VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY

INTERNATIONAL UNIVERSITY

DEPARTMENT OF MATHEMATICS

GRADUATION THESIS

PORTFOLIO OPTIMIZATION USING PARTICLE SWARM OPTIMIZATION METHOD

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING in

FINANCIAL ENGINEERING AND RISK MANAGEMENT

Students Name: Pham Ky Thoai

Students ID: MAMAIU13076

Thesis Supervisor: Dr. Le Nhat Tan

Ho Chi Minh City, Vietnam

July, 2017

Acknowledgements

First of all, I would like to express my deep-felt gratitude to my advisor, Dr. Le Nhat Tan of the Mathematics Department at International Univeristy - Vietnam National Univeristy in Ho Chi Minh City, for his advice, encouragement, enduring patience and wholehearted support. Although I used to want to give up several times due to the thinking that this topic is beyond my ability, he was never ceasing in his belief in me and always generated a great source of motivation for me by words of encouragement. When I got stuck in some problems, he always provided clear explanations and new directions to approach these problems. When I neglected my thesis because of some personal reasons, he always reminded me to focus on doing my thesis to keep up with schedule. Moreover, he also taught me to how to write thesis in a logical and coherent structure, and also supported me in writing MatLab codes for the optimization algorithms. Without his help and support, I do not think that I could totally finish my thesis on time. I wish all students have the honor and opportunity to experience his abilities in this job.

Secondly, I would like to thank my Portfolio Management Teacher, Associate Prof. Vo Thi Quy of the School of Business at International Univeristy - Vietnam National Univeristy. She had a big influence on decision of choosing my thesis topic. Experiencing more than 13 weeks on her course, she gave me the theoretical as well as practical knowledge about Portfolio Management, she also provided me the basic definitions which are commonly used in Investment Management. She also required me to carry out a small project which is to construct an optimal portfolio using 5 stocks from VN-INDEX, as a great opportunity for applying all theoretical knowledge into reality. Thanks to the knowledge I gained from her course as well as her wholehearted support, I am totally attracted to the field of Portfolio Managment. I have found a lot of interesting things in this field. Therefore, I decided to choose the topic of my thesis involving in this field.

Additionally, I want to thank my Financial Mathematics Teacher, Dr. Nguyen Tien Dung of Ho Chi Minh City University of Technology - Vietnam National Univeristy, for his supports when I had some big difficulties in drawing the efficient frontier curves. He also gave me some valuable advice that is really useful for me while doing my thesis.

Moreover, I want also to thank Associate Prof. Nguyen Dinh and Associate Prof. Nguyen Ngoc Hai of the Mathematics Department at International Univeristy - Vietnam National Univeristy, for the knowledge of optimization field. They gave me a solid foundation of optimization algorithms and the practicla knowledge of these algorithms. Thanks to them, I have found many interesting and useful things in this field.

Also, I want to thank my best friend Mr. Le Chi Nhuan of Vinh University for everything he did to me. He not only supported me with the programming part, but also gave me a valuable moral support whenever I was down.

And finally, I must thank my father, my mother, my sisters and my brother for putting up with me during the development of this work with loving support and no complaint. Without them, this paper would not have been possible. I do not have the words to express all my feelings here, only that I LOVE MY FAMILY.

Abstract

One of the most studied issues in the financial investment management is portfolio optimization. In reality, portfolio optimization problems often contain various kinds of non-linear and linear constraints, as well as multi-objective functions. The traditional methods cannot solve these problems efficiently. Therefore, in this paper, we present a technique of meta-heuristic algorithm, named Particle Swarm Optimization (PSO) to portfolio optimization problems. The PSO method is tested on six investment portfolios including three unrestricted portfolios and three restricted portfolios, and a comparative study with Genetic Algorithm and Excel Solver has been conducted. The PSO method demonstrated a great efficiency in computational perfomance in building optimal risky portfolios. Preliminary results show that this techique will be a promising method and will obtain results which are comparable or superior with the state of the art solvers

Table of Contents

Chapter 1

Introduction

Even though many people have probably heard about Portfolio or Portfolio Optimization, I do not think a lot of them know what Portfolio is or what Portfolio Optimization is.

In order to understand more easily, we can link the Portfolio Optimization to a real life case. Suppose that a man had one billion Vietnamese dong in his bank account, and he decided to withdraw his money from the bank and put it in a safe deposit box in his house. One day, when he stayed overnight in his friend's house due to a personal reason, a thief broke into his house and opened the safe deposit box and took his money away. When he came back, he realised that he lost all his money as well as other important things. One day, he decided to told this story to his best friend, and his best friend told that "Never put all your money in the one place when you have the large amount of money". What his friend wanted to advise is that he should divide his money into many small parts and put each part in each different place such as bathroom, living room and bedroom instead of putting all money in a safe deposit box. If he had done this, he could have saved at least a little of money in the case when a thief broke into his house and stole his money. By putting his money in many different places, he could have reduced the probability of losing all his money in the case of a steal.

In reality, you can minimize the risk of losing all money by splitting your money. In other words, you can minimize the risk of losing all money by diversifying the places where you put money. That is one of the fundamental principles of Portfolio Management or Portfolio Optimization. In some cases, you can get rich too by applying this.

Many years ago, Portfolio Optimization is something which is just for big financial institutions and companies. However, with the considerable development of technology and the internet, the trading of financial assets is available to many people these days. Further, the financial market which has experienced considerable changes in recent years make Portfolio Optimization become more highly complicated than it used to. In this environment, financial research and development in computational tools have also flourished. As a result, more and more methods were applied in Portfolio Optimization and the computational tools could perform on the greater amount of data with more precision.

In this paper, we will solve the Portfolio Optimization Problem by means of Particle Swarm Optimization. This problem has received many attention of researchers, especially those who research on Evolutionary Algorithms. Although many papers about this problem has been conducted before, it still has many scope for improvement. This paper will present the basic principles and models in Portfolio Optimization. Finally, we will mention about our new method to solve this problelm and make a comparision our results with other approaches.

1.1 A Portfolio

A portfolio is simply a collection. It can be a collection of any kind of stuff. Usually, a portfolio mentions a group of photographs, drawings and documents of people working in art field or a group of investment assets in finance field.

In finance, a portfolio is a collection of some financial assets such as stocks, bonds and derivative securities and so on, and it is often held by investors, investment companies and financial institutions. Depending on the risk tolerance and investment objective of each investor, each investment portfolio is constructed differently. Nowadays, many investors consider managing their investment portfolio as managing their primary sources of income, therefore, they are willing to invest a large amount of time in researching on the financial market. Besides the profitability which is generated by portfolio management, a portfolio can be exposed to the large amount of risk if it is not managed effectively.

Although there is no rule or restriction in the number of asset types in a portfolio, investing all money in an individual hold is not a good choice. The continuous changes of the financial market result in the fact other securities show different behaviors to these changes and also have different characteristics such as risk, profitability and liquidity. Therefore, nowadays, investors try to create the diversification for their portfolios to reduce the risk resulting from the changes of the market. A diversified portfolio means that its assets have to be diversified with different companies and different industries. Many types of securities can be used to construct a diversified portfolio, but stocks, bonds and cash are considered as the core building blocks of a portfolio.

What we mention above is the strategy to manage a long-term portfolio (more than 1 year of investment) in which investors try to include many types of assets to create the diversification. The more diversification is, the greater risk is reduced. However, for short-term purposes, investors often construct portfolios with a few financial assets to gain short-term profits. This can help investors gain profit in a short period of time, but this is also a high-risk investment which is only suitable for investors with high risk tolerance.

1.2 Portfolio Optimization and Particle Swarm Optimization Algorithm

When an investor makes an investment decision, he will cope with a lot of problems. Some of these are:

1. He has to decide how many assets he should invest in his portfolio. The change in the number of assets during investment time will make him bear extra expenses, because buying and selling of investment assets in any time will incur transaction costs.

2. He also has to decide which asset to invest in. He cannot construct portfolios containing all types of assets since there are thousands of investment assets available in the financial market. Therefore, selecting the assets for the portfolio is a really complicated task for investors.

3. Suppose that he has 1 billion VND for his investment and he has already decided the number of assets and the types of asset to invest in. Then, another emerging problem is that he has to decide how to allocate 1 billion to assets in his portfolio efficiently. Designing the asset allocation for portfolio to minimize risk as well as to maximize the return is not an easy task, which requires the academic knowledge from many fields such as Mathematics and Computer Science which a few of investors have.

4. Even if he has successfully constructed a portfolio, he also need to change and update it to follow the movements of the market. The financial market keeps changing every time. As a result, the characteristics of risk and return of all financial assets also keep changing. Therefore, after successfully constructing a portfolio, he also need to change and update it following the variations of risk and return to satisfy his initial expectation.

The problem in the third case is called the Portfolio Optimization Problem which is a problem concerning the allocation of asset and diversification in order to maximize return for a given amount of risk or to minimize risk for a given amount of return. This problem is raised to answer the question "How to distribute money across the available assets in a portfolio to satisfy the objectives and constraints of investors?". Solving this problems will give you each weight which is correspond to each asset in your porfolio.

In 1952, Harry Markowitz introduced a parametric optimization model for the asset allocation and diversification problem with the objective to minimum risk with the given amount of risk and vice versa in the paper "Portfolio Selection" which is published by *Journal of Finance*. This model is called The Modern Portfolio Theory, Mean-Variance Model or Markowitz Theory. Markowitz model is considered as a standard model for portfolio optimization.

Nowadays, MPT plays an important role in forming the basic of all portfolios. However, this model is based on a lot of assumptions. Some of these are as follows:

1. All investors are basically risk averse.

2. All investors pay no taxes on returns and no transaction costs.
3. All investors have an accurate conception of possible returns.
4. All investors have access to the same information at the same time.
5. All investors are price takers.
6. All investors can lend and borrow an unlimited amount at the risk-free interest rate.

Although MPT has revolutionized the knowledge of people about portfolio management and has been widespreadly considered as a useful tool for portfolio optimization, including many assumptions makes this theorem not easy to apply into reality. In the standard portfolio optimization problem, the considered constraints are budget and no right to short sell. However, portfolio optimazation problems in reality are under many real constraints such as transaction costs, cardinality and tax expenses. Therefore, if you try to remove each of these restrictions in the standard problem to make it suitable to the real world, this model will become more and more highly complicated and, of course, constructing an optimal portfolio will be more difficult.

In the past, some numerical methods such as Quadratic Programming Method is used to dealing with portfolio selection problems. This method can work well in case that the number of assets is less than 15. However, because of the real-word requirement, portfolios in reality often contain more than 15 assets. As a result, the portfolio optimization problems become more complicated and it is really difficult to solve these problems by applying this method.

Moreover, the advancement as well as complexity of the financial market these days result in the strict requirements in addition to large size such as the limitation in running time and the limitation of precision in estimating instance parameters, which make analytical and traditional methods no longer handle portfolio optimization problems efficiently under these requirements. Therefore, researchers have to use heuristic techniques which can achieve successfully high-quality solutions of high-dimensional problems in a short period of time.

Because of the complicatedness and limitation in computation time of portfolio selection problem, meta-heuristic algorithms will be a good choice to meet satisfy those requirements. Some studies about meta-heuristic techniques such as Genetic Algorithm, Simulated Annealing and Tabu Search have been conducted to tackle efficient frontier problem [1]. Pareto Ant Colony Optimization [2] has been proved that it is an effective meta-heuristic for dealing with the portfolio selection problems by making comparisons of its performance with Pareto Simulated Annealing and Non-Dominated Sorting Genetic Algorithm. An artificial neural network model with the Particle Swarm Optimization Algorithm [3] has been used in portfolio management and indicated the advantages of hybrid models. The combination between Fuzzy Analytic Hierarchy Process (AHP) [4] and the portfolio selection problem has been applied to model the uncertain environments. A hybrid Genetic Algorithm approach [5] has been studied to track the Dutch AEX-index.

On the other hand, there are some disadvantages for applying these methods into solving the portfolio selection problem. For instance, Fuzzy approach usually has a lack of learning ability [6], Artificial neural network approach always is often faced with over-fitting problems and is easy to trap into local minima [7]. Although Genetic Algorithm (Alba and Troya, 1999) is an efficient algorithm to tackle the highly difficult and big problems, it requires an increase in the time to converge for finding satisfactory solutions.

In order to overcome these disadvantages, PSO method has emerged to solve portfolio optimization problem. PSO method, which was first introduced by James Kennedy and Russell Eberhart in 1995, is a population-based stochastic optimization technique and also an alternative solution to the complex non-linear optimization problems. The basic idea of this algorithm is initially inspired by simulations of the social behavior, as a representation of the move of organisms in a swarm such as a fish school, ant colony and bird flock. It is based on the natural process of communication as a group, even though each individual in a swarm possesses no knowledge to determine the best position, they can solve highly complicated problems as a group by communicating each other in search space.

In comparison with common optimization algorithms, PSO is a more powerful method since it is really efficient in finding the optimal solution by global search. One of the important things of portfolio selection and optimization problem is sub-optimal solution. While PSO can help users to find that solution, dynamic programming cannot.

The main task of this thesis is to employ a PSO method for portfolio optimization problem.

1.3 Outline of Thesis

In Chapter 2, we will review the knowledge of Markowitz Model or Modern Portfolio Theory, including Diversification, Specific and Systematic Risks, Mathematical Model and Efficient Frontier. We also dicussion about Sharpe Ration, Constrained Optimization, Mean Variance Analysis of Risk and Return and Short Sale to help readers to understand fully everything when moving to next chapters.

In Chapter 3, we will discuss about models for Portfolio Optimization. We will present three models which are used to construct an optimal risky portfolio, including: Markowitz meanvariance, Efficient Frontier, and Sharpe Ration Model.

In Chapter 4, we will discuss about background of PSO and how to choose the values for input parameters of PSO. Moreover, we will give an real illustration for PSO method and test PSO method by using test functions of optimization. Finally, we will show the application of PSO which is discoverd from the past to now.

In Chapter 5, we will show how to apply PSO method into solving the constrained portfolio optimization problems. We will show how to set up fitness function, movement, parameter selection and constraint satisfaction for PSO method.

In Chapter 6, in order to test the efficiency of the proposed PSO method, we will give simulations of PSO method on six portfolios and comparative studies with Genetic Algorithm, Excel Solver and Lagrange Method.

In Chapter 7, we will draw final conclusions and will propose some works for future research.

Chapter 2

Literature Review

2.1 Markowitz Model: Modern Portfolio Theory (MPT)

The Modern Portfolio Theory was first introduced by Harry Markowitz and he was awarded a Nobel Prize for this theory. Nowadays, this theory still remains one of the most influential theories in finance and also form a basic background for almost all strategies of portfolio management. Constructing a portfolio with no risk is impossible in reality, MTP said that the specific risk of portfolio can be rejected by diversification.

2.1.1 Diversification

In finance, diversification means that the risk of an investment portfolio can be considerably reduced by including a variety of assets in that portfolio. In other words, it is the way of containing many different investment assets which not only from different companies but also from different industries.

The logic behind this is that once a company or a industry experiences a decrease, which does mean that all other companies or industries decline too. There are still many companies or industries perform well. Therefore, in a diversified portfolio, the loss resulting from the decrease in prices of some stocks can be compensated by the profit resulting from the increase in those of other stocks. However, the MPT indicated that specific risk of a portfolio can be totally reduced by diversification while systematic risk cannot.

2.1.2 Specific and Systematic Risks

Market risk or Systematic risk is the risk which affects the overall market. It cannot reduced by diversification. Two instances for this risk are interest rates recessions and a war.

Specific risk or Unsystematic risk is the type of uncertainty due to company or industry specific hazard. According to MPT, this risk can be reduced or cancelled by diversification. In other words, this risk will be lost if you include many assets from different companies and different industries.

The more diversified portfolio is, the less the risk of each asset in portfolio contribute to the portfolio risk. Hence, we should hold a well-diversified portfolio instead of investing all money in an individual.

2.1.3 Mathematical mode

Under the MPT, the rate return of a portfolio is the weighted average of the rates of return for the individual investments included portfolio. The weights are the proportion of total value for the investment.

The computation of the porfolio return can be generalized as follows:

$$R_p = \sum_{i=1}^{n} W_i R_i$$

where:

R_p is the rate of return of portfolio i.

W_i is the weight of the portfolio in asset i

R_i is the rate of return of asset i.

The return of the portfolio using matrix notation is:

$$R_p = W^T R = \begin{bmatrix} W_1 & W_2 & W_3 & \dots & W_n \end{bmatrix} \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ \dots \\ R_n \end{bmatrix}$$

The Variance of Portfolio. According to Martkowitz, the variance of a portfolio is the sum of the weighted average of the variances of each asset in portfolio and the weighted covariances of all assets included in portfolio. The variance of a portfolio includes both the variances of the individual assets and the covariances of all individual 2 assets in portfolio.

The general formula for the variance of the portfolio is given as follows:

$$\sigma_p^2 = \sum_{i=1}^{n} W_i^2 \sigma_i^2 + \sum_{i=1}^{n} \sum_{j=1}^{n} W_i W_j Cov_{ij}$$

where:

σ_p^2 is the variance of the portfolio.

σ_i^2 is the variance of rates of return for assets i.

W_i is the weights of the individual assets i in the portfolio, where weights are determined by the proportion of value in the portfolio.

Cov_{ij} is the covariance between the rates of return for assets i and j.

The variance of portfolio using matrix notation is:

$$\sigma_p^2 = W^T \sum W = \begin{bmatrix} W_1 & W_2 & W_3 & \dots & W_n \end{bmatrix} \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} & \dots & \sigma_{1n} \\ \sigma_{12} & \sigma_2^2 & \sigma_{23} & \dots & \sigma_{2n} \\ \sigma_{13} & \sigma_{23} & \sigma_3^2 & \dots & \sigma_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \sigma_{3n} & \dots & \sigma_n^2 \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ \dots \\ W_n \end{bmatrix}$$

And the standard deviation of the portfolio as follow:

$$\sigma_p = \sqrt{\sum_{i=1}^{n} W_i^2 \sigma_i^2 + \sum_{i=1}^{n} \sum_{j=1}^{n} W_i W_j Cov_{ij}}$$

Covariance of Returns is a measure of the degree to which returns of 2 assets move together relative to their mean returns overtime. A positive covariance implies that returns of these 2 assets move in the same direction relative to their mean returns. In contrast to this, a negative covariance means that the returns of these 2 assets move in opposite direction relative to their mean returns.

For 2 assets i and j, the covariance of return is as follow:

$$Cov_{ij}{=}E[R_i - \mu_i][R_j - \mu_j]$$

2.2 The Efficient Frontier

Let's look at the Figure 2.1 In comparisons of portfolio A with B, C and D, portfolio A

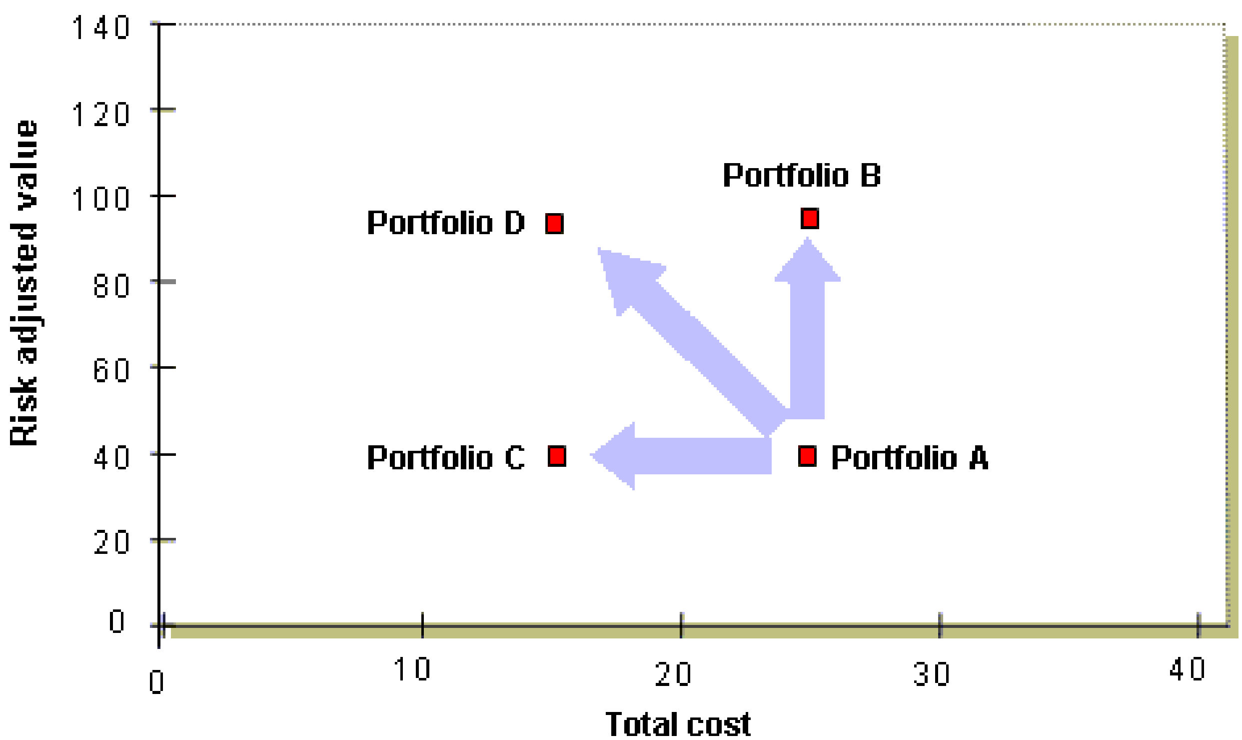


Figure 2.1: Portfolios with different costs and values (www.prioritysystem.com)

is considered as an inefficient portfolio. Although A and C have the same level of risk-

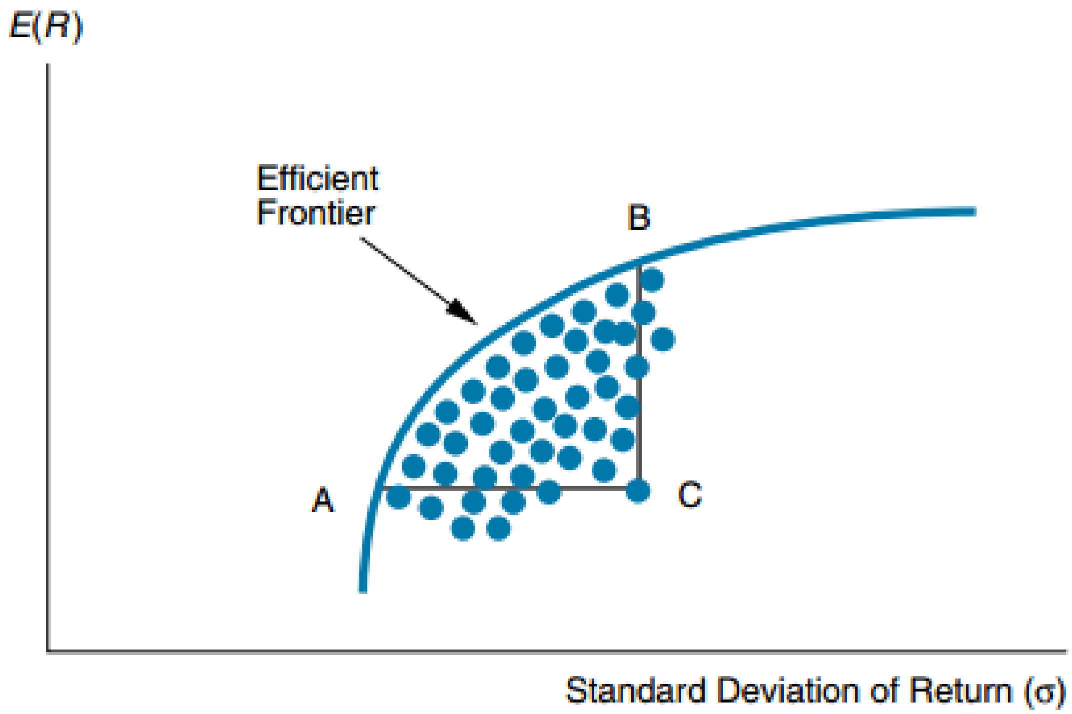


Figure 2.2: Efficient Frontier For Alternative Portfolios

adjusted value, C has lower level of cost. Similarly, even though B and A have the same level of cost, B can gain a higher level of risk-adjusted value.

In comparisons of portfolio D with A, B and C, D is considered as an efficient portfolio. D and C have the same cost, D produces the higher level of risk-adjusted value. Similarly, D and B have the same level of risk-adjusted value, D has the lower level of cost.

Therefore, Harry Markowitz and other economists invented the concept of the efficient frontier which is the curve containing all the best possible asset combinations. In other words, the efficient frontier represents the set of portfolios having the maximum return for the given level of risk, or the minimum risk for the give level of return.

Looking at Figure 2.2. The x-axis is the standard deviation of portfolio and the y-axis is the return of portfolio. Suppose that each point lying in the region of hyperbola is a potential portfolio and the region covering all these points is call a feasible set. In the region covered by hyperbola, for each investment portfolio with a constant level of risk, if we move upwards, we will get a better level of return. Similarly, for each investment portfolio with a constant level of return, if we move leftwards, we will get a lower level of risk. Therefore, the closer portfolios are on the left side and the upward side, the better portfolios are. A combination of standard deviation and return is called efficient if there are no other combinations having a higher return for a the same amount of risk and vice versa.

The set containing all efficient combinations is the efficient frontier and all portfolios lying on this curve are called efficient portfolios. A efficient portfolio exists if and only if there no other portfolios having higher return with the same amount of risk and having lower risk with the same amount of return.

2.3 Sharpe Ratio

In 1966, William Sharpe first introduced the Sharpe Ratio, and this ratio has been one of the popular measures in finance. Nowadays, it is a widely practical tool to compare the portfolios.

Suppose that an investor tells you that he can help you to construct either a portfolio having a return of 80% and a risk of 100% per year or a portfolio with a return of 60% and a risk of 75% ,and you are just allowed to choose one of them. Actually, it is really hard to make a decision on these options because the first portfolio has higher return than the second one while the second portfolio has lower risk than the first one.

Therefore, the values of risk and return of portfolios are not enough to make comparison this portfolio with other ones. Fortunately, Sharpe Ration has emerged and allowed us to solve this. Sharpe Ratio is calculated as follows:

$$\text{Sharpe Ratio} = \frac{R_p - R_f}{\sigma_p}$$

Where:

R_p is the return of the portfolio.

R_f is risk free rate.

σ_p is the standard deviation of the portfolio.

Sharpe Ratio indicated how much excess return you are receiving for extra risk that you incur during time of investment, or it shows the trade-off between risk and return of a portfolio. Depending on Sharpe Ratio, you can determine which portfolio is better. A higher value of Sharpe Ratio means that portfolio provides investors with the higher return for the lower amount of risk. Or, a lower value of Sharpe Ratio indicates that portfolios provides investors with the lower return for the higher amount of risk. Therefore, it is clear that the higher value of Sharpe Ration of portfolios are, the more desirable these portfolios are.

2.4 Mean Variance Analysis of Risk and Return

Although there are some limitations and constraints which are included in the MPT, this theory is still the background for all strategies of constructing portfolios. To use this model for our problem, we have to extract something such as mean return and standard deviation from the data set.

The estimation of mean return and risk (standard deviation) is very important. The more accurate these estimations are, the more realistic and feasible the portfolio which is developed by us will be. Nowadays, there are a variety of ways for estimating these. However, in this paper, we will mention the most popular methods and will use this method for calculating these two estimations.

2.4.1 Return Analysis

One of the popular technical indicators for calculating the return is Moving Averages (MA). There are some different forms of MA. However, the basic motive of all of these forms is similar. The simplest form of MA is Simple Moving Averages (SMA). The SMAs are simply to put the arithmetic average of a set of values or numbers. In order to use them, we just do the following steps:

1. Calculating the return for a short period of time (daily/weely/monthly/yearly).

2. Simple Moving Averages is implemented.

In finance, the closing price of securities or assets is usually used. It can be expessed as follows:

$$R_d = \frac{P_t - R_{t-1}}{R_{t-1}}$$

Where:

R_d is the daily return.

P_t is the closing price of the asset on the t and P_{t-1} is the closing price of the asset on the t - 1 (previous day t).

When we have a set of daily return, we take the SMAs over period of time by using the following equation as follows:

$$R_{dt} = \frac{R_{d1} + R_{d2} + R_{d3} + \ldots + R_{dn}}{N}$$

Where:

R_{dt} is the daily mean return for a period of time t (t time have n days).

N is the number of days in time t.

R_{d1} is the daily return for day 1.

R_{d2} is the daily return for day 2.

R_{d3} is the daily return for day 3 and so on.

Note that, we can also calculate the weekly, monthly or yearly mean return for a period of time t easily by choosing closing prices suitably. However, daily, monthly and weekly mean return are widely used in finance and the period of time t is often greater than 1 year.

One of the major drawbacks of SMAs is a time lag for the estimations because SMAs use the historical data for estimating. To solve this problem, weighted moving averages and exponential moving averages could sometimes be used. However, in this paper, we use SMAs for calculating the mean return.

2.4.2 Risk Analysis

Risk is the volatility of returns. Usually, variance or standard deviation are risk measures. Risk is also estimated by using the historical prices of assets. After having returns, risk estimation will show how much the returns fluctuate over a period of time. The variance is given as follows:

$$\sigma_R^2=\frac{1}{N-1}\sum_{i=1}^{N}(R_i-\bar{R})$$

Where:
R_i is the return at time i.
$\bar{R}$ is the average mean return over the period of time.

After we get the risk of each asset, we need to estimate the total risk associated with portfolio. It can be estimated as the weighted covariance between all assets in portfolio, its formulation is given in Section 2.1. However, that equation is for portfolios with 2 assets. In reality, the number of assets in portfolio is really large, which makes calculating the covariance matrix is more complicated.

2.5 Constrained optimization

Constrained Optimization or Constraint Optimization is the set of numerical methods applied to handle problems where their objective functions are to optimize something with the given constraint, limitation and requirement. If the objective function is the cost function, it needs to be minimized in order to reduce the cost as much as possible. However, if the objective function is reward function, it needs to be maximized in order to gain the reward as much as possible.

In business and finance, the objective functions which are to be minimized typically is the risk function where the risk varies depending on many different factors such as liquidity, dividend and political situation. Moreover, it is typically used to maximize if the objective function is the return function where the return is influenced by many things such as interest rate, commodity price and employment situation.

Constraint of constrained optimization can be an arbitrary Boolean combination of equations such as $g_j(x) = 0$ is an equality constraint, $g_j(x) \geq 0$ is a weak inequality constraint and $g_j(x) > 0$ is a strict inequality constraint. The constraint of optimization problem is divided into 2 types. The first one is **Hard Constraint** in which all restrictions and requirements of the variables have to be satisfied. The second one is **Soft Constraint** which have some variable values that are penalized in the objective function if the constraints on these variables are violated.

The application of constrained optimization is really wide in the fields of finance and economics. In macroeconomics, constraint optimization can be usefully applied to determine the policies of tax or in microeconomics, this also can be used to minimize the cost by determinizing the input factors or maximize the output by considering involving factors.

2.5.1 General Form

The general form of a constrained minimization problem is as follows:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_i(x) = c_i,\ i = 1, \ldots, n. \\ & h_j(x) \geq d_j,\ j = 1, \ldots, m. \end{aligned}$$

Where:

$g_i(x) = c_i,\ i = 1, \ldots, n$ is called Equality Constraint.

$h_j(x) \geq d_j,\ j = 1, \ldots, m$ is called Inequality Constraint.

Both these constraints are required to be satisfied. Therefore, both are Hard Constraints.

2.5.2 Methods of Solution

Equality constraints

If all the constraints of constrained optimization problem are equility ones and not linear ones, one of the most effective methods to tackle this problelm is to use the method of Lagrange multipliers to convert a constrained optimization into an unconstrained optimization. After converting, the objective function of an unconstrained optimization problem will be that the original objective function pluses the product of Lagrange multiplier and the original constraints. If the constraints are all equality constraints and are all linear, a simpler way can be appied to solve. We can solve these constraints for some variables in term of other variables and then substitute the former into the objective function. Finally, we will get a much simpler unconstrained problem with a smaller number of variables.

Inequality constraints

If all the constraints of constrained optimization problem is inequility ones, Fritz John conditions and KarushKuhnTucker conditions can be used to convert the original problems into other forms which is simplier to tackle.

Linear programming

If the objective function and all the constraints of constrained optimization problem are linear. Then, this problem is called a linear programming problem. One of the most effective methods for solving a linear programming problem is simplex method which works by transformation of a linear programming into a standard form by normalizing restrictions and then finding the optimal solution by establishing and updating a simplex tableu.

Quadratic programming

If the objective function is quadratic and all the constraints of constrained optimization problem are linear. Then, this problem is called a quadratic programming problem. In the case that the objective function is convex, the ellipsoid method which minimize convex functions by iterations can be used to solve this problem in polynomial time. If the objective function is not convex, the problem is NP-hardness.

2.6 Short Sell

Suppose that Mr. A believes that the stock of FPT Company will decrease in the future. He calls his friend, Mr. B who currently have 100 shares of FPT, to borrow these 100 shares and A promises that he will return them back to B in the future. After that, A sell these 100 shares with the current price which is 45,000 VND, and A receives a cash inflow of 4500,000 VND. Three weeks laters, FPT's price experiences a decrease with the price is 40,000 VND. Now A buys back 100 shares of FPT with 4000,000 VND to give them to B. As a result, A's profit is 500,000 VND. This is short-selling process.

In finance, short selling is the practice that an investor sells stocks or other financial assets which is not currently owned by him/her, and after that he repurchase them. Investors having short selling behavior always expect that the price of the assets which they borrow will decline in the future. And short-selling allows them to gain the profit because the expense of repurchase is less than the inflow of cash by the initial sale.

In portfolio, investors also short some stocks in their portfolio so that they can obtain higher level of portfolio return. However, they are also exposed to higher level of risk. Because they can short stocks in their portfolios which is called unrestricted portfolios and the weight of assets can be negative and greater than 1.

Although short selling allows investors to benefit from a decrease in a stock's price, it can also results in some big problems. When the large number of investors decide to short a particular stock, which can have a significant influence on the share price of company. Many companies have experienced the declines in their share price due to short selling behaviour of short sellers. Therefore, prohibition on short selling behaviour has been implemented on some occasions.

Chapter 3

Models for Portfolio Optimization (PO)

In this chapter, we will mention and discuss about models using for Portfolio Optimization. Diversification is one of the fundamental principles of investment in finance field. It means that investors try to mix a wide variety of investment assets in their portfolios. The diversification of portfolio can help investors to minimize the risk as well as to maximize the return. Therefore, the portfolio optimization with diversification can be considered as a multi-objective optimization problem.

Nowadays, many methods can be used to tackle multi-objective optimization problems. One of the common methods for cope with a multi-objective optimization problem is to convert it into a single-objective optimization problem. These methods can be easily divided into 2 types: The first one is to choose the most important objective function as the main objective function while the rest of objective function is considered as constraints for the main objective function. Determining the most important objective function depends the decision of each investors. The second one is to construct an evaluation function by weighting the multiple objective functions. The evaluation function is considered as the only objective function for the problem.

3.1 Type 1: Markowitz Mean Variance models

The first method is Markowitz meanvariance model. This model is developed by Harry Markowitz in 1952. This model plays an important role in investment portfolio. Nowadays, it has become a foundation for constructing investment portfolios. This model assumes that among portfolios with the same return, the one with the least variance is the most efficient. "Efficient" means that for the given amount of return, the corresponding variance is minimized or for the given amount of risk, the corresponding return is maximized. In this model, the portfolio variance is considered as the objective function and the mean return is considered as a constraint.

The Markowitz Mean-Variance model for building a risky portfolio is described as:

$$\text{Minimize} \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij} \tag{3.1a}$$

$$\text{Subject to} \sum_{i=1}^{N} w_i r_i = R^*, \tag{3.1b}$$

$$\sum_{i=1}^{N} w_i = 1, \tag{3.1c}$$

$$0 \leq w_i \leq 1 \; i = 1, \ldots, N. \tag{3.1d}$$

Where:

N is the number of assets in investment portfolio.

σ_{ij} is is the covariance matrix between returns of assets i and j.

r_i is the mean return of assets i.

R^* is the desired mean return of the investment portfolio.

w_i is the weight of each asset in the investment portfolio.

The first row indicates that the objective function is the variance function, therefore, the main purpose of this model is to try to reduce the risk of the portfolio as much as possible. The second row means that the return of the portfolio has to be equal to the desired mean return. The third equation says that the sum of weights of all assets available in portfolio must be equal to 1. The final equation says that the weight of each asset in portfolio must be equal to or greater than 0 and less than or equal to 1. Therefore, it means that short-selling is prohibited.

3.2 Type 2: Single objective function model

The second method is to construct an evaluation function for portfolio optimization. It includes 2 models: Efficient Frontier and Sharpe Ratio models. Both are discussed as the following:

3.2.1 Efficient Frontier Model

In order to observe the each value of objective function for each different R value, a new risk aversion parameter $\lambda \in [0,\ 1]$ was invented. With the presence of a new parameter, the Markowitz Mean Variance model can be expressed as follows:

$$\text{Minimize } \lambda[\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij}] - (1-\lambda)[\sum_{i=1}^{N} w_i r_i] \tag{3.2a}$$

$$\text{Subject to} \sum_{i=1}^{N} w_i = 1, \tag{3.2b}$$

$$0 \le w_i \le 1 \; i = 1, \ldots, N. \tag{3.2c}$$

Where:

N is the number of assets in investmet portfolio.

σ_{ij} is is the covariance between returns of assets i and j.

r_i is the mean return of assets i.

w_i is the weight of each asset in the investment portfolio.

The first row indicates that the objective function is the trade-off between portfolio risk and return function. Therefore, the main purpose of this model is to try to reduce the trade-off between portfolio risk and return as much as possible. The second row says that the sum of weights of all assets available in portfolio must be equal to 1. The final row says that the weight of each asset in portfolio must be equal to or greater than 0 and less than or equal to 1 or it has the short-selling constraint.

With λ is 0, this model will become a maximization problem with the objective function which is the mean return of the portfolio regardless of the portfolio risk (variance). However, with λ is 1, this model will be a minimization problem with objective function which is the portfolio risk, regardless the mean return of the portfolio. Therefore, it is obvious that investors' sensitivity to the risk goes up as λ raises from 0 to 1, whereareas it goes down as λ falls to 0.

With the different values of λ, the objective function will have different values including the values of mean return and variance. Finding the combinations between mean return and variance of portfolio by varying parameter λ, we can graph a continuous curve which is called an Efficient Frontier Curve according to the Markowitz Theory [8].

Because the efficient frontier is curve giving the optimal trade-offs between mean return and risk of portfolio, each point on this curve represents an optimal portfolio. Therefore, this states that the portfolio optimization problem is naturally a multi-objective optimization problem. In order to solve a problems with mutiple objective function, a new risk aversion parameter λ is added to covert it into a single-objective function problem.

3.2.2 Sharpe Ratio model

Besides the mean variance efficient frontier, we can maximize the Sharpe Ratio as an alternative way to build an optimal portfolio. This model was developed by Sharpe in 1966 [9], it is also called reward-to-variability ratio. This model was built based on Markowitz's Mean Variance paradigm. The Sharpe Ratio is simple and it is a measure for risk-adjusted return often used to assess the portfolio performance. It is described as the following equation:

$$\text{Maximize}\frac{\sum_{i=1}^{N} w_i r_i - R_f}{\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij}}} \tag{3.3a}$$

$$\text{Subject to}\sum_{i=1}^{N} w_i = 1, \tag{3.3b}$$

Where:

N is the number of assets in investmet portfolio.

σ_{ij} is is the covariance between returns of assets i and j.

r_i is the mean return of assets i.

w_i is the weight of each asset in the investment portfolio.

R_f is risk free rate (i.e. the interest rate on a three-month U.S. Treasury bill).

The first row indicates that the objective function is the reward-to-variability ratio function or Sharpe Ratio function. The main purpose of this model is to try to increase this ratio as much as possible. The second row says that the weights of all assets available in portfolio must sum up to 1.

By selecing the weights w_i of the portfolio, we can maximize the Sharpe Ratio in effect balancing the trade-off between maximizing the return and minimizing the risk at the same time. In this thesis, Sharpe Ratio is applied in PSO to obtain the optimal portfolio.

Chapter 4

Particle Swarm Optimization Algorithm

4.1 Introduction

PSO algorithm, which was first introduced by James Kennedy and Russell Eberhart in 1995[10], is a population-based stochastic optimization technique. The basic idea of this algorithm is initially inspired by simulation of the social behavior, as a representation of the move of organisms in a swarm such as a school of fish, colony of ant and flock of bird. It is based on the natural process of communication as a group. Even though each individual in a swarm does not have wisdom, by interacting or sharing information each other in the search space, they can easily solve highly complicated problems as a group.

PSO may sound really complex, but it is actually a very simple algorithm. PSO could be understood more easily by considering an example of the process of finding food of a bird flocking. The real search space for food is three-dimensional space. At the beginning, the whole flock flies in random directions in three-dimensional space to seek food. However, after a while, some individuals in the flock start to find out the locations containing the food. Depending on the amount of food which is sought, individuals finding out food will send signals to other individuals which are searching in the vicinity and this signal will spread throughout the swarm. Based on the received information, each individual will adjust its direction and velocity to fly toward the location where the food is the most available. Such a transmission mechanism helps the flock of bird to find the location containing the largest amount of food in the search space.

Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) are the most successful techniques of Swarm Intelligence. Swarm Intelligence is the discipline which is based on natural or artificial systems containing many individuals that coordinate mutually according to the collective behavior of decentralized, self-organized system. Fish schooling, bird flocking and bacterial growth are outstanding illustrations for the natural systems of Swarm Intelligence. In PSO, each individual flies through the multidimensional space and it has to depend on experience of itself and of all members in swarm to adjust direction in every iteration to move toward the best location of the entire swarm. Therefore, the PSO algorithm is a part of Swarm Intelligence.

Although PSO has many the same similarities with evolutionary computation techniques such as Genetic Algorithms, which find the optimal solution by updating generation with the initial population of possibly random solutions. PSO does not has bio-inspired operators such as mutation, crossover and selection. Instead, PSO Algorithm focuses on finding the optimal solution by letting possible solutions follow the current optimums of each particle and the optimum of the entire swarm. The greater detailed information of PSO algorithm will be mentioned in the next section.

Nowadays, PSO is becoming a popular method to solve complex tasks such as non-linear and non-convex optimization problems because of simplicity, the easy implement and the fast convergence to the optimal value. As compared with GA and other optimization methods, PSO is faster, more efficient and powerful, and there are few parameters to adjust. Recently, PSO has been successfully applied to many fields such as mathematics, finance and construction and so on.

4.2 Algorithm

In the algorithm of PSO, it starts with the initialization of a population of possibly random solutions. These solutions are positions in the search space. Each solution is considered as a bird, called a particle. Each particle has the following characteristics:

- Each particle has a fitness value which can be calculated by using objective function, it also knows its current position and velocity.
- Each particle has ability to record its individual best performance.
- Each particle knows the best performance of its swarm.

In the search space, every particle will modify their positions over each iteration by adjusting their velocities in order to move according to the personal optimal particles and the global optimal particle. Velocities are adjusted by considering their personal best positions and the best position of the entire group obtained so far during the search process. The new positions which are generated by combining the current positions and the adjusted velocities are the next destinations for particles to reach. Therefore, when the processes of adjusting velocities and updating postions are repeated, the particles have a tendency to move to the better and better positions over the search space. If the number of repetitions (iterations) is large enough, all particles will converge to the optimal position and a satisfactory solution is discovered.

The initial population of N particles in D-dimensional space is denoted as X=$[\vec{X}_1, \vec{X}_2, ..., \vec{X}_N]^T$, which T is transpose operator. Each particle $\vec{X}_i$(i= $1, 2, 3...N$) is expressed as $\vec{X}_i = [X_{i,1}, X_{i,2}, ..., X_{i,D}]$. Besides, the initial velocity of the initial population is denoted as V=$[\vec{V}_1, \vec{V}_2, ..., \vec{V}_N]^T$. Therefore, each particles $\vec{X}_i = [X_{i,1}, X_{i,2}, ..., X_{i,D}]$ will have the velocity which is expressed as $\vec{V}_i = [V_{i,1}, V_{i,2}, ..., V_{i,D}]$ with i from 1 to N and j from 1 to D.

Each particle will use the following information to update its new positions:

- The current position $\vec{X}(t)$,
- The current velocity $\vec{V}(t)$,
- The difference between the personal best position ($\vec{PB}$) and the current position ($\vec{PB}$ - $\vec{X}(t)$),

- The difference between the best position of the entire swarm ($\vec{GB}$) and the current position ($\vec{GB}$ - $\vec{X}(t)$)

Therefore, the velocity change is defined according to the following formulation:

$$\vec{V_{i,j}}(t+1)=w\times\vec{V_{i,j}}(t) + c_1\times r_1\times[P\vec{B}_{i,j}(t) - \vec{X_{i,j}}(\text{t})] + c_2\times r_2\times[G\vec{B}_j(t) - \vec{X_{i,j}}(t)]$$

where:

index j is the number of dimension of particle i

t is the iteration sequence of the particle i.

c_1 and c_2 are positive constant parameters which are called acceleration coefficients. Their functions are to control the maximum step size.

r_1 and r_2 are random numbers in the range of [0,1].

w is a constant number which is called inertia weight.

$\vec{V_{i,j}}(t+1)$ is the velocity of particle i with the jth dimension at iteration t+1.

$\vec{V_{i,j}}(t)$ is the velocity of particle i with the jth dimension at iteration t.

$\vec{X_{i,j}}(t)$ is the current position of particle i with the jth dimension at iteration t.

$P\vec{B}_{i,j}(t)$ is the personal best position with the jth dimension of the particle.

$G\vec{B}_j$ is the best position with the jth dimension of the entire swarm which was obtained so far .

Finally, the new position of particle i, $\vec{X_i}(t+1)$ is calculated as the following fomulation:

$$\vec{X_{i,j}}(t+1) = \vec{X_{i,j}}(t) + \vec{V_{i,j}}(t+1)$$

where:

$\vec{X_{i,j}}(t)$ is the current position of particle i with the jth dimension at iteration t.

$\vec{V_{i,j}}(t+1)$ is the velocity of particle i with the jth dimension at iteration t+1.

$\vec{X_{i,j}}(t+1)$ is the position of particle i with the jth dimension at iteration t+1.

Figure 4.1 shows the search mechanism of PSO in 2-dimensional search space.

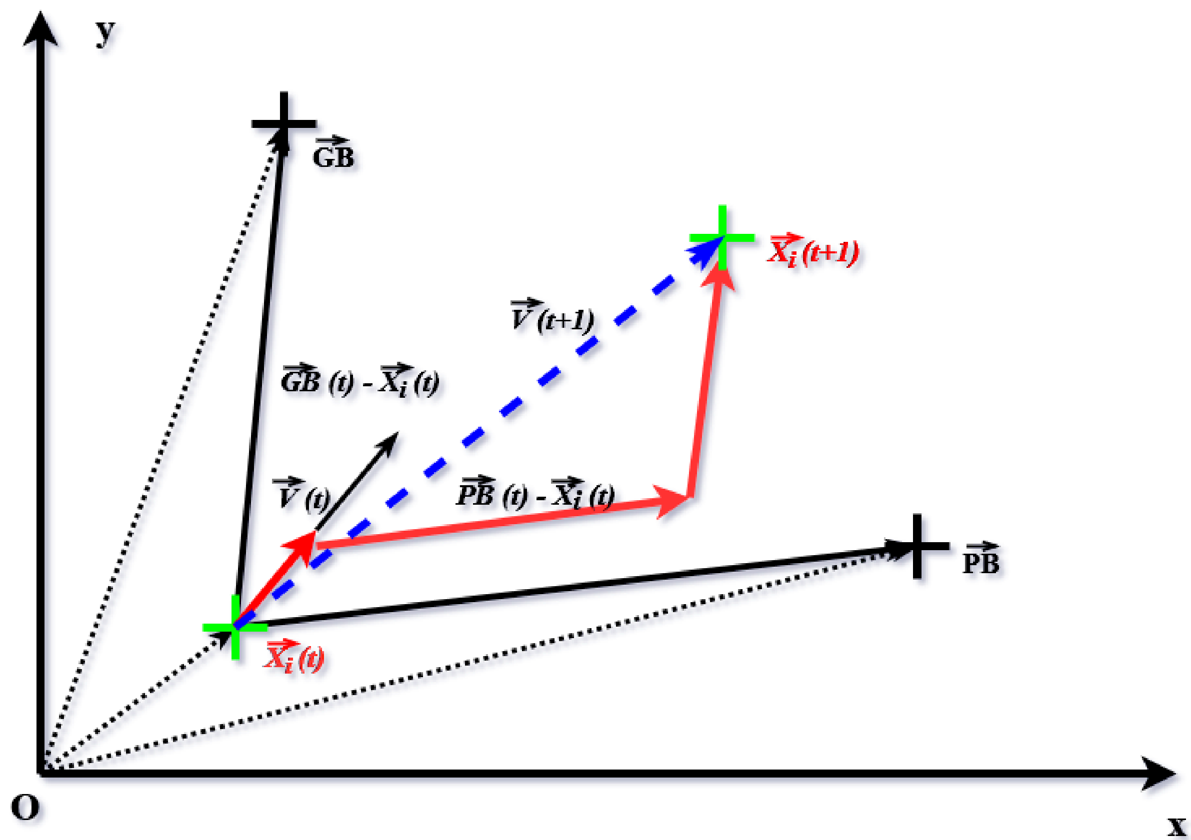


Figure 4.1: Search mechanism of PSO in 2-dimensional search space

In other words, the steps of PSO method can be given in greater detail as follows:

Note that: this is steps of PSO for minimization problems.

1. Set parameter c_1, c_2 and w of PSO and set $Maxite$(the maximum number of iterations).

2. Initialize population of particles with random positions X and velocities V (iteration = 0).

3. Set iteration count $t = 1$ (iteration = 1).

4. Calculate the fitness value of particles $F_i(t) = f(\vec{X}_i(t))$, $\forall i$ and find the index of the best particle g .

5. Choose $P\vec{B}_i(t) = \vec{X}_i(t)$, $\forall i$ and $G\vec{B}(t) = \vec{X}_g(t)$.

6. Update velocity and position of particles
   $\vec{V}_{i,j}(t+1) = w \times \vec{V}_{i,j}(t) + c_1 \times rand() \times (P\vec{B}_{i,j}(t)$ - $\vec{X}_{i,j}(t)) + c_2 \times rand() \times (G\vec{B}_j(t)$ - $\vec{X}_{i,j}(t))$, $\forall i$ and $\forall j$
   $\vec{X}_{i,j}(t+1) = \vec{X}_{i,j}(k) + \vec{V}_{i,j}(k+1)$, $\forall i$ and $\forall j$.

7. Re-calculate the fitness value $F_i(t+1) = f(\vec{X}_i(t+1)$, $\forall i$ and find the index of the best particle $g1$.

8. Update Pbest
   If $F_i(t+1) < F_i(t)$ then $P\vec{B}_i(t+1) = \vec{X}_i(t+1)$ else $P\vec{B}_i(t+1) = P\vec{B}_i(t)$, $\forall i$

9. Update Gbest
   If $F_{g1}(t+1) < F_g(t)$ then $G\vec{B}(t+1) = PB_{g1}^{k+1}$ and set $g = g1$ else $G\vec{B}(t+1) = G\vec{B}(t)$

10. If $t < Maxite$ then $t = t + 1$ and go to step 6 else go to step 11

11. Print optimum solution as $G\vec{B}(t)$

A detailed flowchart of PSO considering the above steps is shown in **Figure 4.2**.

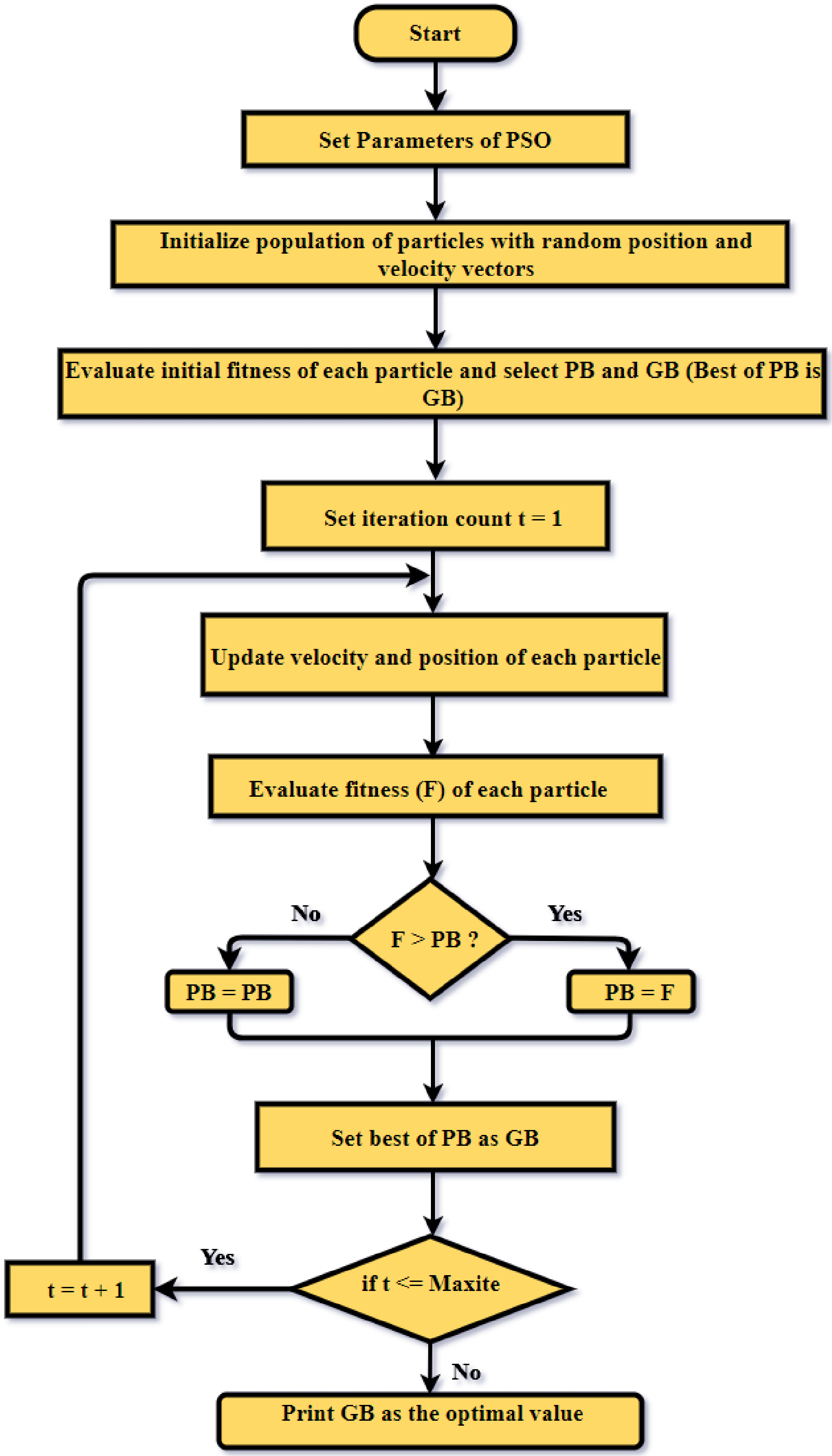


Figure 4.2: Flowchart of PSO method

4.3 Parameter Selection

As we can see the PSO method requires a few of input parameters such as stopping criteria, acceleration coefficients and inertia weight. The choice of these parameters influences significantly the performance of PSO. Therefore, these parameters play a really important role in this algorithm. In this section, different strategies for setting every input parameter of PSO method are explained.

4.3.1 Population Size

The population size which is the number of particles directly influences the performance of PSO Algorithm. Too few particles for initial population will result in the fact that the algorithm will get stuck in local optima. On the other hand, too many particles for initial population will take this algorithm the large amount of time to get the optimal solution.

There is no existing rule in literature for the population size selection. But normally, the population size is directly proportional to the dimension of search space of problem. In other words, the population size should be large when the dimension of search space is high and the population size should be small when the dimension of search space is low. Most papers about PSO method often choose 100 particles for population size [11].

4.3.2 Stopping criteria

It is the number of iterations or the number of times of modifying position for a particle to reach the optimal area. Selecting stopping criteria depends on the dimension of problem and decision of an investor.

There is also no existing rule in literature for selection of stopping criteria. However, too few iterations will not be enough for a particle to reach the promising area while too many iterations also will waste time and expenses. Normally, the number of iterations should be between 500 and 10000 [11].

4.3.3 Acceleration coefficients

Acceleration coefficients which is the weighting of stochastic acceleration terms directly affect the performance of PSO. These coefficients are the important factors for pulling particles towards the personal best position (PB) and the global best position (GB). If these coefficients are too low, theo particles will move too slowly and this will require considerably computational efforts, in many cases, particles cannot converge to the optimal position during search process. On the other hand, if these coefficients are too big, particles will suddenly move in a high level and easily converge to the false optimal location.

Moreover, the value of these coefficient is really important and has significant influences on the behaviour of this algorithm. If the cognitive acceleration coefficient (c_1) increases, it will result in the increase in the attraction of particles towards PB and the decrease in that towards GB. Similarly, if the social acceleration coefficient (c_2) increases, the decrease in the attraction of particles towards PB and the increase in that towards GB will occur.

Initial PSO studies chosen $c_1 = c_2 = 2.0$. Even though good results have been gained, a problem emerged was that that velocities quickly exploded to big numbers. According to the literature [12], acceleration coefficients should be 1.149618, or $c_1 = c_2 = 1.49618$.

4.3.4 Inertia weight

Shi and Eberhart invented inertia weight in order to control velocity, but still to avoid divergence. Its main function is to control the momentum of the particle by including the contribution rate of the previous velocity. It determines how much the new velocity will be influenced by the previous moving direction.

Initial empirical studies of PSO have indicated the value of inertia weight to make ensure convergence [12, 13]. If $w > 1$, divergence behavior will happen when velocities go up, and therefore, the particles cannot move towards the optimal area. If $w < 1$, particles will reduce its speed until they no long move, as long as $2w > (c_1 + c_2) - 2$, as shown in [14, 15]. Also, according to the literature [12], inertia weight should be 0.7298, or $w =$

0.7298.

Empirical results have indicated that a constant value of inertia is 0.7298, or $w = 0.7298$ and the value of acceleration coefficients is 1.49618, or $c_1 = c_2 = 1.49618$ will provide a good convergent behaviour [12].

4.4 Example of PSO method

In this section, we will look at a simple example for understanding more deeply how the PSO work in order to get the optimal value.

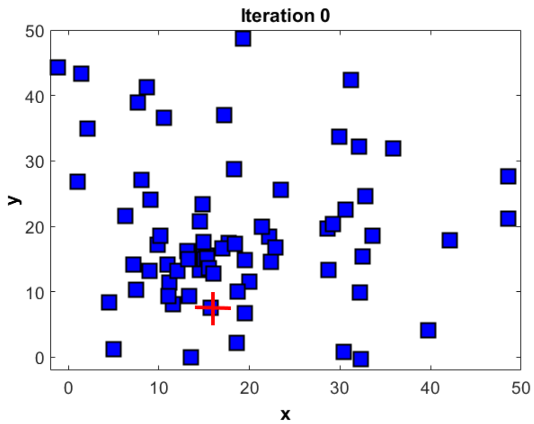
To see how the mechanism of PSO method work when this method is applied to solve an optimization problem, we consider the following basic example:

$$\text{minimize } (x-20)^2 + (y-10)^2$$

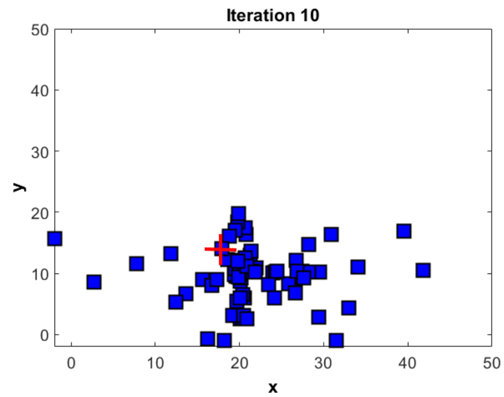
For this particular example, we can easily obtain the optimal solution which is 0, for $x = 20$ and $y = 10$.

Pretending that this is a difficult optimization problem, we cannot obtain the optimal solution of this example by arithmetical calculations performed in the mind. Therefore, we must use the PSO to find the optimal solution.

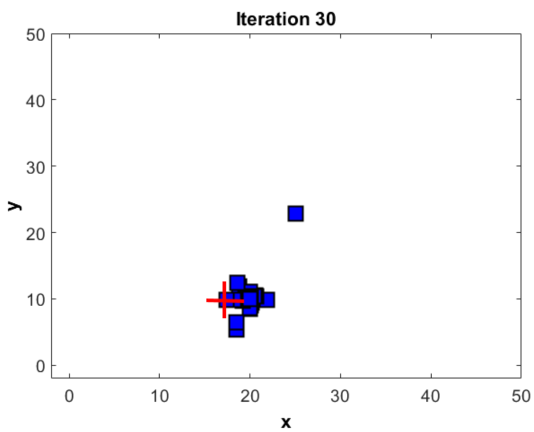
In order to give a clear illustration for this algorithm, we used Matlab as a tool for calculating and plotting the behaviour of a swarm of 100 particles in **Fingure 4.3**



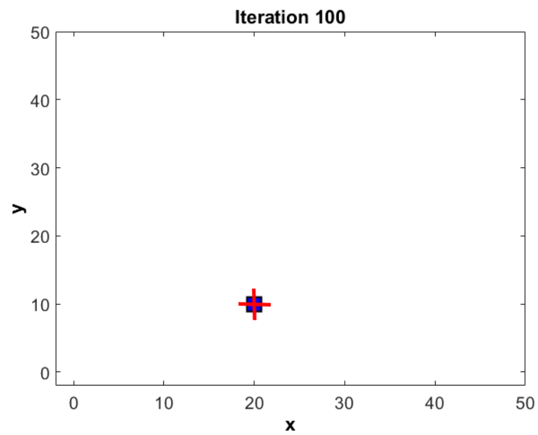
(a) Iteration 0



(b) Iteration 10



(c) Iteration 30



(d) Iteration 100

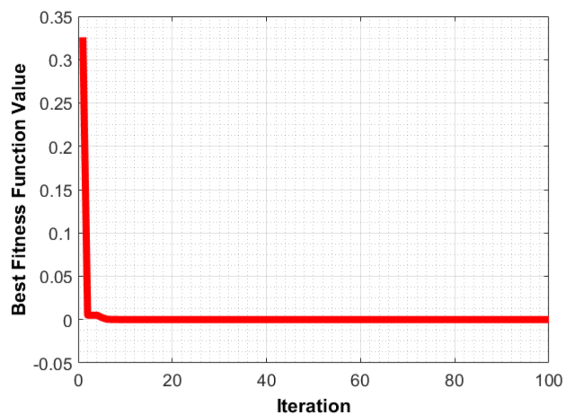
Figure 4.3: Particle Swarm Optimization for a minimization problem (2-dimension)

Fingure 4.3 (a) shows the initial positions of all particles in the population. Each particle is a blue square which is randomly scattered in the 2-dimentional search space.

The red plus sign (+) in **Fingure 4.3 (a)** to **Fingure 4.3 (d)** represents the final optimal weights of x and y which are obtained by PSO Algorithm.

After 10 interations, all particles present a certain pattern of movement, as given in **Fingure 4.3 (b)**. We can see that the behavior of particles in **Fingure 4.3 (b)** is obviously similar to the bird flocking. **Fingure 4.3 (c)** represents the status of all particles after 30 times of changing positions. We can easily see that many particles have moved close to the optimal point. This give us an better illustration about the result of convergence to the place which contains the most available food of bird flocking when each particle tries to follow one closest to the place having the largest amount of food after 30 times of changing positions.

At iteration 100, all the particles landed on the same point which is the optimal solution as shown in **Fingure 4.3 (d)**. We can find that the otimal point is 0 for $x = 20$ and y=10 by using PSO method. This is the same as the real solution of this example.



(a) Updating the best fitness value

Figure 4.4: PSO for a minimization problem (2-dimension)

By looking at from **Fingure 4.3 (a)** to **Fingure 4.3 (d)**, we realize that by adjusting velocity which is based on the personal best position (PB) and the global best position (GB) to modify the position, all particles have a tendency to move onto the optimal point after the specific number of iterations.

In **Fingure 4.4** represents the the process of convergence of the optimal value by using PSO. Because this is a simple problem with 2-dimension, particles of PSO just need 4 or 5 iterations to find the optimal value. Through this example, we can see that PSO probably a good method to tackle optimization problem.

4.5 Optimization test function

In order to prove that the algorithm of PSO works well and exactly, we will use PSO method to find the optimal solutions of test functions of optimization by using Matlab. Test functions for optimization are designed to evaluate characteristics of optimization algorithms, including : convergence rate, precision, robustness and general performance. The evaluation by using test function is one of the most popular ways to check the performance of a new optimization algorithm.

In this section, we will focus on finding the optimal solutions of test functions by using PSO method, then we compare the results of PSO with the actual optimal solutions of these function. Because each test function represents the different situations which optimization algorithm will deal with. Therefore, this way is an effective way to evaluate the precision, covergence and perfomance of PSO algorithm.

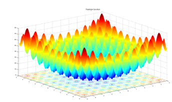
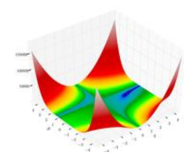
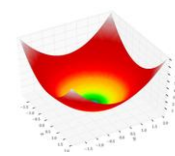
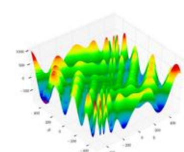
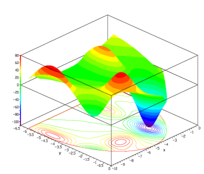
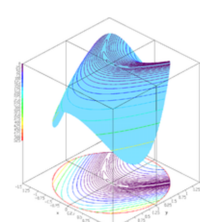
We can see that **Table 4.1** contains six famous test functions of optimization including: 4 single-objective functions and 2 contrained functions. This table gives us information about Name and Formular, Domain of variables and Plot in 3D.

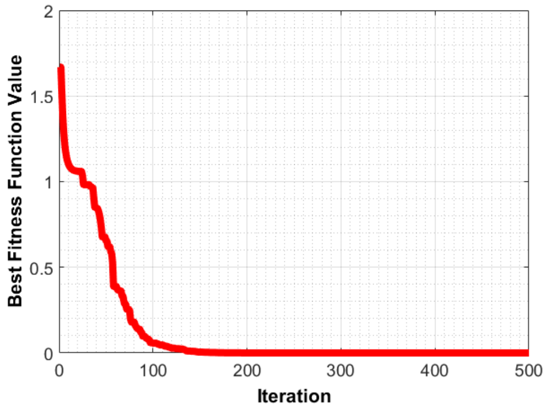
Table 4.2 shows the perfomance of Particle Swarm Optimization on these 6 functions. In other words, this table represents the optimal solutions of test functions obtained by using PSO method and this table also shows the the real optimal solutions of these functions.

After comparing the global minimums obtained by PSO with the actual global minimums, we can realize that PSO algorithm always provides the correct solution. Therefore, PSO is a good method for solving optimization problems with low or high demension.

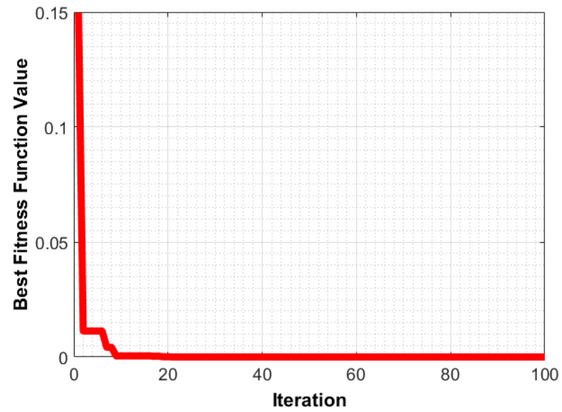
We can see how PSO algorithm update the global minimum value until it reaches the final values as in **Table 4.2** by looking at **Figure 4.5**. The number of iterations depends on the complexity of problems and that is why these numbers are different in each test function.

Table 4.1: Test functions for single-objective and constrained optimization

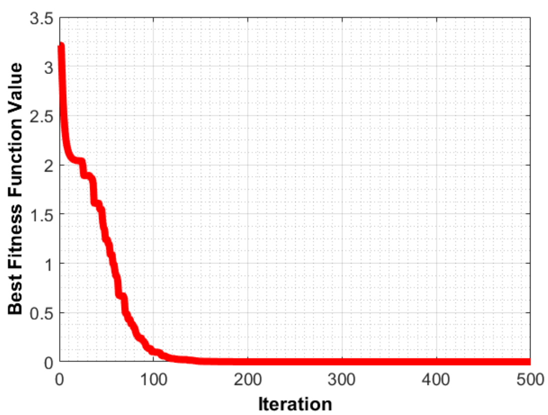
Name	Formular	Domain	Plot
Rastrigin	f=10n+$\sum_{i=1}^{n}[x_i^2$-10cos$(2\pi x_i)]$	-5.12≤x_i≤5.12	
Beale	f=$(1.5-x+xy)^2+(2.25-x+xy^2)^2$ $+(2.625-x+xy^3)^2$	-4.50≤x, y≤4.50	
Sphere	f=$\sum_{i=1}^{n} x_i^2$	-100≤x_i≤100	
Eggholder	f=$-(y+47)sin(\sqrt{\lvert x/2+(y+47)\rvert}$ - $xsin(\sqrt{\lvert x-(y+47)\rvert}$	-512≤x, y≤512	
Mishra's Bird	f=$sin(y)e^{\lvert(1-cosx)^2\rvert}+cos(x)e^{\lvert(1-siny)^2\rvert}$ $+(x-y)^2$ subject to: $(x+5)^2+(y+5)^2<25$	-10≤x≤0, -6.5≤y≤0	
Rosenbrock	f=$(1-x)^2+100(y-x^2)^2$ subject to: $x^2+y^2<2$	-1.5≤x, y≤1.5	



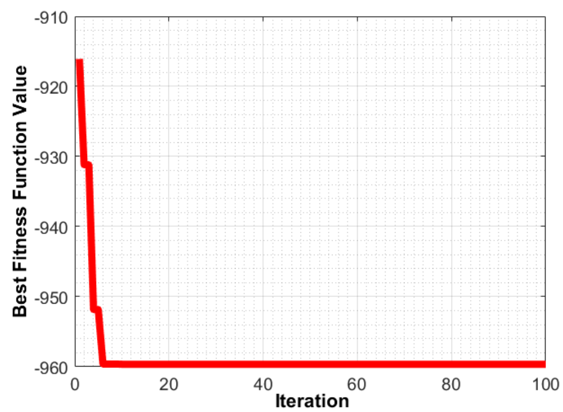
(a) Rastrigin function



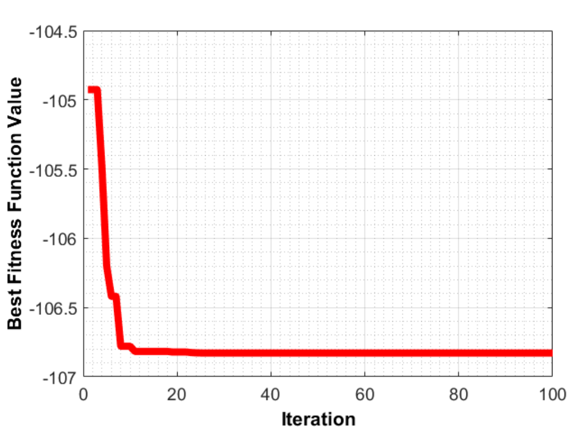
(b) Beale's function



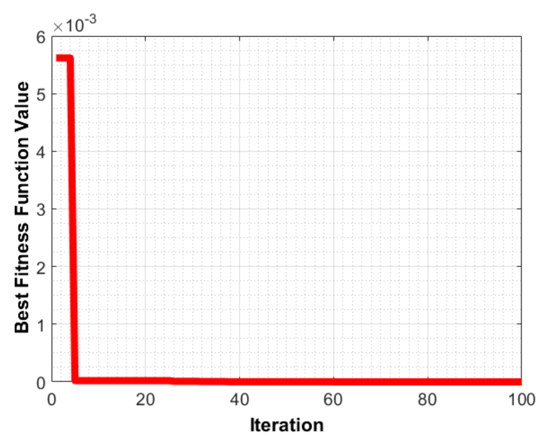
(c) Sphere function



(d) Eggholder function



(e) Mishra's Bird function-constrained



(f) Rosenbrock function constrained to a disk

Figure 4.5: PSO: Updating the global minimum value of test functions

Table 4.2: PSO Performance

Name	Dim	The Actual Global Minimium	The Global Minimum of PSO
Rastrigin function	30	$f(0) = 0$	$f = 0$, $\forall x = 0$
Beale's function	2	$f(3, 0.5) = 0$	$f = 0$ with $x = 3$ and $y = 0.5$
Sphere function	30	$f(0_1, .., 0_{30}) = 0$	$f = 0$, $\forall x = 0$
Eggholder function	2	f(512,404.23) = -959.64	f = -959.64, for $x = 512$ and $y = 404.23$
Mishra's Bird function	2	f(-3.13,-1.5821) = -106.76	f = -106.76, for x = -3.1302 and y = -1.5821
Rosenbrock function constrained to a disk	2	f(1,1) = 0	$f = 0$, for $x = 1$ and $y = 1$

4.6 Application

The PSO algorithm is simple in concept, easy to implement, few input parameters and computational efficient. Therefore, Particle Swarm Optimization are applied into many fields since the first practical application of PSO was in the field of neutral network trainning [16].

Nowadays, PSO algorithms have been developed and improved to solve:

- Fuzzy and neuro-fuzzy systems and control [17, 18].
- robotic [19, 20].

- predictation and forcasting [21, 22, 23].
- design and optimization of communication network [24, 25].
- clustering, classification and data mining [26, 27].
- neutral network [16, 28].
- signal processing [29].
- medical, biological and pharmaceutical [30].

Chapter 5

Particle Swarm Optimization Method for Portfolio Optimization

In this chapter, we will focus on applying PSO method into Portfolio Optimization Problem. As noted in **Chapter 2** that nowadays, the portfolios of investors, investment companies and financial institutions have the large number of assets instead of the limited number of ones in the past. Moreover, there are some drawbacks in meta-heuristic algorithms such as a lack learning ability (Fuzzy), over-fitting problems (ANN). With these special situations, Quadratic Programming, Fuzzy, Artificial Neural Network and Genetic Algorithm methods cannot be suitable to tackle the portfolio optimization problems efficiently. As a result, we should apply other methods to handle this problem efficiently. Particle Swarm Optimization is one of these methods.

In order to apply PSO method in Portfolio Optimization, we need to do the following steps:

1. Set up the fitness function for PSO. The objective functions of portfolio optimization problems are always considered as the finess functions for PSO.

2. Set up the particle movement for PSO.

3. Set up parameters for PSO.

4. Set up constraint satisfaction for PSO

5.1 Fitness function

Fitness function plays a critical role in the PSO method. Each particle in the population of PSO has a fitness value, and it moves in the search space with respect to its previous position where it has met the best fitness value PB and the previous position of the entire swarm where swarm has met the best fitness value GB. In this thesis, the Sharpe Ratio (according equation **3.3a**) and the Efficient Frontier (according equation **3.2a**) will be used as objective functions for the problem regardless of restricted or unrestricted portfolios. They are defined as follows:

$$\text{Minimize} \quad f_P = \lambda[\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij}] - (1-\lambda)[\sum_{i=1}^{N} w_i r_i]$$

And,

$$\text{Maximize} \quad f_P = \frac{\sum_{i=1}^{N} w_i r_i - R_f}{\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij}}}$$

Where:

where f_P is the fitness value of particle p.

At every iteration, the personal best position of each particle and the best position of the entire swarm are updated if there is an improvement in the values of the best fitness.

5.2 Particles movement

As indicated in **Chapter 4**, at each iteration, each particle will adjust its velocity to change the current position towards the optimal value. The velocity is adjusted by considering its personal best position and the best particle of the swarm obtained so far. The velocity change is defined as:

$$\vec{V_{i,j}}(t+1) = w \times \vec{V_{i,j}}(t) + c_1 \times r_1 \times [\vec{PB}_{i,j}(t) - \vec{X_{i,j}}(\mathrm{t})] + c_2 \times r_2 \times [\vec{GB_j}(t) - \vec{X_{i,j}}(t)]$$

And the new position of each particle is calculated by:

$$\vec{X_{i,j}}(t{+}1) = \vec{X_{i,j}}(t) + \vec{V_{i,j}}(t{+}1)$$

5.3 Parameter Selection

Parameter Selection influences directly the entire performance of PSO. In order to make sure convergence, precision as well as other important factors, different strategies for setting each PSO parameter are considered as follows:

- Population Size: 100.
- The number of interations: 500 to 10000.
- Acceleration Coefficients: $c_1 = c_2 = 1.49618$.
- Inertia weight: 0.7298.
- Initial velocity: 10 percent of the initial position.

5.4 Constraint satisfaction

There are 2 kinds of risky portfolios in the financial market [30]: unrestricted and restricted portfolios. Unrestricted risky portfolio means that short-selling is allowed. Investors constructing this type of portfolio will choose to sell a security that he/she does not own by borrowing based on the belief that he/she can buy it back after a period of time at lower price to make a profit. In other words, the weight of each asset in the unrestricted risky portfolios can be negative numbers or greater than 1. For restricted risky portfolios, the short-selling is prohibited. The investors have no right to borrow a security from others for sale. The weight of each assets in the restricted risky portfolios must be a positive number and not be greater than 1.

In reality, besides the contraints from short-selling, both unrestricted and restricted optimal risky portfolios are often under many other constraints or requirements such as transportation costs, tax expenses and many other personal requirements of owners. However, in this paper, we will ignore all above constraints and only focus on the short sale constraints and the requirement that the sum of weights of all assets in portfolio must be 1.

5.4.1 Sharpe Ratio Model.

The portfolio optimization problem (including fitness function and constraints) for a restricted risky portfolio is expressed as:

$$\text{Maximize} \frac{\sum_{i=1}^{N} w_i r_i - R_f}{\sqrt{\sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \sigma_{ij}}} \tag{5.1a}$$

$$\text{Subject to} \sum_{i=1}^{N} w_i = 1, \tag{5.1b}$$

$$0 \leq w_i \leq 1 \; i = 1, \dots, N. \tag{5.1c}$$

The portfolio optimization problem (including fitness function and constraints) for a unrestricted risky portfolio is defined as:

$$\text{Maximize} \frac{\sum_{i=1}^{N} w_i r_i - R_f}{\sqrt{\sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \sigma_{ij}}} \tag{5.2a}$$

$$\text{Subject to} \sum_{i=1}^{N} w_i = 1, \tag{5.2b}$$

$$i = 1, \dots, N. \tag{5.2c}$$

When the number of investment assets in a risky portfolio goes up, finding an optimal risky portfolio will become an increasingly highdimensional optimization problem with bigger constrains. Therefore, this problem will be more difficult to solve.

In the case of restricted portfolio, the personal best position (PB) and the global best postion (GB) are evaluated by Equations. **5.1 (a)** – **5.1 (b)**. On the other hand, the

PB and GB are evaluated using Equations **5.2 (a)** – **5.2 (b)** if this is an unrestricted portfolio. In order to make sure a valid movement in the search space, whenever a particle modifies its velocity to move to a new position in the search space, all the constraints as well as requirements of the portfolio must be met. When the terminating condition has been fulfilled, particles will totally reach the global optimum or nearly reach that optimum.

5.4.2 Efficient Frontier Model.

The portfolio optimization problem (including fitness function and constraints) for a restricted risky portfolio is expressed as:

$$\text{Minimize } \lambda[\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij}] - (1-\lambda)[\sum_{i=1}^{N} w_i r_i] \tag{5.3a}$$

$$\text{Subject to} \sum_{i=1}^{N} w_i = 1, \tag{5.3b}$$

$$0 \leq w_i \leq 1 \; i = 1, \dots, N. \tag{5.3c}$$

The portfolio optimization problem (including fitness function and constraints) for a unrestricted risky portfolio is defined as:

$$\text{Minimize } \lambda[\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij}] - (1-\lambda)[\sum_{i=1}^{N} w_i r_i] \tag{5.4a}$$

$$\text{Subject to} \sum_{i=1}^{N} w_i = 1, \tag{5.4b}$$

$$i = 1, \dots, N. \tag{5.4c}$$

The personal best position (PB) and the global best postion (GB) are evaluated by Equations **5.3 (a)** – **5.3 (b)** for the restricted portfolios. On the other hand, the PB and GB are evaluated using Equations. **5.4 (a)** – **5.4 (b)** in the case of the unrestricted portfolio. Similarly, particles have to make sure that all constraints of the problem must be satisfied during the search process. In efficient frontier model, this problem is not fixed because it depends on risk aversion of investors.

Chapter 6

Experiments and Discussion

6.1 Data

The experiments of PSO for the portfolio optimization by using two models: Sharpe Ratio Model and Efficient Frontier Model have been performed on six risky portfolios, including three restricted risky portfolios with 8 stocks, 15 stocks and 30 stock, and three unrestricted risky portfolios with the same numbers of stocks. A restricted portfolio means that it does not allow short-selling. In other words, the weight of each asset in a restricted portfolio must be positive, less than or equal to 1. On the other hand, an unrestricted portfolio does not have the short-selling constraint. It means that the weight can be positive, negative and greater than 1.

All securities from risky portfolios with 8 stocks and 15 stocks are chosen from VN-30 INDEX. Securities from risky portfolio with 30 stocks are selected from VN-30 INDEX and VN-INDEX due to the unavailability of historical data of some stocks in VN-30 INDEX. The names of stocks in each porfolio are listed in Appendix. The reason for choosing this dataset is that we want to test PSO method with real historical data instead of artifical data. The simulation by using the artifical data does not provide good illustrations for the real financial market. Such stock data is openly available at online sources such as vietstock.vn, www.vndirect.com.vn and www.cophieu68.vn. In this paper, we use daily price information from vietstock.vn and the time period is from 01/03/2012 to 16/06/2017. We downloaded the historical prices over that time period and use them for calculations of the daily mean return and daily standard deviation by using the daily closing prices. After that, we calculate the yearly mean return and yearly standard deviation by using daily ones. We can easily get the covariance matrix of stocks by using the daily returns.

6.2 Lagrange Method

6.2.1 Introduction

Joseph-Louis Lagrange introduced Lagrange multiplier method. In mathematical optimization, this method is a way to obtain the local maxima and minima of a given function which is under some equality constraints.

Suppose we have a optimization problem as following:

$$\text{Maximize } f(x) \tag{6.1}$$

$$\text{Subject to } g_i(x) = 0; i = 1, \ldots, N \tag{6.2}$$

Suppose that f and g have continuous first partial derivatives. In order to find the optimal solution of f, Lagrange multiplier method works as follows "Put the objective function and the constraints into a single minimization problem (called Lagrangian expression), but multiply each constraint by a new factor (λ_i) which is called Lagrange Multipliers" and the Lagrange expression is given by:

$$\mathcal{L}(x, \lambda) = f(x) + \textstyle\sum_{i=1}^{N} \lambda g(x)$$

where:
$\mathcal{L}(x, \lambda)$ is the Lagrangian and it depends also on λ, which is a vector of multipliers.

We will find the roots of the gradient of the objective function with respect to x to find the optimal solution for minimization and maximization problems (in case of convex or concave function). In other words, set the gradient of $\mathcal{L}$ equal to the zero vector.

$$\bigtriangledown_x \mathcal{L}(x, \lambda) = \bigtriangledown_x f(x) + \textstyle\sum_{i=1}^{N} \bigtriangledown_x \lambda_i g_i(x) = 0$$

and find the critical points of $\mathcal{L}$. After that we will consider each solution by plugging each one into f. Which one gives the largest (or smallest) value is the maximum (or minimum) point you are finding.

6.2.2 Lagrange Method for Markovitz Model

As discussed in Chapter 3, Markovitz Model can be expressed as:

$$\begin{aligned} &\text{Minimize} \quad \lambda[\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij}] - (1-\lambda)[\sum_{i=1}^{N} w_i r_i] \\ &\text{Subject to} \quad \sum_{i=1}^{N} w_i = 1 \\ &\qquad\qquad\quad 0 \le w_i \le 1\, i = 1, \ldots, N. \end{aligned}$$

This optimization problem can be solved using the Lagrange method. Finding the optimal solution of the above problem is equivalent to finding the solution of the folllowing function:

$$\mathcal{L} = \tfrac{1}{2}\lambda[\textstyle\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij}] - (1-\lambda)[\sum_{i=1}^{N} w_i r_i] + a_1(\sum_{i=1}^{N} w_i - 1)$$

Where a_1 is the constant Lagrange multiplier coefficient. Set the gradient of $\mathcal{L}$ equal to the zero vector and find the critical points. These points have to satisfy the following system:

$$\begin{cases} \mathcal{L}'_{w1} = 0 \\ \mathcal{L}'_{w2} = 0 \\ \vdots \\ \mathcal{L}'_{wi} = 0 \\ \sum_{i=1}^{N} w_i - 1 = 0 \end{cases} \Leftrightarrow \begin{cases} \lambda.w_1\sigma_1^2 + \lambda.w_2\sigma_{12} + ... + \lambda.w_i\sigma_{1n} - \frac{1}{2}(1-\lambda).r_1 + a_1 = 0 \\ \lambda.w_1\sigma_{12} + \lambda.w_2\sigma_2^2 + ... + \lambda.w_i\sigma_{2n} - \frac{1}{2}(1-\lambda).r_2 + a_1 = 0 \\ \vdots \qquad \vdots \qquad \ddots \qquad \vdots \qquad \vdots \quad \vdots \\ \lambda.w_1\sigma_{1n} + \lambda.w_2\sigma_{2n} + ... + \lambda.w_i\sigma_n^2 - \frac{1}{2}(1-\lambda).r_n + a_1 = 0 \\ w_1 + w_2 + ... + w_n = 1 \end{cases}$$

$$\Leftrightarrow \text{AX} = \text{B}$$

Where: $A = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} & \ldots & \sigma_{1n} & 1 \\ \sigma_{12} & \sigma_2^2 & \sigma_{23} & \ldots & \sigma_{2n} & 1 \\ \sigma_{13} & \sigma_{23} & \sigma_3^2 & \ldots & \sigma_{3n} & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ \sigma_{1n} & \sigma_{2n} & \sigma_{3n} & \ldots & \sigma_n^2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$, $B = \begin{bmatrix} \frac{1}{2}(1-\lambda).r_1 \\ \frac{1}{2}(1-\lambda).r_2 \\ \frac{1}{2}(1-\lambda).r_3 \\ \vdots \\ \frac{1}{2}(1-\lambda).r_n \\ 1 \end{bmatrix}$, $X = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_i \\ a_1 \end{bmatrix}$

Solving the above linear system by X = A^{-1}B, we will get the solution for the initial problem (weight $w_1, w_2, ..., w_i$).

We have to bear it in mind that Lagrange Method is only valid in the case of short-selling allowed. In this paper, we will apply this method to solve Portfolio Optimization Problems on three unrestricted portfolios by using Excel as a tool for solving. Moreover, we also compare the results of Langrange Method with those of Excel Solver.

Table 6.1: Markvitz Model and Excel Solver in Unrestricted Risky Portfolio

		SD			**ER**			**Fitness Value**		
λ	**Method**	**8**	**15**	**30**	**8**	**15**	**30**	**8**	**15**	**30**
	Mark.M	19.76%	15.66%	13.82%	7.41%	9.99%	9.88%	3.91%	2.45%	1.91%
1	Excel Solver	19.76%	15.66%	13.82%	7.41%	9.99%	9.88%	3.91%	2.45%	1.91%
	Mark.M	20.24%	17.01%	19.02%	10.87%	17.93%	40.57%	2.60%	0.81%	-0.80%
0.9	Excel Solver	20.24%	17.01%	19.02%	10.87%	17.93%	40.57%	2.60%	0.81%	-0.80%
	Mark.M	22.09%	21.64%	32.47%	15.19%	27.85%	78.94%	0.86%	-1.82%	-7.35%
0.8	Excel Solver	22.09%	21.64%	32.47%	15.19%	27.85%	78.94%	0.86%	-1.82%	-7.35%
	Mark.M	26.01%	30.02%	52.23%	20.75%	40.60%	128.28%	-1.49%	-5.87%	-19.39%
0.7	Excel Solver	26.01%	30.02%	52.23%	20.75%	40.60%	128.28%	-1.49%	-5.87%	-19.39%
	Mark.M	32.90%	42.81%	79.56%	28.16%	57.60%	194.06%	-4.77%	-12.05%	-39.64%
0.6	Excel Solver	32.90%	42.81%	79.56%	28.16%	57.60%	194.06%	-4.77%	-12.05%	-39.64%
	Mark.M	44.13%	61.78%	118.34%	38.54%	81.41%	286.15%	-9.53%	-21.62%	-73.05%
0.5	Excel Solver	44.13%	61.78%	118.34%	38.54%	81.41%	286.15%	-9.53%	-21.62%	-73.05%
	Mark.M	62.39%	90.99%	176.84%	54.11%	117.12%	424.29%	-16.89%	-37.15%	-129.48%
0.4	Excel Solver	62.39%	90.99%	176.84%	54.11%	117.12%	424.29%	-16.89%	-37.15%	-129.48%
	Mark.M	94.16%	140.31%	274.59%	80.05%	176.64%	654.52%	-29.44%	-64.58%	-231.96%
0.3	Excel Solver	94.16%	140.31%	274.59%	80.05%	176.64%	654.52%	-29.44%	-64.58%	-231.96%
	Mark.M	159.05%	239.54%	470.33%	131.94%	295.67%	1114.98%	-54.96%	-121.77%	-449.56%
0.2	Excel Solver	159.05%	239.54%	470.33%	131.94%	295.67%	1114.98%	-54.96%	-121.77%	-449.56%
	Mark.M	355.64%	538.05%	1057.88%	287.61%	652.77%	2496.35%	-132.37%	-298.00%	-1127.61%
0.1	Excel Solver	355.64%	538.05%	1057.88%	287.61%	652.77%	2496.35%	-132.37%	-298.00%	-1127.61%

As we can see from the above table, the results obtained by using Lagrange Method are similar to these of Excel Solver in all values of λ. It means that Excel Solver can give us the same optimal solutions as Lagrange Method. Although Excel Solver and Lagrange Method can produce the same solution, Excel Solver is easier to implement and it is still valid in both cases of short selling which is allowed or not. Therefore, in this paper, we will use Lagrange Method as a standard method to evaluate the performance of PSO in unrestricted portfolios and Excel Solver to evaluate the performance of PSO in restricted portfolios.

In Excel, Solver is part of a suite of commands sometimes called what-if analysis tools. Thanks to Solver, you can obtain an maximum or minimum value for a formula in one cell called the objective cell subject to constraints, or requirements, on the values of other formula cells on a worksheet. Solver works with a group of cells which is called decision variable cells. These cells take part in calculating the formulas in the objective and constraint cells. Solver adjusts the values in the decision variable cells to satisfy the limits on constraint cells and gain the result you want for the objective cell.

The constraint, objective and decision variable cells and the formulas together form a Solver model; the final values obtained by Solver are a solution for this model. Solver uses many methods, from linear programming and nonlinear optimization to genetic and evolutionary algorithms, to find solutions.

After defining a problem for the Excel Solver, you can choose one of the following methods:

GRG Nonlinear. Generalized Reduced Gradient Nonlinear algorithm is applied to solve problems that are smooth nonlinear. For example, problems in which at least one of the constraints is a smooth nonlinear function of the decision variables.

LP Simplex. The Simplex LP Solving is a method which is based the Simplex algorithm which introduced by George Dantzig. This model is applied to solve Linear Programming problems. For example, problems consist of a single objective represented by a linear equation that must be maximized or minimized.

Evolutionary. This method is used to solve non-smooth problems, which are the most difficult type of optimization problems to handle since some of the functions are non-smooth or even discontinuous, and therefore it is hard to determine the direction in which a function is increasing or decreasing.

6.3 Experiments and Discussion

In order to evaluate the performance of PSO method, we compare the results of PSO method with those of another heuristic Algorithm, named GA and the results of Excel Solver and Lagrange Method. In this experiment, PSO method has been developed by using Matlab as a software tool for calculating. GA has been developed by using The Optimization Tool Box in Matlab, and Excel Solver and Lagrange Method are implemented on Microsoft Excel 2016.

In this paper, mean return, standard deviation and covariance matrix are estimated in one year rather than one day.

6.3.1 Sharpe Ratio Model

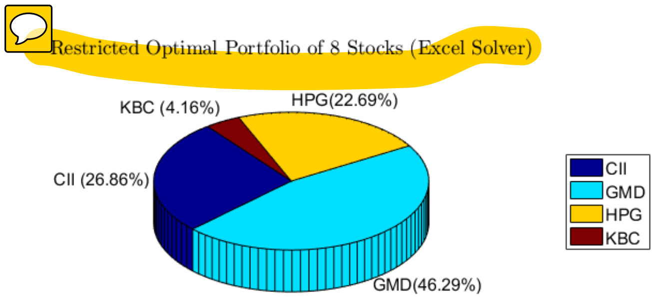
Table 6.2: The results of risky portfolios with 8 stocks by using PSO, GA, Excel Solver and Langrange Method

		Portfolio	
Method	**Item**	Restricted (%)	Unrestricted (%)
PSO	ER	21.69%	54.28%
	SD	27.75%	62.79%
	Sharpe Ratio	60.17%	78.48%
GA	ER	20.61%	41.83%
	SD	26.49%	49.24%
	Sharpe Ratio	58.93%	74.79%
Excel Solver/Langrange Method	ER	21.69%	54.28%
	SD	27.74%	62.79%
	Sharpe Ratio	60.17%	78.48%

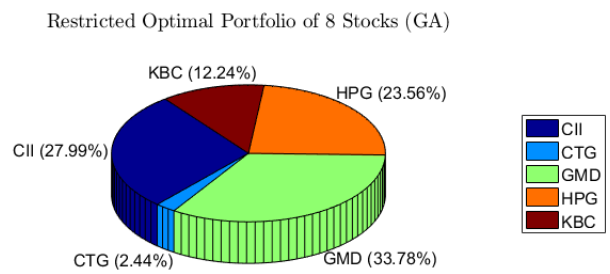
Table 6.2, **Table 6.4** and **Table 6.6** represent the results of risky portfolios with 8 stocks, 15 stocks and 30 stocks respectively in 2 cases by using different methods: PSO,

GA, Excel Solver and Lagrange Method.

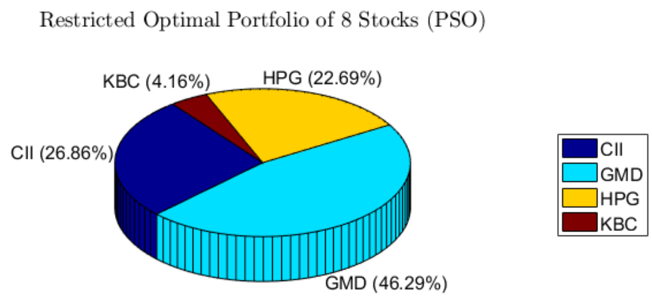
In risky portfolio with 8 stocks (**Table 6.2**), we can see that the values of Sharpe Ratio obtained by using PSO method are 60.17% (restricted portfolio) and 78.48% (unrestricted portfolio) which are similar to the values by using Excel Solver in restricted case and Lagrange Method in unrestricted case. However, these are 58.93% (restricted portfolio) and 74.79% (unrestricted portfolio) for GA. As we can see that the values of Sharpe Ratio obtained by using PSO method seem better than the values from the GA and are likely similar to those of Excel Solver in restricted portfolio and Lagrange Method in unrestricted portfolio. As we know that Sharpe Ratio indicates how much excess return you are receiving for extra risk that you incur and the purpose of this model is to maximize the Sharpe Ratio. Based on the results of PSO, Excel Solver and Lagrange Method, we can conclude that PSO has found the optimal solution exactly. It means that PSO method can help us to get the maximum excess return for extra risk we incur by adjusting the weight of each asset in portfolios with 8 stocks. We can also see that the ER and SD in unrestricted portfolios are always higher than those in restricted portfolios. The reason for this is that the short selling which is valid in unrestricted portfolios allows investors not only to earn higher return, but also to be exposed to higher risk at the same time.



(a) Excel Solver



(b) GA Method



(c) PSO Method

Figure 6.1: Sharpe Ratio Model: Weights for Restricted Portfolio with 8 stocks

Table 6.3: Sharpe Ratio Model: Weight of Asset for Unrestricted Portfolio with 8 stocks

Method	**Stocks**							
	BVH	CII	CTG	DPM	FPT	GMD	HPG	KBC
PSO	13.71%	72.74%	11.46%	-98.29%	-97.21%	100.00%	67.45%	30.16%
GA	1.95%	48.06%	-9.76%	-70.46%	-33.60%	86.22%	58.34%	19.23%
Lagrange Method	13.68%	72.69%	11.46%	-98.19%	-97.09%	100%	67.40%	30.05%

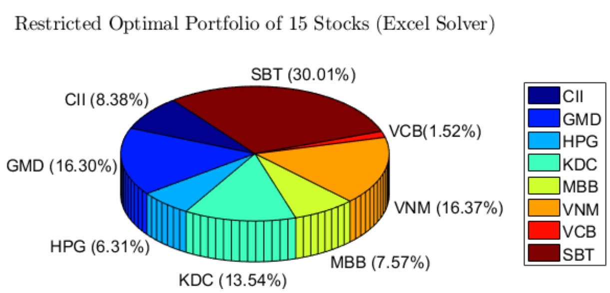
Figure 6.1 and **Table 6.3** show the stock allocation for a portfolio with 8 stocks.

Figure 6.1 represents the stock allocation of portfolio with 8 stocks obtained by PSO, GA and Excel Solver. We has decided that our restricted portfolio will have 8 stocks. However, in order to maximize the Sharpe Ratio, these methods indicated that we just need to invest in some of them. In **Figure 6.1**, the optimal allocation obtained by using PSO and Excel Solver indicates that in order to maximize the value of Sharpe Ratio, we just need to spend money on 4 stocks (CII, GMD, HPG and KBC) with the given weights of each stock (26.86%, 46.29%, 22.69% and 4.16% respectively) while GA suggests that we need to invest money in 5 stocks (CII with 27.99%, CTG with 2.44%, GMD with 33.78%, HPG 23.56% and KBC with 12.24%). From **Figure 6.1**, we can easily realize that the weight of each stock in portfolio using 3 above methods in are positive and less than 1 as a result representing the stock allocation for restricted portfolio.

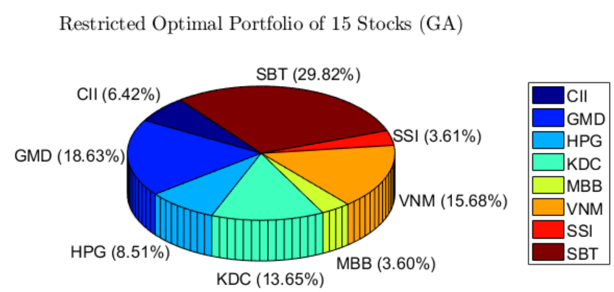
In **Table 6.2**, these weights can positive, negative or greater than 1 such as -9.76% for CTG (GA) and -98.29% for DPM (PSO) for representing the unrestricted portfolio. In order to maximize the Sharpe Ratio in this case, we need to buy some stocks and short sell other stocks. In **Table 6.2**, the optimal stock allocation by using PSO and Excel Solver indicates that we have to have short sell GMD and FPT (100% and 97% respectively), and buy the rest of stocks with the given weights in table while we have to short sell CTG, DMP and FPT (-9.76%, -70.46%, -33.06% respectively), and buy others with the given weight if we follow that of GA. With the optimal stock allocations in **Figure 6.1** and **Table 6.3**, our portfolio will get the values of Sharpe Ratio, SD and ER as given in **Table 6.2**.

Table 6.4: The results of risky portfolios with 15 stocks by using PSO, GA, Excel Solver and Langrange Method

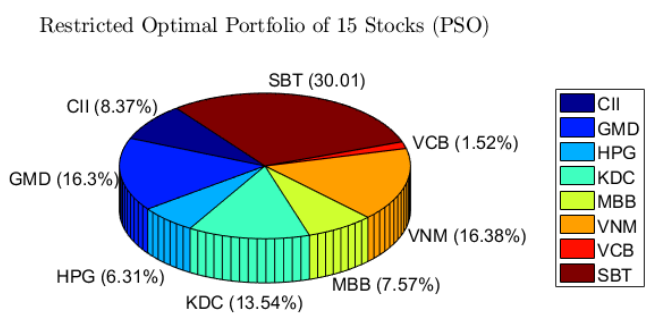
		Portfolio	
Method	**Item**	Restricted (%)	Unrestricted (%)
PSO	ER	20.69%	80.18%
	SD	19.20%	60.78%
	Sharpe Ratio	81.75%	123.69%
GA	ER	21.05%	65.30%
	SD	19.80%	51.40%
	Sharpe Ratio	81.05%	117.32%
Excel Solver/Langrange Method	ER	20.69%	80.18%
	SD	19.20%	60.78%
	Sharpe Ratio	81.74%	123.69%



(a) Excel Solver method



(b) GA method



(c) PSO method

Figure 6.2: Sharpe Ratio Model for Portfolio with 15

Similarly, in the risky portfolio with 15 stocks in **Table 6.4**, we can also see that the values of Sharpe Ratio obtained by using PSO method are 81.75% (restricted portfolio) and 123.69% (unrestricted portfolio), which are nearly similar to the values by using Excel Solver (81.74%) for restricted portfolio and the results of Lagrange Method (123.69%) for unrestricted portfolio while these are 81.05% (restricted portfolio) and 117.32% (unrestricted portfolio) for GA. Based on the results of PSO, Excel Solver and Lagrange Method, we can also conclude that PSO is likely effective for finding the optimal solution and allow us to get the maximum excess return for extra risk we incur by adjusting the weight of each asset in portfolio with 15 stocks. Hence, PSO is probably a great choice for solving portfolio optimization problem with 15 variables.

Table 6.5: Weight of Asset for unrestricted risky portfolio with 15 stocks

	Stocks							
Method	BVH	CII	CTG	DPM	FPT	GMD	HPG	KBC
PSO	-1.71%	30.36%	-41.45%	-96.66%	-90.50%	60.58%	33.32%	0.89%
GA	-4.24%	28.80%	-10.63%	-38.56%	-88.05%	59.84%	27.92%	-4.68%
Excel	-1.71%	30.36%	-41.45%	-96.66%	-90.50%	60.58%	33.32%	0.89%
	KDC	MBB	MSN	VNM	VCB	SSI	SBT	
PSO	40.24%	56.16%	-90.01%	60.08%	47.40%	21.99%	69.31%	
GA	31.67%	38.62%	-80.05%	39.45%	0.10%	48.73%	51.37%	
Excel	40.24%	56.16%	-90.01%	60.08%	47.40%	21.99%	69.31%	

Figure 6.2 shows the stock allocation of restricted portfolio with 15 stocks using PSO, GA and Excel. Again, we has decided that our restricted portfolio will have 15 stocks. However, in order to maximize the Sharpe Ratio, these methods indicate that we just need to invest in some of them. In **Figure 6.2**, the optimal allocations obtained by using PSO, Excel Solver and GA indicate that in order to maximize the value of Sharpe Ratio, we just need to spend money on 8 stocks (CII, GMD, HPG, KBC, MBB, VNM, VCB and SBT). Each method has the different weights for these 8 stocks. However, PSO and Excel Solver

have the nearly same weights of these 8 stocks.

Table 6.5 shows the stock allocation of urestricted portfolio with 15 stocks using PSO, GA and Lagrange Method. In **Table 6.5**, the optimal stock allocations by using PSO and Lagrange Method show that we need to short sell BVH, CTG, DPM, FPT and MSN (-1.71%, -41.45%, -96.66%,-90.5% and -90.01% respectively), and buy the rest of stocks with the given weights in the table while we have to short sell BVH, CTG, DPM, FPT, MSN and KBC (-4.24 %,-10.63%, -38.56%, -88.05% and -4.68% respectively) and buy others if we follow that of GA.

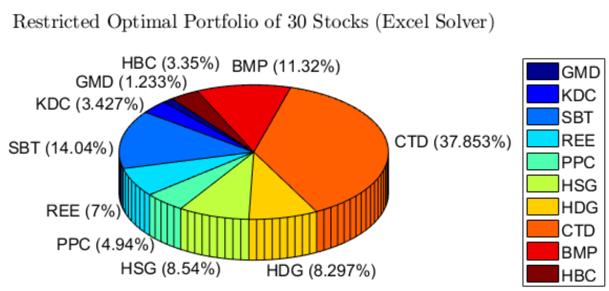
With the optimal stock allocations given in **Figure 6.2** and **Table 6.5**, our portfolio will get the values Sharpe Ratio, SD and ER as in **Table 6.4**.

Table 6.6: The results of risky portfolios with 30 stocks by using PSO, GA, Excel Solver and Lagrange Method

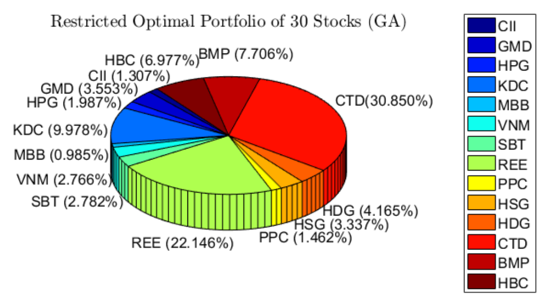
		Portfolio	
Method	**Item**	Restricted (%)	Unrestricted (%)
PSO	ER	34.08%	210.82%
	SD	21.06%	86.54%
	Sharpe Ratio	138.11%	237.68%
GA	ER	31.05%	197.60%
	SD	20.07%	86.41%
	Sharpe Ratio	129.77%	222.89%
Excel Solver/Lagrange Method	ER	34.08%	210.86%
	SD	21.06%	86.61%
	Sharpe Ratio	138.10%	237.68%

Similarly, in the risky portfolio with 30 stocks in **Table 6.6**, we can also see that the values of Sharpe Ratio obtained by using PSO method are 138.11%(restricted portfolio) and 237.68% (unrestricted portfolio), which are nearly similar to the values by using Excel Solver (138.10%) for restricted portfolio and the results of Lagrange Method (237.68%)

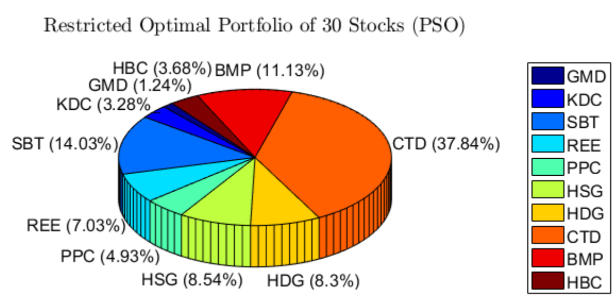
for unrestricted portfolio while these are 129.77% (restricted portfolio) and 222.89% (unrestricted portfolio) for GA. Based on the results of PSO, Excel Solver and Lagrange Method, we can also conclude that PSO is probably effective for finding the optimal solution for portfolio problems with 15 stocks. Hence, PSO is probably a great choice for solving portfolio optimization problem with 30 variables.



(a) Excel Solver method



(b) GA method



(c) PSO method

Figure 6.3: Sharpe Ratio Model for Portfolio with 30

Figure 6.3 (restricted portfolio) shows the stock allocations of restricted portfolio with 30 stocks using PSO, GA and Excel. Although we has decided that our restricted portfolio will have 30 stocks. However, in order to maximize the Sharpe Ratio, these methods indicate that we just need to invest in some of them. In **Figure 6.3**, the optimal allocations obtained by using PSO and Excel Solver indicate that in order to maximize the value of Sharpe Ratio, we have to only spend money on 10 stocks in 30 ones (GMD,KDC,SBT, REE, PPC, HSG, HDG, CTD, BMP and HBC) whereas that of GA suggest that we have to invest money in 14 stocks (CII, GMD, HPG, KDC, MBB, VNM, SBT, REE, PPC, HSG, HDG,

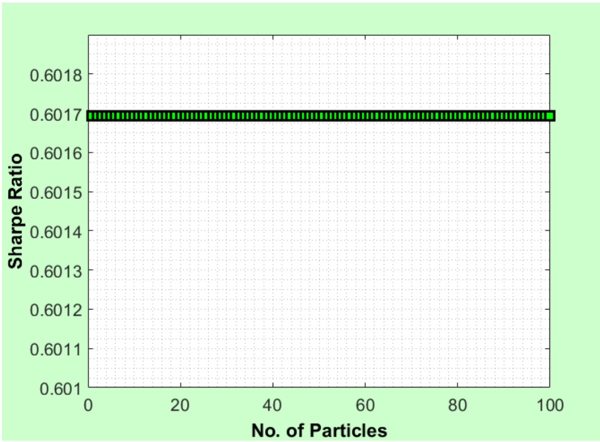
Table 6.7: Weights of unrestricted risky portfolio with 30 stocks

Method	Stocks							
	BVH	CII	CTG	DPM	FPT	GMD	HPG	KBC
PSO	15.07%	23.06%	-25.07%	-84.22%	-81.53%	37.53%	21.12%	36.79%
GA	-5.60%	32.63%	-36.38%	-63.63%	-69.90%	39.99%	-9.01%	40.66%
Lagrange Method	15.12%	23.12%	-25.01%	-84.36%	-81.65%	37.62%	21.11%	36.82%
	KDC	MBB	MSN	VNM	VCB	SSI	SBT	VIC
PSO	26.51%	54.61%	-78.67%	32.24%	36.32%	2.98%	41.64%	-64.74%
GA	32.90%	42.84%	-50.66%	-18.17%	65.47%	1.26%	42.18%	-59.21%
Lagrange Method	26.52%	54.60%	-78.69%	32.30%	36.30%	3.08%	41.68%	-64.78%
	REE	PPC	STB	PVD	PHR	KDH	ITA	HSG
PSO	85.79%	50.92%	-3.26%	-66.82%	22.00%	-20.04%	-88.36%	46.82%
GA	25.29%	62.40%	0.86%	-68.93%	30.10%	-3.05%	-87.80%	67.30%
Excel	85.75%	50.97%	-3.30%	-66.81%	21.99%	-20.08%	-88.38%	46.87%
	HDG	CTD	HAG	BMP	HBC	DMC		
PSO	32.61%	100.00%	-64.54%	41.36%	20.87%	-51.00%		
GA	35.92%	99.30%	-72.37%	52.80%	22.81%	-49.95%		
Lagrange Method	32.70%	100.00%	-64.65%	41.31%	20.89%	-51.04%		

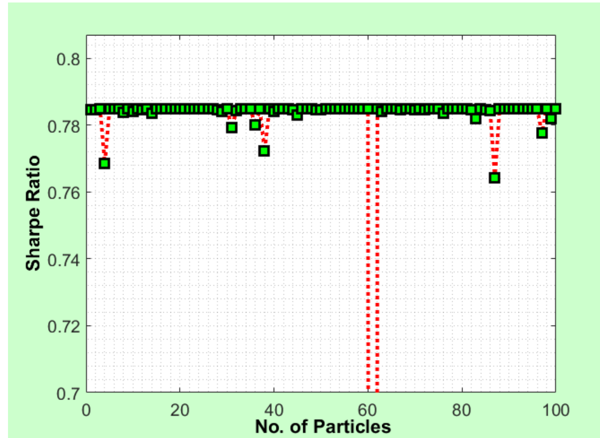
CTD, BMP and HBC). Each method has the different weights for these stocks. However, PSO and Excel Solver have the nearly same weights of these stocks.

In **Table 6.7**, the optimal stock allocations by using PSO and Lagrange Method show that we have to have short sell CTG, DPM, FPT, MSN, VIC, STB, PVD, KDH, HAG and DMC and buy the rest of stocks with the given weights in the table while we have to short sell BVH, CTG, DPM, FPT, HPG, MSN, VNM, CII, PVD, KDH, ITA and HAG and buy others if we follow that of GA.

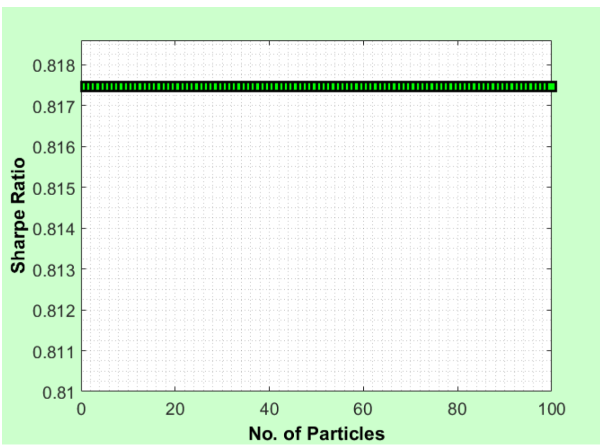
With the optimal stock allocations given in **Figure 6.3** and **Table 6.7**, our portfolio will get the values of Sharpe Ratio, SD and ER as in **Table 6.6**.



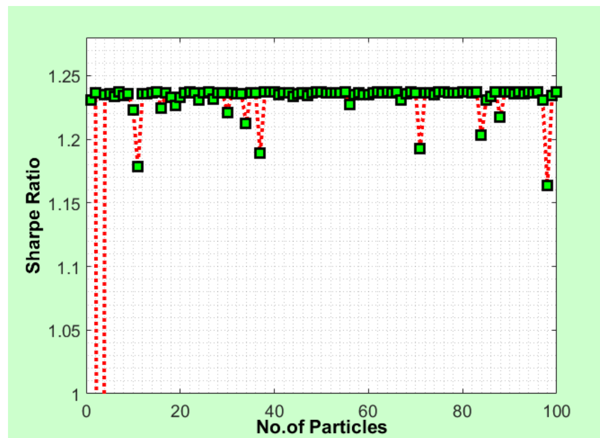
(a) Restricted Portfolio with 8 stocks



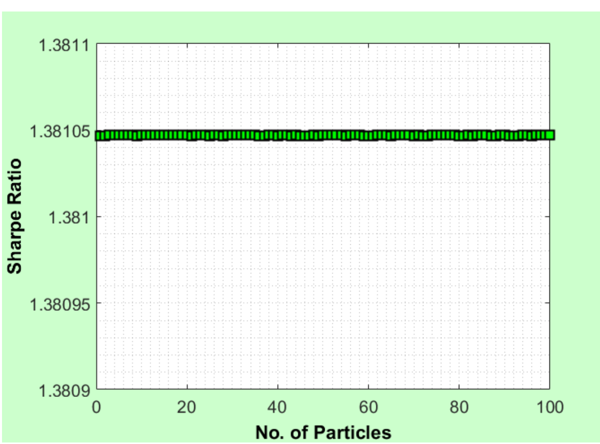
(b) Unrestricted Portfolio with 8 stocks



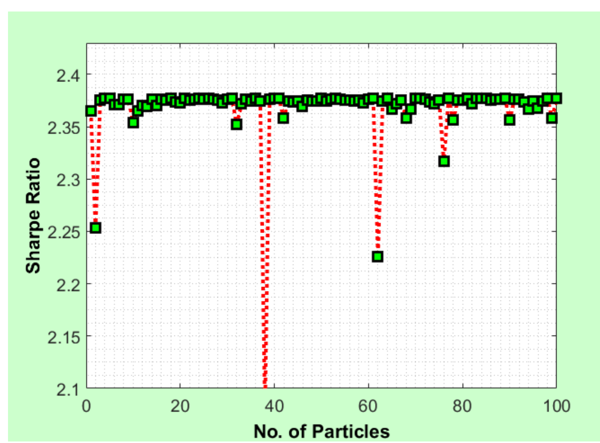
(c) Restricted Portfolio with 15 stocks



(d) Unrestricted Portfolio with 15 stocks

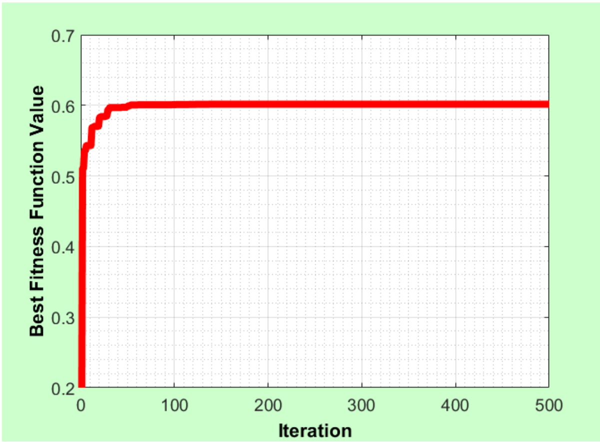


(e) Restricted Portfolio with 30 stocks

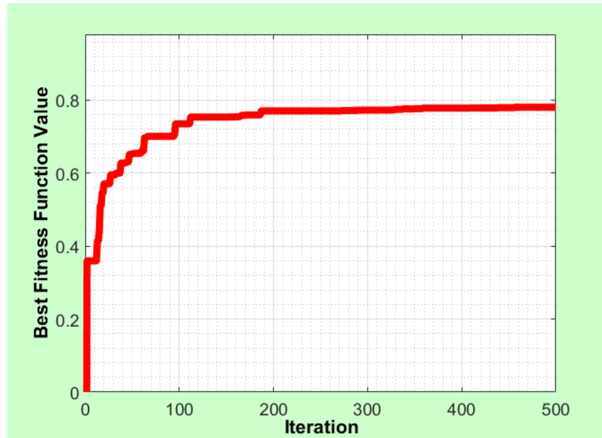


(f) Unrestricted Portfolio with 30 stocks

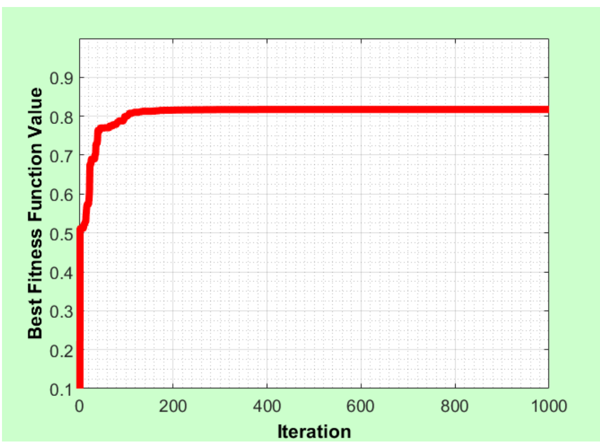
Figure 6.4: PSO particles updating process.



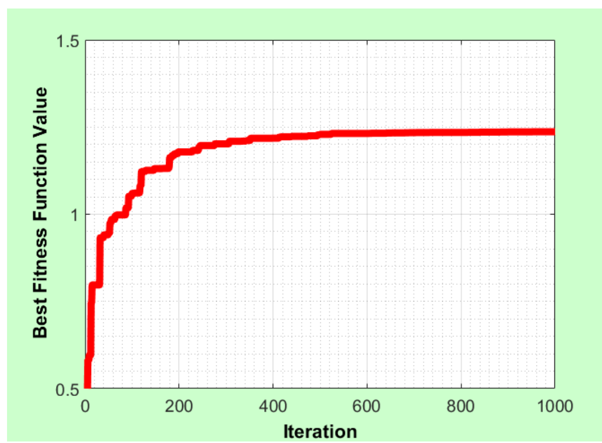
(a) Restricted Portfolio with 8 stocks



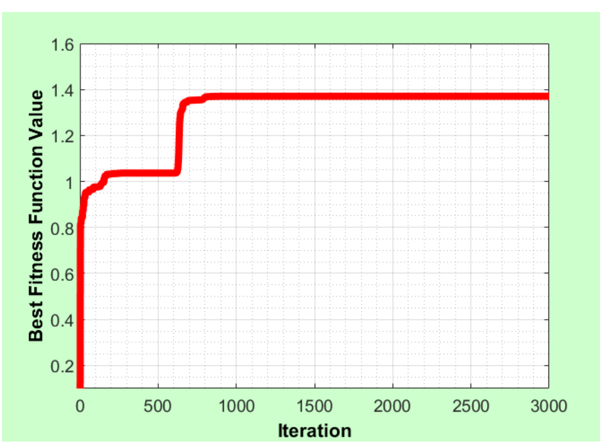
(b) Unrestricted Portfolio with 8 stocks



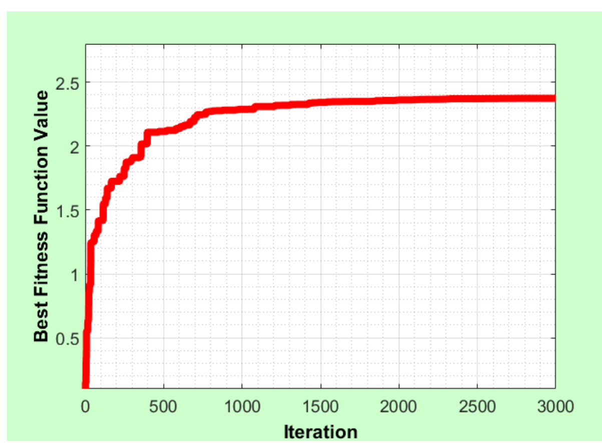
(c) Restricted Portfolio with 15 stocks



(d) Unrestricted Portfolio with 15 stocks

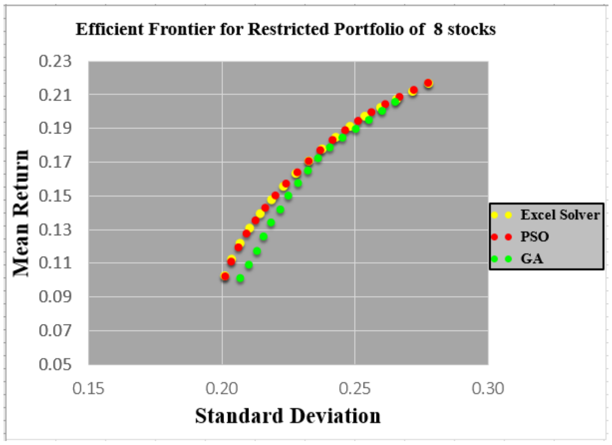


(e) Restricted Portfolio with 30 stocks

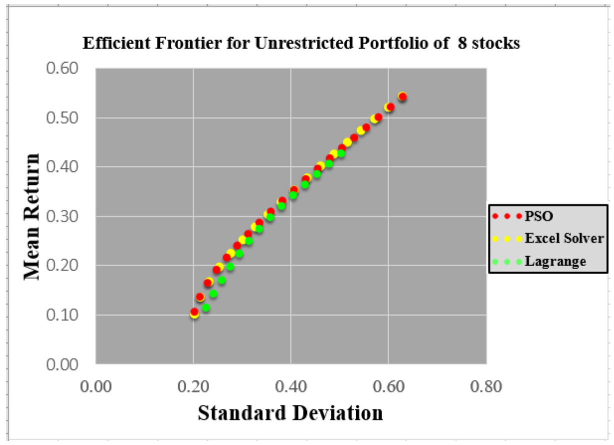


(f) Unrestricted Portfolio with 30 stocks

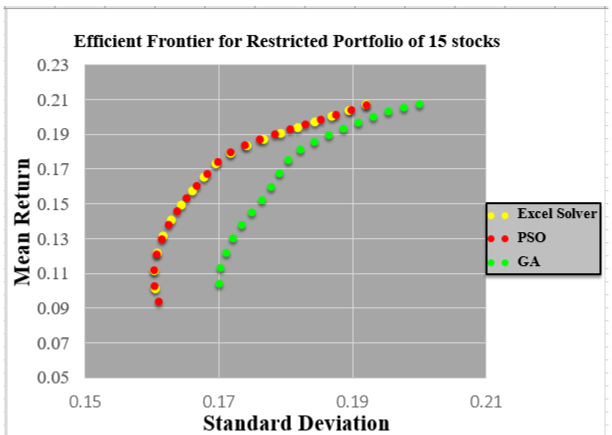
Figure 6.5: PSO convergence characteristic



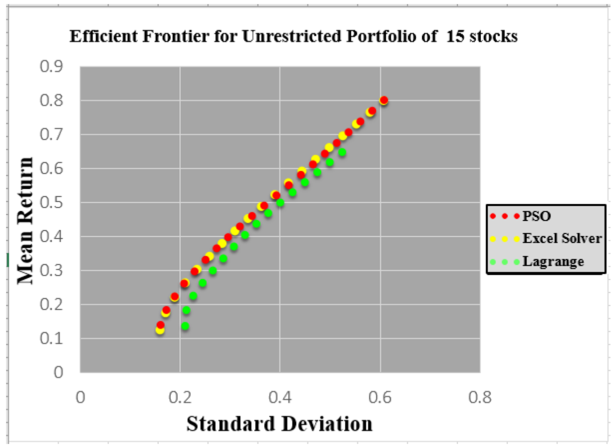
(a) Restricted Portfolio with 8 stocks



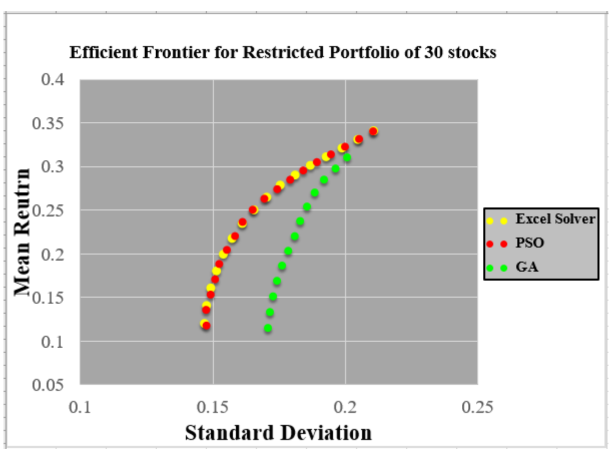
(b) Unrestricted Portfolio with 8 stocks



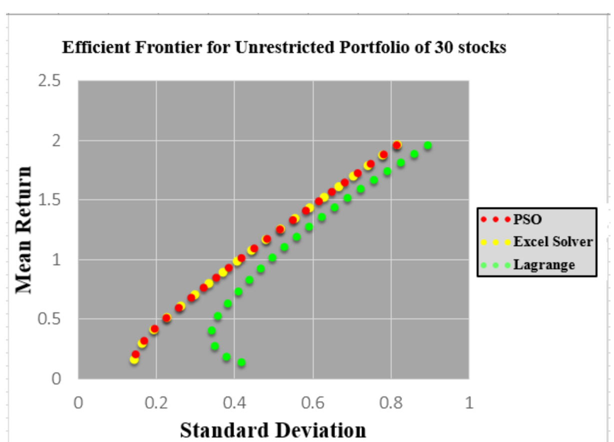
(c) Restricted Portfolio with 15 stocks



(d) Unrestricted Portfolio with 15 stocks



(e) Restricted Portfolio with 30 stocks



(f) Unrestricted Portfolio with 30 stocks

Figure 6.6: The efficient frontier of the portfolio gotten from PSO, GA and Excel Solver

The updating process of PSO's particles for portfolio optimization for 6 portfolios, including three restricted portfolios and three unrestricted portfolios, and with the termination condition of 500 steps (iterations) for portfolio with 8 stocks, 1000 steps for portfolio with 15 stocks and 3000 steps for portfolio with 30 stocks, is given as in **Figure 6.4**, where the number of particles is 100 and Sharpe Ratio is chosen as a fitness function. This figure illustrates the behavior of 100 particles after the given number of iterations in 2 dimensional space.

As we discussed in Chapter 4, in the algorithm of PSO, it starts with the initialization of a population of random particles (solutions). In the search space, every particles will modify their positions to move towards the optimal value over each iteration. In **Figure 6.4**, comparing **Figure 6.4 (a)** and **Figure 6.4 (b)**, we can see that 100 particles in restricted portfolio of 8 stocks converge to the best fitness value (the optimal value) after 500 steps. However, this does not occur in the unrestricted portfolio with 8 stocks in which a few particles such as (3rd, 88th and 38th particles) tend to move towards the optimal value, but are still lying below the optimal value, not reach it totally. This also happens in the portfolio of 15 stocks (**Figure 6.7 (c)** and **Figure 6.7 (d)**), in the case of the unrestricted portfolio with 15 stocks, the number of particles lying below the optimal value is more than those in portfolio with 8 stocks. In portfolio with 30 stocks (**Figure 6.7 (e)** and **Figure 6.7 (f)**), most particles in restricted portfolio converge to the optimal value, but the small number of particles need more steps for reaching that value totally. However, in the unrestricted portfolio, the number of particles lying below the optimal value is more than those in portfolio with 15 and 8 stocks. Therefore, we can conclude that the more variables the portfolio problems have, the more iterations the particles need to converge to the optimal value.

Figure 6.5 shows the the process of convergence of the optimal value of PSO method in 6 portfolios. Comparing three unrestricted portfolios (**Figure 6.5 (a),(c),(e)**) with three restricted portfolios (**Figure 6.5 (b),(d),(f)**), we can easily see that the convergence of the restricted portfolios seems better than that of the unrestricted portfolios. The reason for this is that the fitness functions of restricted portfolio have more constraints, which results in the fact that the particles will move in the smaller search space. As a result, in PSO, convergence to the best fitness value in restricted portfolios will happend faster than that in unrestricted portfolios. Regarding to the strategy for choosing portfolio for optimization purposes, it is suggested that PSO method which is applied in restricted portfolios will be more efficient than that in unrestricted portfolios because of requiring less steps for convergence to the optimal value. In other words, getting the optimal value in restricted porfolios by using PSO will be faster than getting it in unrestricted portfolios.

The termination condition plays a very important role in finding the optimal risky portfolio, because we have to make balance between efficiency and precision. According to many testing about PSO method, the termination condition is 500 steps for the portfolios of 8 stocks, 1000 steps for the portfolios of 15 stocks, and 3000 steps for the portfolios of 30 stocks. In the GA method, the termination condition is 500 generations for the portfolios of 8 stocks, 2000 generations for the portfolios of 15 stocks, and 4000 generations for the portfolios of 30 stocks. Moreover, the input parameters for GA method are choosen for these 6 portfolios as follows: Population type = Double vector; Population size = 100; chromosome length = 64-bit; Scaling function = Rank; Selection function = Stochastic uniform; Elite Count = 0.05∗PopulationSize; Crossover Rate = 0.8; Mutation Function = Constraint Dependent; Fraction = 0.2; Iterval = 20; and Termination Condition = 2000 generations.

After finding a set of six optimal portfolios obtained by using different methods: PSO, GA, Excel Solver and Lagrange Method, we can trade out the efficient frontier curves of these portfolios in **Figure 6.6**. We can easily see that the efficient frontier curves obtained by PSO method are nearly above the other curves obtained by GA in both cases of portfolios. And, these curves by PSO also overlap with the curves obtained by Excel Solver in restricted portfolios and overlap with those by Lagrange Method in unrestricted portfolio. As we know, the efficient frontier curve represents the optimal combinations between standard deviation and return of the optimal portfolios. Based on the efficient frontier curves of PSO, Excel Solver and Lagrange Method, we can make a conclusion that PSO method is likely efficient in finding the efficient fronttier curves and allows us to get maximum level of mean return for the same amount of risk or get minimum level of risk for the same amount of return. On the other hand, we can see that the efficent frontier curves of GA are always lower than the others in all cases. In portfolio with 8 stocks, the differences between the curves of GA and those of PSO and Excel are not large. However, these differences become larger when the size of portfolio is increasing. It seems that GA is not an effective method when it is applied to solve problems with high dimension. The larger the number of variables in a problem is, the less effective GA become to solve this problem.

6.3.2 Efficient Frontier Model

According to the efficient frontier model, the results of the optimal risky portfolios of 8 stocks, 15 stocks and 30 stocks developed by using PSO, GA, Excel Solver methods for restricted risky portfolios are given in the **Table 6.8** and using PSO, GA, Lagrange methods for unrestricted risky portfolios are shown in **Table 6.9**. For each value of λ, we will have a different fitness function and a different optimal value. The purpose of this model is to minimize the trade-off between risk and return of a portfolio, it means the lower is better. As we can see from **Table 6.8** that when the value of λ runs from 0 to 1, all the fitness values obtained by using PSO methods are similar to that of Excel Solver and lower than those of GA. Similarly, in **Table 6.9**, the fitness values obtained by PSO are also similar to those of Lagrange Method and lower than those of GA. It is clear that in both types of portfolios, portfolio standard deviation decreases and the mean return of the portfolio increases when the number of the stocks increases. The reason for this is diversification. According to the Morkovitz Theory, the risk associated with the whole portfolio and the return of portfolio can be considerably reduced by containing a wide variety of investment assets in portfolio. The higher number of assets in portfolio, the better the risk and return of the portfolio will be. Therefore, the best fitness values are also

Table 6.8: Restricted Risky Portfolio

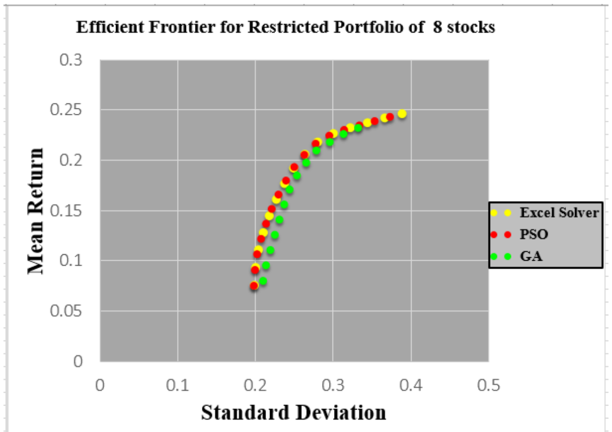
		SD			**ER**			**Fitness Value**		
λ	**Method**	**8**	**15**	**30**	**8**	**15**	**30**	**8**	**15**	**30**
1	PSO	19.80%	16.03%	14.73%	7.56%	10.83%	12.41%	3.92%	2.57%	2.16%
	GA	19.97%	16.95%	31.07%	8.06%	12.03%	15.63%	3.99%	2.87%	3.03%
	Excel	19.80%	16.03%	14.70%	7.56%	10.83%	11.73%	3.92%	2.57%	2.16%
0.9	PSO	20.27%	17.22%	16.64%	10.91%	17.98%	25.51%	2.61%	0.87%	-0.06%
	GA	20.09%	18.01%	33.96%	8.90%	17.80%	25.19%	2.74%	1.14%	0.56%
	Excel	20.27%	17.22%	16.74%	10.91%	17.98%	25.77%	2.61%	0.87%	-0.06%
0.7	PSO	24.74%	18.88%	22.74%	19.07%	20.42%	36.13%	-1.44%	-3.63%	-7.22%
	GA	25.41%	19.16%	55.80%	19.68%	20.03%	27.98%	-1.39%	-3.44%	-5.81%
	Excel	24.74%	18.88%	22.75%	19.07%	20.42%	36.13%	-1.44%	-3.63%	-7.22%
0.5	PSO	27.92%	22.92%	29.70%	21.79%	22.95%	41.59%	-7.00%	-8.85%	-16.38%
	GA	27.05%	21.42%	98.10%	21.13%	21.66%	29.03%	-6.91%	-8.54%	-12.61%
	Excel	27.92%	22.92%	29.70%	21.79%	22.95%	41.59%	-7.00%	-8.85%	-16.38%
0.3	PSO	31.35%	28.22%	34.68%	23.05%	24.89%	43.71%	-13.19%	-15.03%	-26.99%
	GA	28.83%	25.11%	141.15%	21.95%	22.54%	31.05%	-12.88%	-13.89%	-20.48%
	Excel	31.35%	28.22%	34.68%	23.05%	24.89%	43.71%	-13.19%	-15.03%	-26.99%
0.1	PSO	38.87%	28.49%	36.44%	24.64%	24.92%	44.20%	-20.67%	-21.61%	-38.45%
	GA	30.65%	25.26%	166.27%	22.81%	22.63%	33.28%	-19.59%	-19.73%	-29.43%
	Excel	38.87%	28.49%	36.44%	24.64%	24.92%	44.20%	-20.67%	-21.61%	-38.45%
0	PSO	38.87%	34.22%	36.44%	24.64%	25.05%	44.20%	-24.64%	-25.05%	-44.20%
	GA	32.79%	26.54%	176.07%	23.41%	22.80%	35.43%	-23.41%	-22.80%	-35.43%
	Excel	38.87%	34.22%	36.44%	24.64%	25.05%	44.20%	-24.64%	-25.05%	-44.20%

better when the number of stocks increases. Morever, thanks to the short selling, mean return and standard deviation in unrestricted portfolios are higher than those in restricted portfolios. Based on the results of PSO, Excel and Lagrange, we can realize that PSO is probably an effective method for providing the minimum trade-offs between risk and return of portfolio. Again, the differences between the results of GA and the those of PSO, Excel Solver and Lagrange are also rise when the size of portfolio increases.

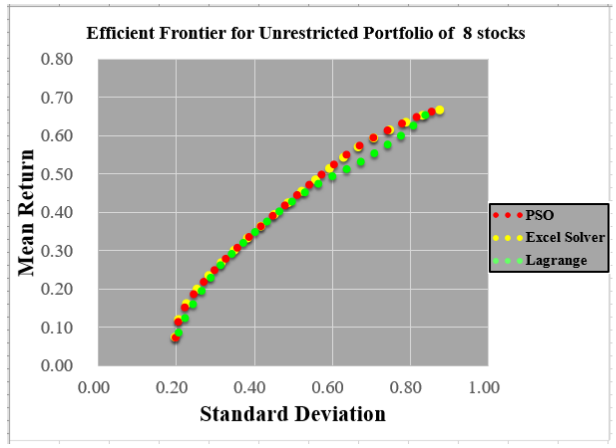
After finding sets of the optimal combinations between mean return and standard deviation with different values of λ of six portfolios by usin methods: PSO, GA, Excel Solver and Lagrange Method, we can trade out the continuous curves, called the efficient frontier in **Figure 6.7**. From all parts of this figure, it is easy to see that the efficient frontier curves obtained by PSO method are above the other curves obtained by GA and overlap with the curves of Excel Solver in restricted case. Also, this curves by PSO are above the curves of GA and overlap with ones of Lagrange Method in unrestricted case. Based on the efficient frontier curves of PSO, Excel Solver and Lagrange Method, we can make a conclusion that PSO method is likely efficient in finding the efficient fronttier curves and allows us to get maximum level of mean return for the same amount of risk or get minimum level of risk for the same amount of mean return. On the other hand, we can see that the efficent frontier curves of GA are always lower than those of PSO, Excel and Lagrange. In portfolio with 8 stocks, the differences between the curves of GA and those of other methods are not large. However, these differences become larger when the size of portfolio is increasing. It seems that GA is not an effective method to solve portfolio problems with high dimension.

Table 6.9: Unrestricted Risky Portfolio

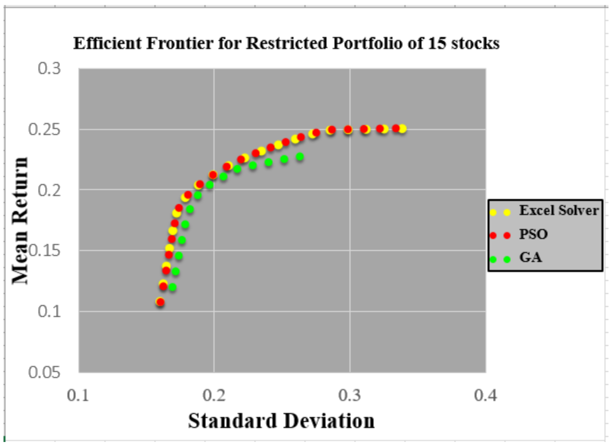
		SD			ER			Fitness Value		
λ	**Method**	**8**	**15**	**30**	**8**	**15**	**30**	**8**	**15**	**30**
1	PSO	19.76%	15.66%	13.82%	7.41%	9.99%	9.88%	3.91%	2.45%	1.91%
	GA	20.41%	19.81%	25.11%	8.60%	11.19%	17.36%	4.17%	3.92%	6.30%
	Lagrange	19.76%	15.66%	13.82%	7.41%	9.99%	9.88%	3.91%	2.45%	1.91%
0.9	PSO	20.24%	17.01%	19.02%	10.87%	17.93%	40.57%	2.60%	0.81%	-0.80%
	GA	21.12%	20.51%	28.40%	10.40%	15.14%	42.53%	2.97%	2.27%	3.00%
	Lagrange	20.24%	17.01%	19.02%	10.87%	17.93%	40.57%	2.60%	0.81%	-0.80%
0.7	PSO	26.01%	30.02%	52.23%	20.75%	40.60%	128.28%	-1.49%	-5.87%	-19.39%
	GA	29.49%	36.06%	55.80%	24.11%	43.48%	119.71%	-1.15%	-3.94%	-14.12%
	Excel	26.01%	30.02%	52.23%	20.75%	40.60%	128.28%	-1.49%	-5.87%	-19.39%
0.5	PSO	44.13%	61.78%	109.36%	38.54%	81.41%	263.25%	-9.53%	-21.62%	-71.83%
	GA	43.89%	49.41%	98.10%	38.32%	60.51%	209.94%	-9.53%	-18.05%	-56.85%
	Lagrange	44.13%	61.78%	109.36%	38.54%	81.41%	263.25%	-9.53%	-21.62%	-71.83%
0.3	PSO	68.97%	78.21%	153.79%	58.52%	97.52%	339.67%	-26.69%	-49.92%	-166.82%
	GA	66.85%	72.95%	141.15%	53.09%	83.19%	289.65%	-23.76%	-42.26%	-142.99%
	Lagrange	68.97%	78.21%	153.79%	58.52%	97.52%	339.67%	-26.69%	-49.92%	-166.82%
0.1	PSO	88.17%	101.96%	186.82%	66.97%	106.76%	367.32%	-52.50%	-85.69%	-295.68%
	GA	77.92%	79.11%	166.27%	60.41%	87.27%	323.32%	-48.30%	-72.28%	-263.34%
	Lagrange	88.17%	101.96%	186.82%	66.97%	106.76%	367.32%	-52.50%	-85.69%	-295.68%
0	PSO	88.17%	124.42%	194.84%	66.97%	109.16%	369.71%	-66.97%	-109.16%	-369.71%
	GA	83.84%	91.20%	176.07%	65.59%	95.32%	331.94%	-65.59%	-95.32%	-331.94%
	Lagrange	88.17%	124.42%	194.84%	66.97%	109.16%	369.71%	-66.97%	-109.16%	-369.71%



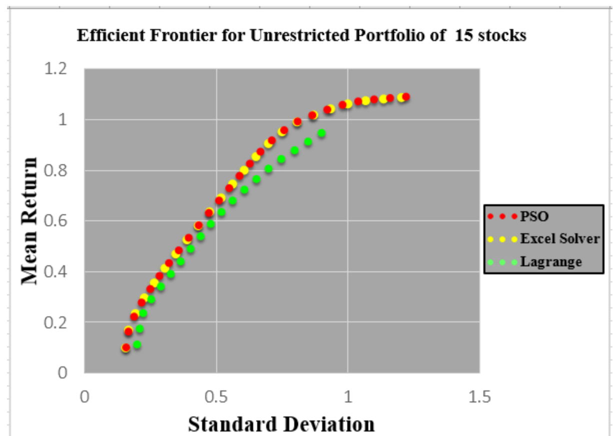
(a) Restricted Portfolio with 8 stocks



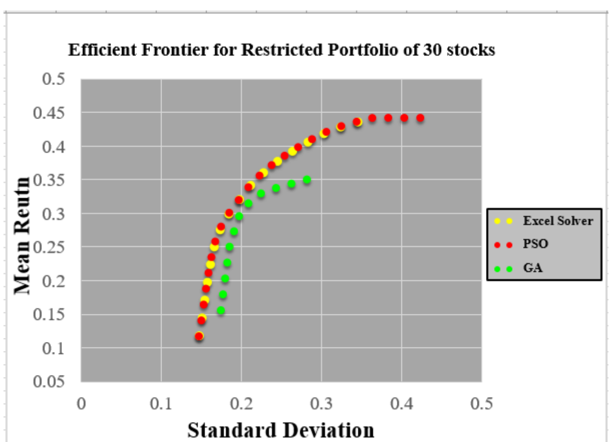
(b) Unrestricted Portfolio with 8 stocks



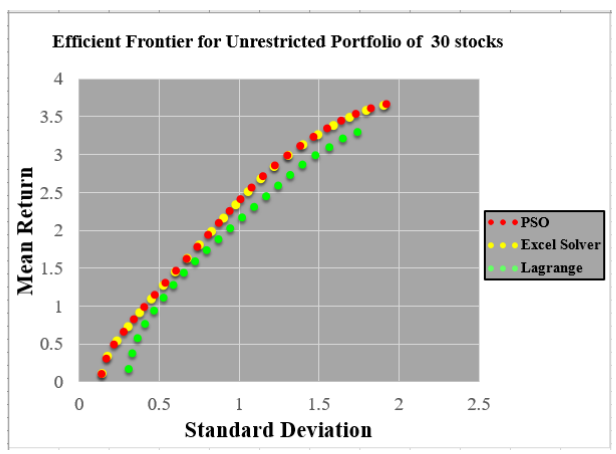
(c) Restricted Portfolio with 15 stocks



(d) Unrestricted Portfolio with 15 stocks



(e) Restricted Portfolio with 30 stocks



(f) Unrestricted Portfolio with 30 stocks

Figure 6.7: The efficient frontier of the portfolio gotten from PSO, GA and Excel Solver

After having the sets of optimal portfolios on two models: Sharpe Ratio (SR) model and Efficient Frontier (EF) model by three different methods, including: Excel Solver, GA and PSO for restricted portfolios, and GA, PSO and Lagrange Method for unrestricted portfolios. Based on the results of the experiment on three unrestricted portfolios and three restricted portfolios, we can make a conclusion that PSO is a good and effective method in general. Through this paper, PSO method represents the great efficiency and effectiveness of handling constrained optimization problems with high-dimension.

Chapter 7

Conclusion

One of the fundamental principles of finance investment is diversification. This term means that investors try to mix a wide variety of investment assets such as stock, bond and derivative security in their portfolios in order to minimize the portfolio risk and to maximize return of portfolio. This thesis concentrates on handling the problems of portfolio optimization which is one of the most popular problems in finance investment management. A meta-heuristic method, called Partcile Swarm Optimization, has been developed and applied to solve the investment portfolio optimization problems, where the objective function are based on Markowitz Model (Efficient Frontier Model) and Sharpe Model, and the constraints are also based on two these models and short selling requirement.

The PSO method has many similarites to other optimization methods inspired by biology. Like GA method, PSO is also a population-based optimization technique and it works by initializing a population of possibly random solutions and finding the optimal fitness value, then updating that value over a serie of iterations (steps). Unlike GA method, PSO does have an explicit selection process because all particles still exist over time. Instead, a memory of the personal best position or the global best postion plays an alternative role as selection. In order to evaluate the perfomance of PSO, we must make a valid comparison with other methods. The test problems of six different portfolios, including three unrestricted portfolios and three restricted portfolios are solved by using PSO and the results of these problems obtained by PSO are used for comparing with the results of Genetic Algorithms (GA), Excel Solver and Lagrange Method. These experiments and comparisons demonstrated the effectiveness of the PSO method.

The main learning mechanisms of PSO method is a metaphor of social behavior in which an individual in a swarm found a better solution and other members will copy that solution for improving the optimal solution of the entire swarm. Of course, these learning mechanisms exist in many fields in reality such as business and construction. Good strategies of business, good constructions of buildings, and other good strategies in different fields will stimulate imitation and this leads to subsequent adjustments for better verisons. Therefore, Particle Swarm method will be a successful and influential optimization tool to apply into a variety of applications, and this thesis proved that PSO is a potential method to use for application to financial modeling, that is portfolio optimization in this paper.

Future research may be conducted to further investigate the application of some derived models or hybrid models of PSO methods to other investment strategy problems such as tracking the benchmark index, rebalance and so on. Others further investigations may be conducted on PSO method such as taking new values for input parameter, inventing a new parameter, and so on for improving the efficiency, precision and convergence of this method to cope with portfolios with the large number of assets in investment management.

References

[1] Chang, T. J., Meade, N., Beasley, J. E., & Sharaiha, Y. M. (2000). Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research, 27* (13), 1271-1302.

[2] Doerner, K., Gutjahr, W. J., Hartl, R. F., Strauss, C., & Stummer, C. (2004). Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. *Annals of operations research, 131* (1), 79-99.

[3] Giovanis, E. (2009). The Arbitrage Pricing Theory and the Capital Asset Pricing Models and Artificial Neural Networks Modeling with Particle Swarm Optimization (PSO).

[4] Tiryaki, F., & Ahlatcioglu, B. (2009). Fuzzy portfolio selection using fuzzy analytic hierarchy process. *Information Sciences, 179* (1), 53-69.

[5] Jeurissen, R., & van den Berg, J. (2005). Index tracking using a hybrid genetic algorithm. In *Computational Intelligence Methods and Applications, 2005 ICSC Congress on* (pp. 6-pp). IEEE.

[6] Chan, M. C., Wong, C. C., Tse, W., Cheung, B., & Tang, G. (2002). Artificial intelligence in portfolio management. *Intelligent Data Engineering and Automated LearningIDEAL 2002*, 159-166.

[7] Casas, C. A. (2001). Tactical asset allocation: an artificial neural network based model. In *Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on* (Vol. 3, pp. 1811-1816). IEEE.

[8] Markowitz, H. (1952). Portfolio selection. *The journal of finance* , 7(1), 77-91.

[9] Sharpe, W. F. (1966). Mutual fund performance. *The Journal of business* , 39(1), 119-138.

[10] Kennedy, R. (1995, November). J. and Eberhart, Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks IV, pages* (Vol. 1000).

[11] Alam, M. N. (2016). Particle Swarm Optimization: Algorithm and its Codes in MATLAB.

[12] Eberhart, R. C., & Shi, Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on* (Vol. 1, pp. 84-88). IEEE.

[13] Shi, Y., & Eberhart, R. C. (1998, March). Parameter selection in particle swarm optimization. In *International conference on evolutionary programming* (pp. 591-600). Springer, Berlin, Heidelberg.

[14] Van Den Bergh, F. (2007). *An analysis of particle swarm optimizers* (Doctoral dissertation, University of Pretoria).

[15] Van den Bergh, F., & Engelbrecht, A. P. (2006). A study of particle swarm optimization particle trajectories. *Information sciences, 176(8),* 937-971.

[16] Karwowski, J., Okulewicz, M., & Legierski, J. (2013, September). Application of particle swarm optimization algorithm to neural network training process in the localization of the mobile terminal. In *International Conference on Engineering Applications of Neural Networks* (pp. 122-131). Springer, Berlin, Heidelberg.

[17] Turki, M., Bouzaida, S., Sakly, A., & M'Sahli, F. (2012, March). Adaptive control of nonlinear system using neuro-fuzzy learning by PSO algorithm. In *Electrotechnical Conference (MELECON), 2012 16th IEEE Mediterranean* (pp. 519-523). IEEE.

[18] Oliveira, M. V. D., & Schirru, R. (2009). Applying particle swarm optimization algorithm for tuning a neuro-fuzzy inference system for sensor monitoring *Progress in Nuclear Energy, 51(1),* 177-183.

[19] Hereford, J. M., Siebold, M., & Nichols, S. (2007, April). Using the particle swarm optimization algorithm for robotic search applications. In *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE* (pp. 53-59). IEEE.

[20] Li, W., & Wang, G. Y. (2010, October). Application of improved PSO in mobile robotic path planning. In *Intelligent Computing and Integrated Systems (ICISS), 2010 International Conference on*(pp. 45-48). IEEE.

[21] Seidy, E. E. (2016). A New Particle Swarm Optimization Based Stock Market Prediction Technique. *publicationin International Journal of Advanced Computer Science and Applications,* 7(2)

[22] Jamous, R. A., Seidy, E. E., & Bayoum, B. I. (2016). A Novel Efficient Forecasting of Stock Market Using Particle Swarm Optimization with Center of Mass Based Technique. *International Journal of Advanced Computer Science and Applications,* 7(4), 342-347.

[23] Tripathy, A. K., Mohapatra, S., Beura, S., & Pradhan, G. (2011). Weather Forecasting using ANN and PSO. *International J. Scientific & Eng. Res, 2,* 1-5.

[24] Papagianni, C., Papadopoulos, K., Pampas, C., Tselikas, N. D., Kaklamani, D. T., & Venieris, I. S. (2008, October). Communication network design using particle swarm optimization. In *Computer Science and Information Technology, 2008. IMCSIT 2008. International Multiconference on* (pp. 915-920). IEEE.

[25] Al-Awami, A. T., Abido, M. A., & Abdel-Magid, Y. L. (2007). Application of PSO to design UPFC-based stabilizers. In *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization.* InTech.

[26] Khan, A., Bawane, N. G., & Bodkhe, S. (2010). An analysis of particle swarm optimization with data clustering-technique for optimization in data mining. *IJCSE) International Journal on Computer Science and Engineering, 2(07),* 2223-2226.

[27] Szabo, A., & de Castro, L. N. (2013). A constructive data classification version of the particle swarm optimization algorithm. *Mathematical Problems in Engineering, 2013.*

[28] Khajeh, M., Kaykhaii, M., & Sharafi, A. (2013). Application of PSO-artificial neural network and response surface methodology for removal of methylene blue using silver nanoparticles from water samples. *Journal of Industrial and Engineering Chemistry, 19(5),* 1624-1630.

[29] Gabbouj, M. (2010). The New Digital Signal Processing Technology: Particle Swarm Optimization. In *2010 International Workshop on Information Communnication Technology, ICT 2010, Bangkok, Thailand, August 24-25..*

[30] Tsai, M. C., Chen, K. H., Su, C. T., & Lin, H. C. (2012). An application of PSO algorithm and decision tree for medical problem. In *2nd International Conference on Intelligent Computational Systems (ICS'2012)* (pp. 124-126).

[31] Benninga, S. (2008). Financial modeling. *MIT Press Books, 1.*

Appendix A

Stock Index

A.1 Porfolio with 30 stocks

1. Bao Viet Holdings (HOSE: BVH)
2. Ho Chi Minh City Infrastructure Investment JSC (HOSE: CII)
3. Vietnam Joint Stock Commercial Bank for Industry and Trade (HOSE:CTG)
4. Petrovietnam Fertilizer and Chemicals Corporation (HOSE: DPM)
5. FPT Corporation (HOSE: FPT)
6. Gemadept Corporation (HOSE: GMD)
7. Hoa Phat Group Joint Stock Company (HOSE: HPG)
8. Kinh Bac City Development Share Holding Corporation (HOSE: KBC)
9. KIDO Group (HOSE: KDC)
10. Military Commercial Joint Stock Bank (HOSE: MBB)
11. Masan Group Corporation (HOSE: MSN)
12. Viet Nam Dairy Products Joint Stock Company (HOSE: VNM)
13. Bank for Foreign Trade of Vietnam (HOSE: VCB)
14. Sai Gon Securities Incorporation (HOSE: SSI)

15. Thanh Thanh Cong Tay Ninh Joint Stock Company (HOSE: SBT)
16. Vingroup Joint Stock Company (HOSE: VIC)
17. Refrigeration Electrical Engineering Corporation (HOSE: REE)
18. Pha Lai Thermal Power Joint Stock Company (HOSE: PPC)
19. Sai Gon Thuong Tin Commercial Joint Stock Bank (HOSE: STB)
20. PetroVietnam Drilling and Well Services Corporation (HOSE: PVD)
21. Phuoc Hoa Rubber Joint Stock Company (HOSE: PHR)
22. Khang Dien Investment and Trading House JSC (HOSE: KDH)
23. Tan Tao Investment and Industry Corporation (HOSE: ITA)
24. Hoa Sen Group (HOSE: HSG)
25. Ha Do Group Joint Stock Company (HOSE: HDG)
26. Coteccons Construction Joint Stock Company (HOSE: CTD)
27. Hoang Anh Gia Lai Joint Stock Company (HOSE: HAG)
28. Binh Minh Plastics Joint Stock Company (HOSE: BMP)
29. Hoa Binh Construction Group Joint Stock Company (HOSE: HBC)
30. Domesco Medical Import Export Joint Stock Corporation (HOSE: DMC)

A.2 Porfolio with 15 stocks

1. Bao Viet Holdings (HOSE: BVH)
2. Ho Chi Minh City Infrastructure Investment JSC (HOSE: CII)

3. Vietnam Joint Stock Commercial Bank for Industry and Trade (HOSE:CTG)
4. Petrovietnam Fertilizer and Chemicals Corporation (HOSE: DPM)
5. FPT Corporation (HOSE: FPT)
6. Gemadept Corporation (HOSE: GMD)
7. Hoa Phat Group Joint Stock Company (HOSE: HPG)
8. Kinh Bac City Development Share Holding Corporation (HOSE: KBC)
9. KIDO Group (HOSE: KDC)
10. Military Commercial Joint Stock Bank (HOSE: MBB)
11. Masan Group Corporation (HOSE: MSN)
12. Viet Nam Dairy Products Joint Stock Company (HOSE: VNM)
13. Bank for Foreign Trade of Vietnam (HOSE: VCB)
14. Sai Gon Securities Incorporation (HOSE: SSI)
15. Thanh Thanh Cong Tay Ninh Joint Stock Company (HOSE: SBT)

A.3 Porfolio with 8 stocks

1. Bao Viet Holdings (HOSE: BVH)
2. Ho Chi Minh City Infrastructure Investment JSC (HOSE: CII)
3. Vietnam Joint Stock Commercial Bank for Industry and Trade (HOSE:CTG)
4. Petrovietnam Fertilizer and Chemicals Corporation (HOSE: DPM)
5. FPT Corporation (HOSE: FPT)

6. Gemadept Corporation (HOSE: GMD)
7. Hoa Phat Group Joint Stock Company (HOSE: HPG)
8. Kinh Bac City Development Share Holding Corporation (HOSE: KBC)

Appendix B

MATLAB codes for PSO

Listing B.1: Save the following codes in MATLAB script file (*.m) and save as ofun.m.

```
---------------------------------------------------------------------------
Save the following codes in MATLAB script file (*.m) and save as ofun.m.
---------------------------------------------------------------------------
% The Objective Function (Minimization Problem)
function f=fitnessfunction(x)
% objective function (minimization)
f=x(1)+ x(2);
% defining penalty for each constraint
PEN=1000; % penalty on each constraint violation
lam=PEN;
Z=0;
% constraints (all constraints must be converted into ≤0 type)
% if there is no constraints then comments all c0 lines below
c0=[];
c0(1)=x(1)+ x(2);% ≤0 type constraints
%Apply all equality constraints as a penalty function
for k=1:length(c0),
    Z=Z + lam*c0(k)^2*TesteqH(c0(k));
end
f=f+Z;   % fitness function
% Test if equalities hold so as to get the value of the Index function
% getH1(c0) which is something like the Index in the interior-point methods
---------------------------------------------------------------------------
```

```
Save the following main program codes in MATLAB script file (*.m) as ...
    run_pso.m
(any name can be used) and run
---------------------------------------------------------------------------
tic
clc
clear all
close all
rng default
LB=[-1 -1 -1 -1 -1 -1 -1 -1 -1 -1  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
    -1 -1 -1 -1 -1 -1 -1]; % Lower Bounds of Variables
UB=[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]; % Upper ...
    Bounds of Variables
% Set Input Parameters for PSO Algorithm
n = 30; % The Number of Variables
p = 100; % The Size of Population
w = 0.7298;% Inertia Weight
c1 = 1.49618; % Personal Acceleration Coefficient
c2 = 1.49618; % Global Acceleration Coefficient
maxite = 1000; % Set Stopping Criteria
Number.Run = 1;% Set The Maximum Number of Runs
%-------- Start Main Program of Particle Swarm Optimization---------------%
for run=1:Number.Run
    run
    % The Initialization of PSO Algorithm-----------------------------Start
    Ini.Position = Initialization_PSO(p,LB,UB);
    Position = Ini.Position; % Initial Population
    Velocity = 0.1*Ini.Position; % Initial Velocity
    for i=1:p
        fvalue0(i,1) = fitnessfunction(Ini.Position(i,:)); % Find the ...
            fitness values of each particle
    end
    [fmin0,index0] = min(fvalue0); % Take the smallest value and its ...
       position from a set of fitness values
```

```
PB = Ini.Position; % Initial Personal Best Position
GB = Ini.Position(index0,:); % Initial Global Best Position
% The Initialization of PSO Algorithm-------------------------------End
% PSO Algorithm---------------------------------------------------Start
iteration=1;
while iteration ≤ maxite
% Velocity Updates
    for i=1:p
        for j=1:n
          Velocity(i,j)=w*Velocity(i,j) + ...
              c1*rand()*(PB(i,j)-Position(i,j))+...
              + c2*rand()*(GB(1,j)-Position(i,j));
        end
    end
    % Position Updates
    for i=1:p
        for j=1:n
            Position(i,j)=Position(i,j)+ Velocity(i,j);
        end
    end
    % Handling Boundary Violations
    Position = HandleViolation(Position,LB,UB);
    % Evaluating Fitness Value
    for i=1:p
        fvalue(i,1) = fitnessfunction(Position(i,:));
    end
    % Updating PB and Fitness value
    for i=1:p
        if  fvalue(i,1)<fvalue0(i,1)
            PB(i,:)=Position(i,:);
            fvalue0(i,1)= fvalue(i,1);
        end
    end
    [fmin,index]=min(fvalue0); % Finding The Best Particle
```

```
        S.bestfitness(iteration,run)=fmin; % Storing Best Fitness
        S.interationcount(run)=iteration; % Storing Iteration Count
       % Updating GB and The Best Fitness
        if fmin<fmin0
            GB=PB(index,:);
            fmin0=fmin;
        end
        % Displaying Results
        disp(sprintf('%8g %8g %8.4f',iteration,index,fmin0));
        iteration=iteration+1;
    end
    % PSO Algorithm-----------------------------------------------------End
    GB;
    ffvalue =fitnessfunction(GB);
    fff(run)=ffvalue;
    rgbest(run,:)=GB;
    disp(sprintf('---------------------------------------'));
end
%--------- End Main Program of Particle Swarm Optimization ---------------%
disp(sprintf('\n'));
disp(sprintf('*********************************************************'));
disp(sprintf('Results-----------------------------'));
disp(sprintf('The Best Fitness Value and The Run of The Best Fitness ...
   Value:'));
[bestfun,bestrun]=min(fff)
disp(sprintf('The Position of The Best Fitness Value:'));
best_variables=rgbest(bestrun,:)

toc
disp(sprintf('*********************************************************'));
% PSO Convergence Characteristic
b=S.bestfitness(1: S.interationcount(bestrun),bestrun);
plot(b, '-k','LineWidth',3);
grid minor
```

```
xlabel('Iteration');
ylabel('Best Fitness Function Value');
grid on;
---------------------------------------------------------------------------
% All subfunctions are listed here
---------------------------------------------------------------------------
Save the following codes in MATLAB script file (*.m) and save as TesteqH.m.
---------------------------------------------------------------------------
% Test if equalities hold
function H=TesteqH(g)
    if g==0,
        H=0;
    else
        H=1;
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
---------------------------------------------------------------------------
Save the following codes in MATLAB script file (*.m) and save as
Initialization_PSO.m.
---------------------------------------------------------------------------
% Intial locations of particles
function [initialposition]=Initialization_PSO(n,Lb,Ub)
ndimension=length(Lb);
for i=1:n,
    initialposition(i,1:ndimension)=Lb+rand.*(Ub-Lb);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
---------------------------------------------------------------------------
Save the following codes in MATLAB script file (*.m) and save as
HandleViolation.m.
---------------------------------------------------------------------------
% handling boundary violations
function x=HandleViolation(x,Lb,Ub)
n=length(x);
```

```
for i=1:n,
 % Apply the lower bound
 x_hv=x(i,:);
 A=x_hv<Lb;
 x_hv(A)=Lb(A);

 % Apply the upper bounds
 B=x_hv>Ub;
 x_hv(B)=Ub(B);
 % Update this new move
 x(i,:)=x_hv;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
-------------------------------------END----------------------------------
```