2.2 Two Phase of Evolvement

In the Adaptive GA model, the population evolvement process can be divided into two consecutive phases. In first phase, the GA performs as a conventional GA, using the same segment of data, and evolves a certain number of generations; and the yielded population from phase one will be the starting population of phase two. During the second phase, the population will only proceed 10 generations with the same segment of data, and then one month of new data will be included in, while the data of the earliest month will be discarded. As the fitness value of each GA individual will be computed by different data segment, the GA population will be evolving in a continuously changing environment, and should generate the individuals that are more fitting to the new information. The two-phase process of the sliding window is shown in Figure 3.3.

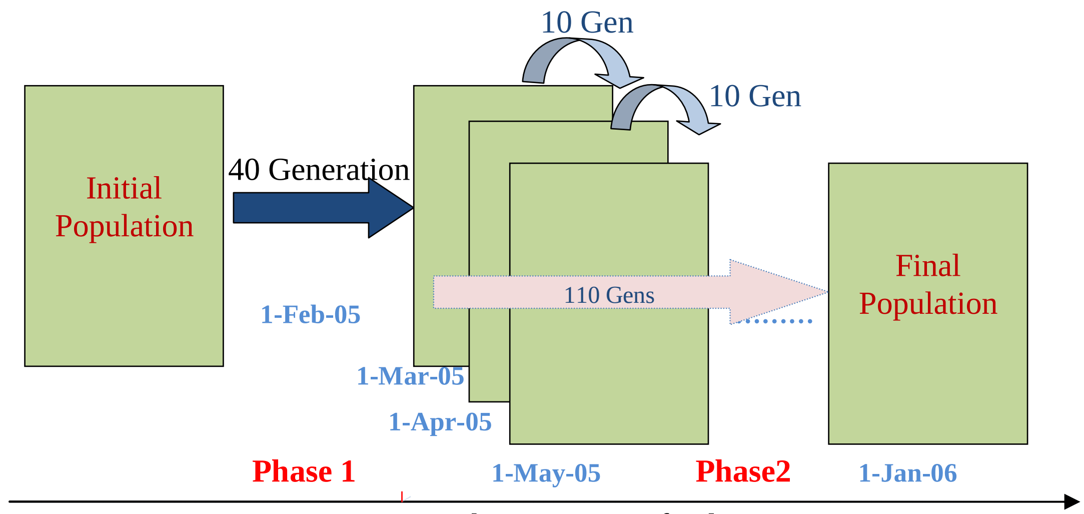


Figure 3.3 Two Phase Process of Adaptive GA

3.2.3 Make a Portfolio from the Final Population

In the previous work, we usually pick one best–fitted individual from the last population of the GA and select the stocks that are resented by '1' in that individual. The weakness of this method is obvious because the valuable information in the other individuals of the final population will be rejected. Also it is unworkable for a real portfolio selection that the number of the selected stocks can possibly vary from very few to one hundred. Thus, to construct a portfolio more efficiently from the last population, we accumulate the number of '1', which represents the selection of each one stock, from each individual of the entire population for every stock, and select the best 10-15 stocks which have the biggest number of '1's to construct our portfolio. This process is illustrated as the Figure 3.4.

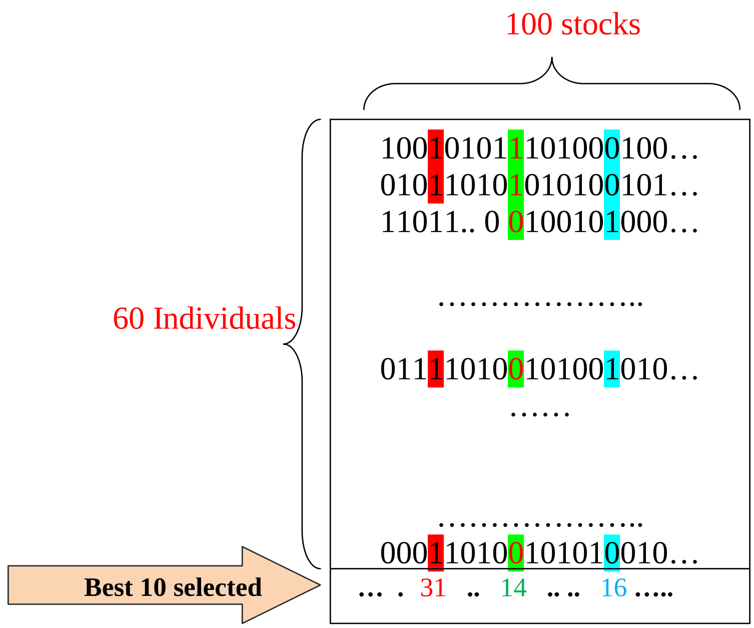


Figure 3.4 Select Portfolio from One Population

3.3 Programming Models and Method Selection

In programming model, two class, individual and population are defined in c++ to represent the population and the chromosome. In the class of individual, there are 3 pubic members and 5 public functions. In the class of population, 5 public members and 10 public functions are included. The initial population will be generated by random function. The operation of crossover and mutation on each generation also relies on randomly generated numbers. In the model development, several key features require practical adjustments.

```
class Individual {
 public:
string   chromosome ;
int    chromosomeLength ;
double fitness;

 void init();
 Individual() ;
~Individual() ;
 double getFitness() ;
void   setFitness(double fitness) ;
void  output();
};
```

```
class Population {

 public:
          int generationRandom;
          int popsize;
          int chromosomeLength;
          float aveFitness;
          vector<Individual > individual;
 Population(int pop_size ,int chrom_length);
~Population();
 void printPopulation();
 void getFitness(float [][10], float[][10], int month);
 void selection();
 void crossover();
 void mutation();
 int  getLargest();
 float get_Profit(float [][10],float[]);
 void getGenerationRandom(int Generation_random);
```

Table 3.2   Definition of Classes

1) Initiation of the population.

To randomly generate N chromosome as the original population, the common idea would be as following:

```
for(i=0; i<=popsize; i++)
  {
        individual_a. init();
        population.add(individual_a);
   }
```

Table 3.3 Population Initiation

In real experiments, because of the limitation of the computer generated random number, if generation time of these individuals are within one second, the program will generate exactly same chromosome each single time, such as Figure 3.5:

Individual 1 : 1000111010010110001101011

Individual 2 : 1000111010010110001101011

....

Individual 30: 1000111010010110001101011

Figure 3.5 Initiation of First Generation 1

To solve this problem, we need to generate a long random binary string in one time, and divide it into N parts, each of which has the chromosome length of N, while N equals to population size, as Figure 3.6.

$$\overbrace{\text{1100101011110001010010100...}}^{\text{Individual 1}}\overbrace{\text{01011101011101000101010101…}}^{\text{Individual 2}}\overbrace{\text{010100…...}}^{\text{Individual 3}}$$

Figure 3.6 Initiation of First Generation 2

2) Implementation of selection, crossover and mutation

a. Selection Methods

After decide how to encode the portfolio and initiation of the first generation, the second decision to make is how to perform selection. The selection method determines the principle to choose the individuals in the population that will create offspring for the next generation, and how many offspring each will produce. Since the goal of the selection is to increase the appearance of the fitter individuals in the population, the selection has to be well-balanced. Too strong selection will cause the highly fit individuals to take over the population and reduce the diversity for future generation, while too weak selection causes too slow evolution.

Two popular methods are implemented and tested in this model. First method is Fitness-Proportionate with "Roulette Wheel", in which each individual is assigned a slice of the "roulette wheel", with the size of the slice being proportional to the individual's fitness. Every time the wheel is spun, the individual under the wheel's marker is selected to be in the pool of parents for the next generation. There will be N times of spin, where N is the size of the population. The size of the slice for each individual is equal to the probability that one chromosome produces its offspring, which is decided by function: $P_i = \frac{Fitness(Ci)}{\sum_{i=1}^{n} Finess(Ci)}$. This method is shown in Table 3.4.

```
sub_sum = 0;
for(i=1; i<=popsize; i++)
{
        individual[i].lower = sub_sum;
        sub_sum += Pi;
        individual[i].upper= sub_sum;
 }
for(i=1; i<= popsize; i++)
{
        r[i]=(rand()%100)/100.0;
        if (individual[m].lower < r[i] < individual[m].upper )
        {
                select individual[m];
        }
}
```

Table 3.4 Fitness-Proportionate with "Roulette Wheel"

The other selection method is Tournament selection. In this selection method, two individuals are randomly chosen from the population, and again a random number *r* between 0 and 1 is then generated to compare with a predetermined parameter *c* (for example, *c*= 0.85). If r < *c*, the fitter individuals is selected to be a parent; otherwise the less fit one is selected. The two are then return to the original population and eligible to be selected again. The table 3.5 describes this process as:

```
for(i=1; i<= popsize; i++)
{
        r[i]= (rand()%100)/100.0;
        k  = rand()% popsize;
        m = rand()% popsize;
        if (r[i] < c)
           select    get_larger( individual[k].fitness ,  individual[m].fitness);
        else
           select    get_smaller( individual[k].fitness ,  individual[m].fitness);
}
```

Table 3.5 Tournament Selection

b. Crossover

After forming a new population by selection process, these new parents are ready to be performed the operator of crossover. In our model, a two-point crossover is applied to add the diversity of the population. First, N/2 pairs(N is the number of individuals) of parents are randomly selected and with probability *Pc*(typically 0.6-0.8), each pair performs crossover to generate offspring. In two-point crossover, two positions are picked at random again and the segments between them are swapped, as shown in Table 3.6.

```
for(i=1; i<= num_pair; i++)
{
        k  = rand()% popsize;              // No repeated k
        m = rand()% popsize;               // No repeated m

        r[i]= (rand()%100)/100.0;
        if (r[i] < Pc)
        {
           start_point  = rand() % chromosomelength;
           end_point   = rand() % chromosomelength;
        for(j = start_point  ; j<= end_point   ; j++)
              {
                 Exchange   individual[k].chromosome[j]
                            individual[m].chromosome[j]   ;
              }
        }
}
```

Table 3.6.Two-point Crossover

c. Mutation

The mutation operation is very similar to the crossover. By the probability of *Pm*, each bit of the individuals can be operated. As the total number of bits in one population is large, the probability *Pm* is around 0.05, which means 20-40 bits will be flipped by each mutation.

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 Test and Results

To test and compare the performance of both the conventional and adaptive GA, We conducted four series of experiments in different aspects 1) How does the fitness value evolves along with the generations; 2) How does the operators affect on the performance of GA; 3) The relationship between the price of single stock and the investment volume on it; 4) Comparison on Rate of Return with conventional and adaptive GA.

1) Fitness value — Number of generation

   This is the basic functional test of a GA model, and we prefer to know both overall performance and best evolving result. Therefore two fitness values need to be tested, which is a) average fitness value of population, b) the best individual fitness value. Figure 4.1 and Figure 4.2 shows that in both conventional and adaptive GA model, the average fitness value increases more smoothly than the best individual fitness value, which indicates that the overall fitness of the population moves well. The best fitness curve is less stable, showing that the random crossover and mutation may cause the unexpected increase and decrease of the best fitness individual.

2) Operators — Fitness value

Figure 4.3 shows the comparison of the two selection methods. The results indicates that the method of Fitness-Proportionate with “Roulette Wheel” works better than method of Tournament Selection, which more depends on the generation of the random numbers.

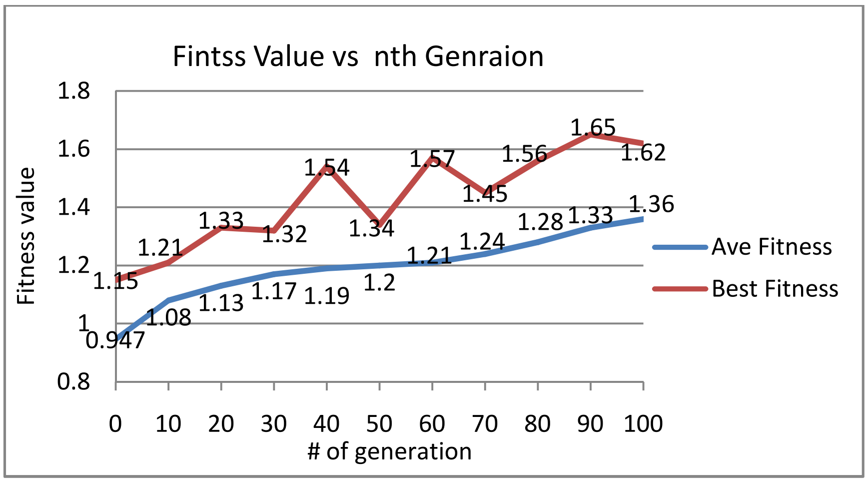


Figure 4.1 Conventional GA- Fitness value Vs # of Generation

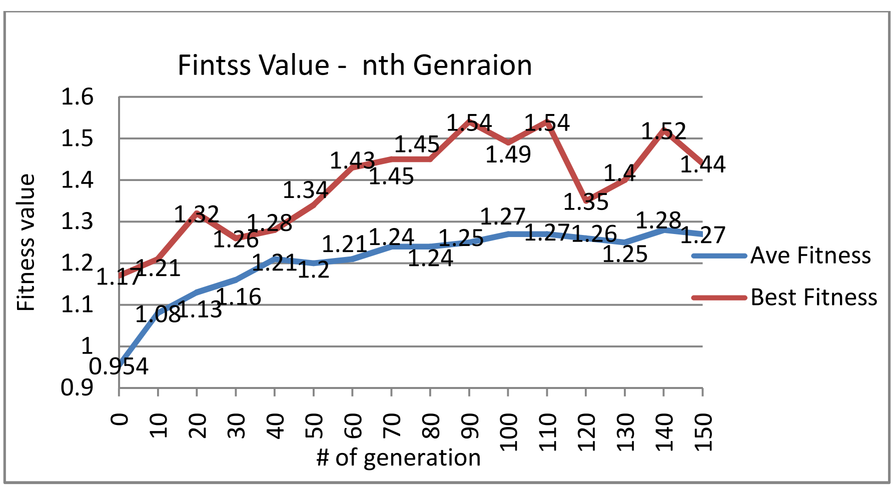


Figure 4.2 Adaptive GA- Fitness value Vs # of Generation

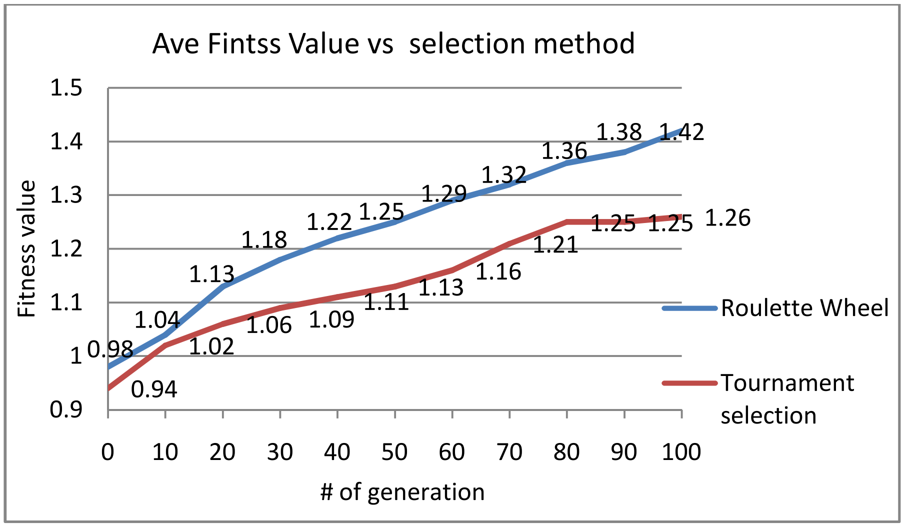


Figure 4.3 Fitness value Vs # of Generation

3) Comparison on Rate of Return between two type of GA models

In this section, we run the test five times for both conventional GA and adaptive GA respectively, from each of which, 10 stocks were selected, and the overall Rate of Return was ranked and compared. In order to evaluate the performance of the models more effectively, the natural Rate of Return of the entire stock pool, which means investing equally in each of 100 stocks, was also computed. The result shown in Table 4.1 indicates that the adaptive GA performed much better in Rate of Return.

Ranking	Conventional GA %	Adaptive GA %	Improvement
1	27.4	33.4	21.8 %
2	17.2	32.5	30.8 %
3	9.1	28.6	201.6 %
4	8.6	12.4	44.2 %
5	3.7	10.1	173.9 %
Average	15.08	23.24	54.8 %
Natural	10.2	10.2	

Table 4.1 Results on Rate of Return

4) Price of single stock and popularity of one stock

As the fitness value of each stock is decided by P_{AVEj}/P_0, the ratio of the 256-day average price of the trading day price, we need to track that how much of the stock price trend can affect the popularity of the stock, which is in one population the number of individuals that has put the stock as 'select'. If we also consider this number of popularity as the amount of hypothetical amount of investment, then a potential profit of stock can be computed to testify weather the GA's decision is reasonable or not. Three stocks are chosen to perform this test, each of which has a different type of price curve. Figure 4.4.1 - 4.4.3 shows the results of Stock labeled 'AAC', American Campus Communities, indicating that the popularity of this stock in one population increases as the evolvement of generations because its price decreases. Thus the potential profit goes up along with the GA's evolvement. The actual selling price of the stock is $0.43, which is on the date of 1-July-2006, half of year later.

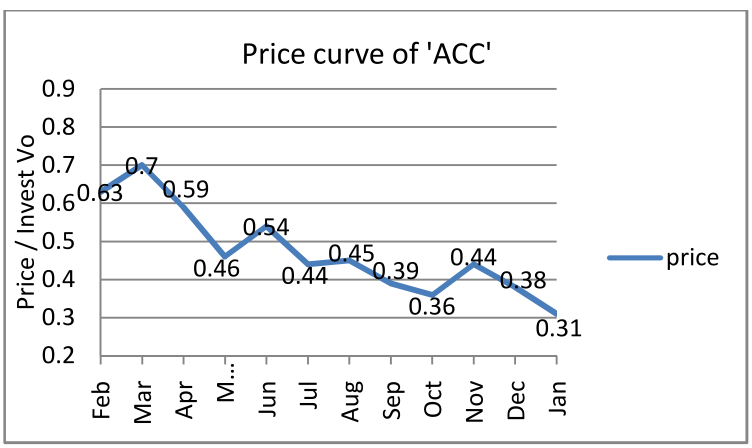


Figure 4.4.1 Price curve of 'ACC'

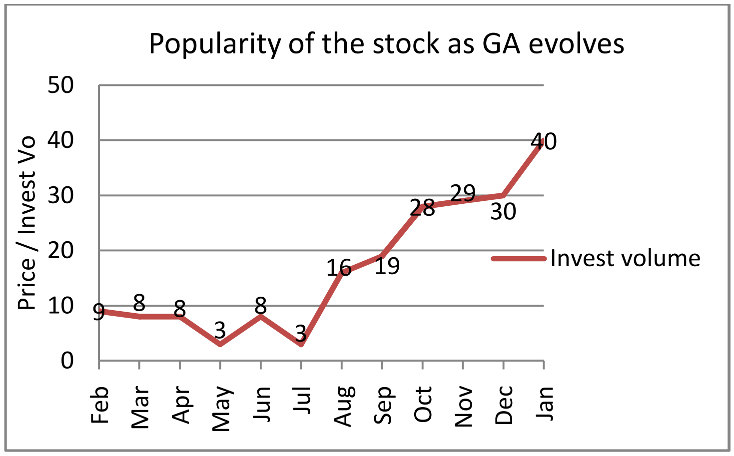


Figure 4.4.2 Popularity of stock of ‘ACC’ as GA evolves. The figure shows that by the month of February, the GA has evolved 40 generations (Figure 3.3), and the popularity of the stock is ‘9’, which means there are 9 individuals in the 40th generation that has put ‘1’ on this stock. After 110 generations evolvement (10 generations each month) , the popularity of the stock rose to ‘40’.

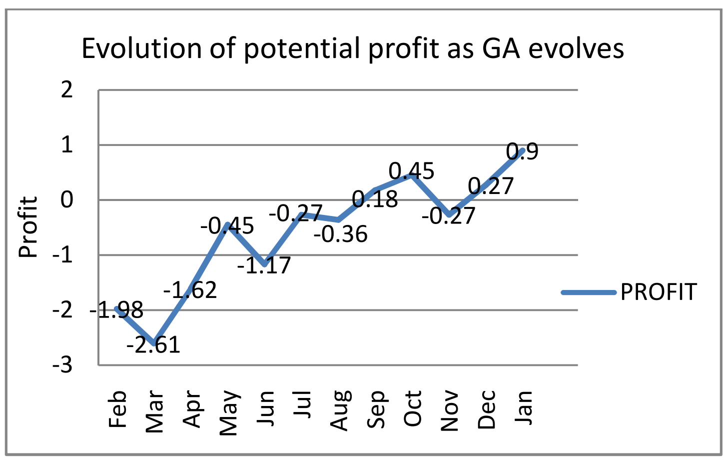


Figure 4.4.3 Evolution of potential profit of ‘ACC’. The figure shows that if we consider stock’s popularity as hypothetical amount of investment, the potential profit increases as the GA evolves.

Figure 4.5.1 - 4.5.3 shows the results of Stock labeled ‘DMC’, Document Security Systems, indicating that when the price increases sharply, the popularity of this stock goes down, meaning that the purchase of this stock is not suggested,

and it will not appear in our selected portfolio as well. The Stock labeled 'ABL' (American Biltrite Inc) has shown more random features of GA. As the price is relative stable, the responding popularity suggested by GA is stable with small amount of random noise; the hypothetical profit then fluctuates from negative to positive, depending on the stock's selling price, shown in Figure 4.6.1 - 4.6.3.

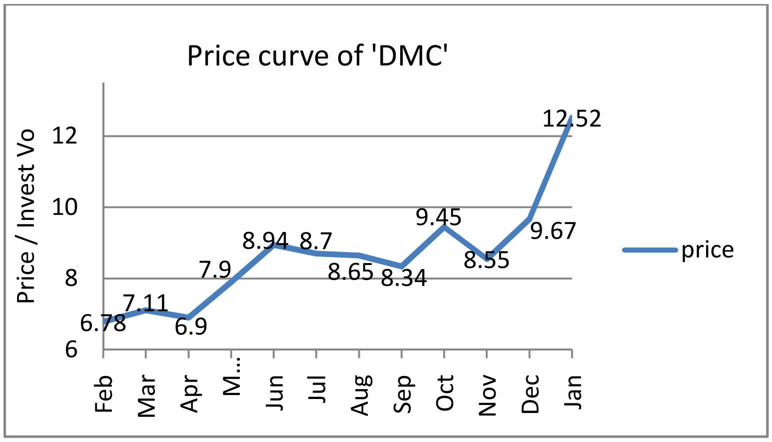


Figure 4.5.1 Price curve of 'DMC'

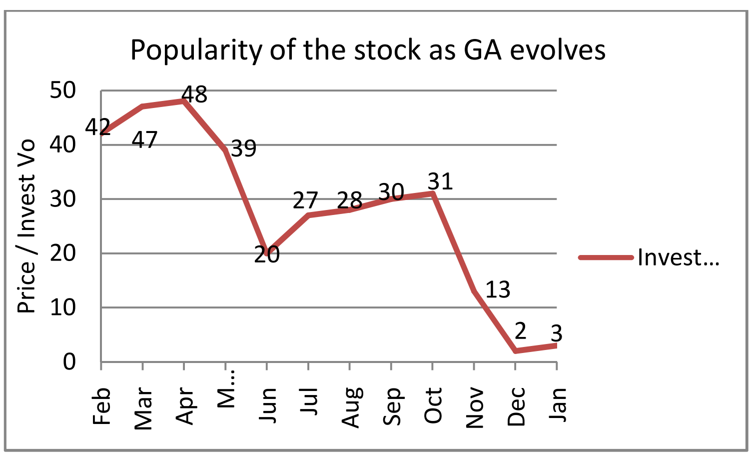


Figure 4.5.2 Popularity of stock 'DMC' as GA evolves. The figure shows that because of the sharp increase of the stock price, the popularity of this stock goes down from '42' to '3' as GA evolves, then the stock 'DMC' will definitely be rejected from our portfolio.

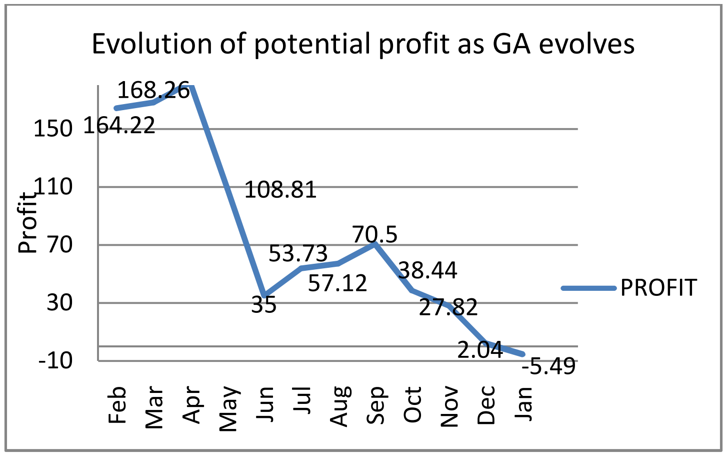


Figure 4.5.3 Evolution of potential profit of 'DMC'. The figure has testified the GA's validity by showing that the potential profit of this stock goes down from '164.2' to '-5.49'.

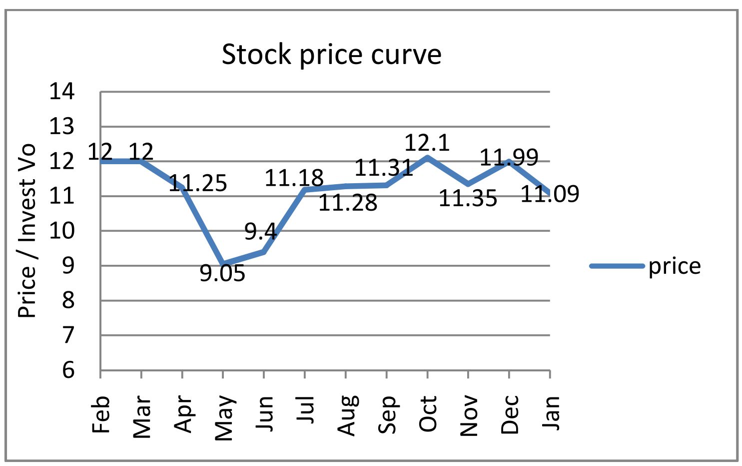


Figure 4.6.1 Stock price curve of 'ABL'

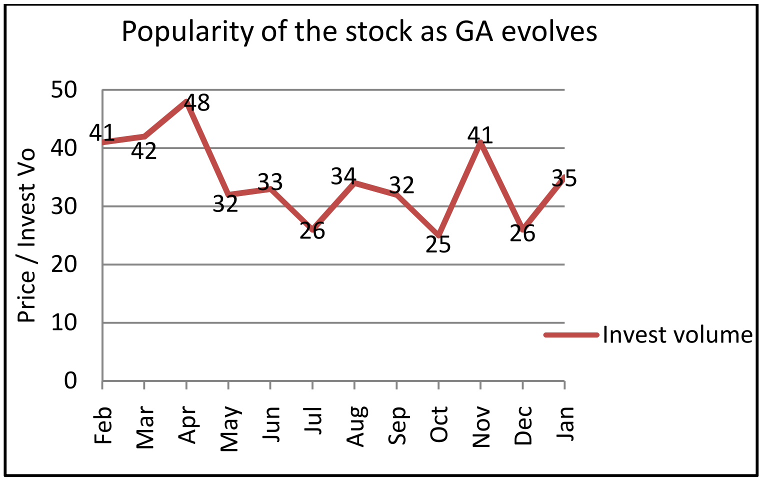


Figure 4.6.2 Popularity of stock 'ABL' as GA evolves. The figure shows that because of the relative stability of price, the popularity yielded by GA move randomly up and down, which indicate that GA is not perfect in evaluating one single stock.

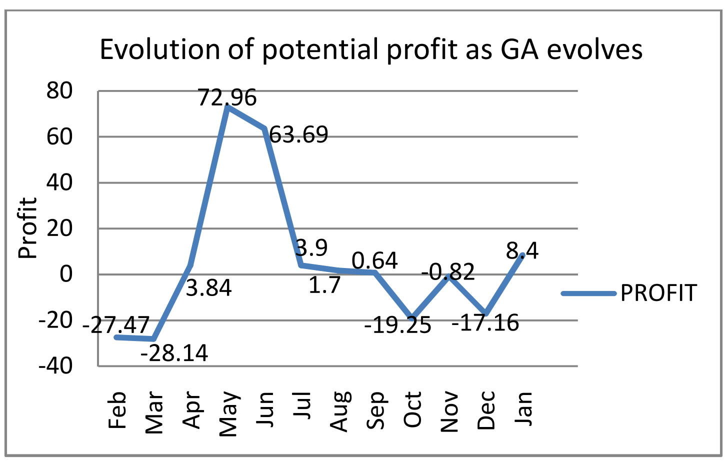


Figure 4.6.3 Evolution of potential profit of 'ABL'. The figure indicates that GA's theory and model still has its randomness and limitation as well.

4.2 Conclusion and Future Work

In this research model, we designed and implemented the conventional GA model first and then proposed an adaptive GA model to construct a financial portfolio out of a

100 stock pool. The experimental results show that the algorithm runs correctly and the population moves to the directions of increasing the fitness value. Generally the adaptive GA has shown better overall Rate of Return than the conventional GA. When we track the popularity of a single stock along with the time period and the generations, we find that the popularity of stock increases as the price goes down normally, and vice-versa. This shows that the basic idea of the stock selecting has been realized by this adaptive GA model. In our future work, improvements and enhancements have to be made in following aspects:

1) The relatively small difference between the 256-day average price and the trading day price of stock results in the slower movement of the fitness value, indicating that some other fitness functions could be applied and tested.
2) Since the adaptive GA that we implemented is only to adapt the price change of each stock, and make modification of the fitness value of the each individual, but not the algorithm itself. The work on this direction should be very interesting and challenging.
3) This GA model can be applied and redeveloped into other financial area, such as option pricing and financial distress predicting.

REFERENCES

[1] Wen-Shiu Lin, Jiah-Shing Chen and Ping-Chen Lin, “A study on Investment Decision Making Model: Genetic algorithms approach”.

[2] M.Srinivas, Lalit M. Patnaik, “Genetic algorithm: A survey”, 1994 IEEE.

[3] Strinivas M, Patnaik L m, “Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms”. IEEE Trans Syst. Man and Cybernetics, 1994.

[4] Wei-Chen, Runtong Zhang and Wei-guo, Zhang, “ Management of stochastic portfolios using improved genetic algorithm”, in proceeding of the 6th world congress on intelligent control and automation, June 21-23, 2006, Dalian, China.

[5] Suleiman K. Kassicieh, Thomas L, Paez, “Investment decisions using genetic algorithms”, 1997 IEEE.

[6] Man-Chung Chan, Chi-Cheong Wong, Bernard K-S Cheung, Gordon Y-N Tang, “Genetic Algorithm in Multi-Stage Asset Allocation system”.

[7] Xia Liu, Hongxia Chu and Kejun Wang, “Studying of Portfolio investment fuzzy decision model based on Genetic Algorithm”, in proceedings of IEEE Intenational Conference on Mechatronics & Automation, Niagara Falls. July 2005 .

[8] Wen-shui Lin Jiah-Shing Chen and Ping-Chen Lin, “A Study on Investment Decision Making Model: Genetic Algorithms Approach” 1999 IEEE.

[9] R.K.H. Galvao ,V.M Becerra and M. Abou-Seada, “Variable selection for Financial distress classification using a genetic algorithm” .

[10] Pei-Wen Huang and Chao-Lin Liu, “Using genetic algorithms for feature selection in prediction financial distresses with support vector machines”, 2006 IEEE Internation Conference on Systems, Man, and Cybernetics October 8-11, 2006, Taibei, Taiwan.

[11] Rui Jiang and K.Y. Szeto, " Extraction of investment strategies based on Moving Averages: A genetic algorithm approach" 2003 IEEE.

[12] Badawy, F.A. Abdelazim, H.Y. Darwish, M.G. Fahima A. Badawy, "Genetic Algorithm for predicting the Egyptian stock market", the New Knowledge Society: ITI 3rd International Conference.

[13] Cyril Schoreels, Brian Logan, Jonathan M. Garibaldi "Agent based genetic algorithm employing financial technical analysis for making trading decisions using historical equity market data", in proceedings of the IEEE/WIC/ACM international Conference on intelligent Agent Technology(IAT' 04)

[14] J.Li and E.P.K. Tsang, " Investment decision making using FGP: a case study," in proceeding of the 1999 Congress on Evolutionary Computation, IEEE Press, Washington DC, USA.

[15] J.Li and E.P.K. Tsang, "Improving technical analysis predictions: an application of genetic programming," in Proceedings of 12th International Florida AI Research Society Conference, Orlando, Florida, 1999.

[16] E. P. K. Tsang and J. Li, "Combining ordinal financial predictions with genetic programming", in proceedings of the second International Conference on Intelligent Data Engineering and Automated learning(IDEAL-2000),Hong Kong.

[17] Jin Li and Sope Taiwo, " Enhancing Financial Decision Making Using Multi-Objective Financial Genetic programming", 2006 IEEE Congress on Evolutionary Computation , Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada

[18] Yan-chun Liu; Liang Sun, "Comparative Research of Portfolio Model Using Fuzzy Theory", Innovative Computing Information and Control, 2008. ICICIC '08. 3rd International Conference on 18-20 June 2008

[19] Po-Chang Ko; Ping-Chen Lin; Yao-Te Tsai, "A Nonlinear Stock Valuation Using a Hybrid Model of Genetic Algorithm and Cubic Spline", Innovative Computing, Information and Control, 2007. ICICIC '07. Second International Conference on 5-7 Sept. 2007

[20] Chang, Jui-Fang; Tsaur, Tien-Wang; Huang, Jo-Chun, “Applying Investment Satisfied Capability Index and Genetic Algorithms to Construct the Stocks Portfolio”, Innovative Computing Information and Control, 2008. ICICIC '08. 3rd International Conference on 18-20 June 2008

[21] Shu-Heng Chen; Who-Chiang Lee, “Option Pricing with Genetic Algorithms: A second Report”, Neural Networks, 1997., International Conference on Volume 1, 9-12 June 1997.

[22] Shu-Heng Chen; Who-Chiang Lee, “Option Pricing with Genetic Algorithms: A second Report : separating out-of-the-money from in-the-money” Intelligent Processing Systems, 1997. ICIPS '97. 1997 IEEE International Conference .

[23] Tung, W.L.; Quek, C, “GenSo-OPATS: “a brain-inspired dynamically evolving option pricing model and arbitrage trading system”, Evolutionary Computation, 2005. The 2005 IEEE Congress.

[24] Jin Li and Sope Taiwo, “Enhancing Financial Decision Making Using Multi-Objective Financial Genetic programming”, Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference

[25] M H Lim, D Wunch and K W Ho ,“An evolutionary programming methodology for portfolio selection ”, Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference.

[26] Shigeo Mori, Kotaro Hirasawa and Jinglu Hu, “A stock price prediction model by using genetic network programming” , SICE Annual conference in Sapporo August 4-6, 2004 .

[27] Alma Lilia Garcia-Almanza and Edward P.K. Tsang, “Simplifying Decision Trees learned by genetic programming”, Management Science and Engineering, 2007. ICMSE 2007 International Conference.

[28] Nils Svngard, Peter Nordin and Stefan Loloyd, “ Using Genetic Programming with Negative Parsimony pressure on exons for portfolio optimization” 2003 IEEE.

[29] Irwin Ma, Tony wong, Thiagas Sankar, “ Forecasting the volatility of a financial Index by wavelet transform and evolutionary algorithm” , 2004 IEEE International conference on systems , Man and cybernetics.

[30] Hayward, S, "Setting up performance surface of an artificial neural network with genetic algorithm optimization": in search of an accurate and profitable prediction of stock trading" , Evolutionary Computation, 2004.

[31] Orito, Y.; Yamazaki, G, "Index fund portfolio selection by using GA" , Computational Intelligence and Multimedia Applications, 2001. ICCIMA 2001. Proceedings. Fourth International Conference.

[32] Iba, H.; Sasaki, T.; "Using genetic programming to predict financial data" , Evolutionary Computation, 1999.

VITA

YI DENG

Candidate for the Degree of

Master of Science

Thesis: Genetic Algorithms for Financial Portfolio Selection

Major Field: Computer Science

Biographical:

Personal Data: Born in Sichuan, China, and February 7, 1981.

Education:
Completed the requirements for the Master of Science degree with major in Computer Science at Oklahoma State University, Stillwater, Oklahoma in July, 2008.

Bachelor of Computer Science, University of Science & Technology Beijing, Beijing, China, 2002

Experience:
2005-2007 Graduate Teaching Assistant, Computer Science Department, Oklahoma State University.
2003-2004 Software Developer, R&D department, SINA.COM Corporation.

Name: Yi Deng Date of Degree: December, 2008

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: Genetic Algorithms for Financial Portfolio Selection

Pages in Study: 39 Candidate for the Degree of Master of Science

Major Field: Computer Science

Scope and Method of Study: In this thesis, we explore and classify the applications and improvements on GA in these researches, then design a conventional GA model which applies classic GA mechanism to select a portfolio from a 100 AMEX stock pool. Because the generations of this conventional GA model only evolves based on one fixed data segment, the results it yields are not stable enough for the extremely dynamic financial market. In order to keep the populations more dynamically reflecting the current market circumstance, we propose an adaptive GA model to improve the convention model. The enhanced model will only evolves certain number of generations using one segment of history stock data and change to another data segment. We implement both of the conventional and adaptive GA models and compare their performances using data of one same stock pool. The experimental results have shown that the adaptive GA model is more reliable and gains higher average Rate of Return. In addition, alternative GA operators are implemented and tested in the adaptive model to find the optimal solution to improve the Rate of Return.

Findings and Conclusions: (1) In both conventional and adaptive GA models, the average fitness value of each generation increases along with GA evolvement, which shows that GA has worked generally correctly. (2) In the two alternative selection methods that have been implemented in conventional GA models, "Roulette Wheel" works more effectively than "Tournament Selection". (3) Basically, the adaptive GA model has higher average ROR that the conventional GA because of its longer evolvement circle and better utilization of stock price information. (4) Our adaptive GA model is more effective in make a decision on stocks have a steep price curve, while it shows weakness and randomness in evaluating the stocks which has a relatively flat curve.

ADVISER'S APPROVAL: Xiaolin Li