Portfolio management using computational intelligence approaches. Forecasting and Optimising the Stock Returns and Stock Volatilities with Fuzzy Logic, Neural Network and Evolutionary Algorithms.

Portfolio management using computational intelligence approaches

Forecasting and Optimising the Stock Returns and Stock Volatilities with Fuzzy Logic, Neural Network and Evolutionary Algorithms

Prisadarng SKOLPADUNGKET

Submitted for the Degree
of Doctor of Philosophy

Department of Computing

University of Bradford

2013

Abstract

Author: Prisadarng SKOLPADUNGKET

Title: Portfolio management using computational intelligence approaches

Keywords: Portfolio optimisation, Realistic constraints, Multi-objective genetic algorithm, estimation error, model risk, Fuzzy model selection, Strength Pareto Evolutionary Algorithm 2

Portfolio optimisation has a number of constraints resulting from some practical matters and regulations. The closed-form mathematical solution of portfolio optimisation problems usually cannot include these constraints. Exhaustive search to reach the exact solution can take prohibitive amount of computational time. Portfolio optimisation models are also usually impaired by the estimation error problem caused by lack of ability to predict the future accurately. A number of Multi-Objective Genetic Algorithms are proposed to solve the problem with two objectives subject to cardinality constraints, floor constraints and round-lot constraints. Fuzzy logic is incorporated into the Vector Evaluated Genetic Algorithm (VEGA) to but solutions tend to cluster around a few points. Strength Pareto Evolutionary Algorithm 2 (SPEA2) gives solutions which are evenly distributed portfolio along the effective front while MOGA is more time efficient. An Evolutionary Artificial Neural Network (EANN) is proposed. It automatically evolves the ANN's initial values and structures hidden nodes and layers. The EANN gives a better performance in stock return forecasts in comparison with those of Ordinary Least Square Estimation and of Back Propagation and Elman Recurrent ANNs. Adaptation algorithms for selecting a pair of forecasting models, which are based on fuzzy logic-like rules, are proposed to select best models given an economic scenario. Their predictive performances are better than those of the comparing forecasting models. MOGA and SPEA2 are modified to include a third objective to handle model risk and are evaluated and tested for their performances. The result shows that they perform better than those without the third objective.

Acknowledgement

I would like to express my deepest gratitude to my principal academic advisor Dr. Keshav Dahal for his continued supports, guidance, and kindness to me. This research project would not have been possible without his support and encouragement.

I would like to thank Dr. Napat Harnpornchai, who has always provided me with good ideas and advice for this research work. And also my old friend, Mr. Anon Nakornthab, who did the proof reading to this thesis.

My research is greatly grounded on a considerable number of previous research works. I also would like to give credit to such laborious works of those researchers who attempted to enhance portfolio optimisation body of knowledge.

Table of Content

Table of figures

Table of Tables

Chapter 1

Introduction

A portfolio of assets is a collection of investable assets which have returns expected at the end of a definite period called “investment period.” As a result, the expected return of a portfolio is the weighted average of return of all assets in that portfolio according to their proportions. The actual return of a portfolio, which is the realized return at the end of the investment period, may not be equal to the expected return due to the uncertainty inherent in the expectations of returns of assets. In the case that all assets in a portfolio are risk free, we can presume that the expected return and the actual return of a portfolio be equal. Although there are differences in many aspects among assets or securities in an investment portfolio, by investment purposes, other aspects except those related to their returns and risks are disregarded. Prior to the inception of Modern Portfolio Theory in 1952, investors and portfolio managers made their investment decisions based on the classical investment tools, i.e. the dividend cash flow model (coined by John Burr Williams in 1938) for stocks and the yield to maturity for fixed income instruments (Oberuc 2004).

Modern Portfolio Theory originated in a paper by Harry M. Markowitz in 1952 (Markowitz 1952). The theory states that a portfolio manager should not select assets due only to characteristics that are particular to the assets, i.e. their expected returns and volatilities, but he needs to consider how each asset co-moves with all other assets. Moreover, by taking into account these co-movements which usually are measured by correlations among assets, he can construct a portfolio that has less risk given the same expected return than a portfolio constructed by ignoring the interaction between assets (Elton 1997). Since 1952, Modern Portfolio Theory has become a well-developed paradigm and an academic field. Originally Markowitz suggested that the expected returns of any assets in question should be the historical means of their returns, and the expected volatilities should be their historical variances (or standard deviations as the square root of variances). This suggestion would be appropriate if assets in question were all normally distributed, and the investment horizon was long enough.

Unfortunately, in practical settings the aforementioned assumptions are not true. A portfolio manager usually finds that the assets in question have no normal distributions. The investment horizon is also quite a short time period since most portfolio managers are subjected to report their performance monthly, quarterly and annually. To achieve a better portfolio return at a more optimal risk, a portfolio manager needs to forecast both expected returns and volatilities at the end of investment period of all assets s/he wishes to include in the portfolios rather than just simply to take the historical means and variances as the expected returns and volatilities (Oberuc 2004). There are volumes of research on forecasting models and algorithms of stock returns and volatilities attempting to make the forecasts consistently accurate through all period of time regardless of instantaneous market conditions. However, most of them failed both aspects of accuracy and consistency (Triana 2000).

Moreover, besides the aforementioned input problems, Markowitz's original model, which can obtain a closed-form solution, suffers from imposing some unrealistic assumptions. To make portfolio models more realistic, we need to set their assumptions in accordance with stock exchange customs and regulations by which they can only be solved by search algorithms, exhaustive or approximate.

The Markowitz model with some modification later by Black (Black 1972) to allow short-selling (allowing negative weights of assets) has a closed-form solution. By removing some realistic assumptions, such as the non-negativity constraints (i.e. no short sell on any assets are allowed); the integer constraints (i.e., shares of assets cannot be divided into lower than their trading units), etc., the model has a general form with only the assets' expected returns, the variance, and covariance of the assets as parameters. On the other hand, if the non-negativity constraint is imposed, there exists no general form (closed-form) solution for the optimisation problem. Although the model with non-negativity constraint can be solved efficiently by specialised algorithms and other ad hoc methods, imposing other constraints (e.g. the integer constraint or maximum number of asset constraint) will cause large-scale problems, and the model becomes unsolvable by mixed integer non-linear programming or other exact solution algorithms, within a reasonable time (Busetti 2000).

In the process of investment decision, a portfolio manager usually faces an abundance of choices of investment assets. Also, he may need to make timely decisions in a rapidly changing financial market. This represents a tough optimisation problem, which continues to present a challenge for efficient optimisation solution techniques (Maringer 2005). A variety of different techniques have been employed to solve the portfolio optimisation problem. The main drawback of techniques for exact solution is that the number of combinations of states that must be searched increases exponentially with the size of the problem and becomes computationally prohibitive (Crama 2003). Furthermore, these techniques are poor in handling the nonlinear objective and constraint functions, and several assumptions are generally required to make the problem solvable using reasonable computational resources (Maringer 2005). Alternatively, some heuristic-based techniques use algorithms to find approximate solutions for instances of NP-hard problems in a reasonable time (Blum 2001). By using heuristics, the optimisation problems can be tackled in polynomial time with a trade-off for their optimality. In some circumstances of practicality, the speed to reach the acceptable approximate solutions is very critical. Feasible near-optimum solutions are acceptable, but untimely ones are not. The simple heuristic solution approaches are based on specialised techniques that work particularly well for a given problem, but are only of limited applicability to other problems (Blum 2003.) Furthermore simple heuristics, based on greedy search algorithms, tend to stop in inferior local optima.

In order to overcome the above limitations, researchers from the 1990's onwards have focused much attention on meta-heuristic solution techniques (Blum 2003 and Yilmaz 2011). Meta-heuristics are general intelligent searches that could find the ways out of local optima. Despite their intelligence and generality, performances of meta-heuristics depend on the problem settings. A number of research publications have accumulated over the period of more than 20 years on the applications of meta-heuristic approaches addressing portfolio management and optimisation issues.

Portfolio optimisation is an important business problem involving selections of assets or classes of assets to be included in an investment portfolio in the beginning of an investment period. The problem is always in the dynamic domain since information and factors that affect optimisation are changing over time. An efficient portfolio optimisation system, therefore, needs to be adaptive. An adaptive business intelligent system consists of three modules working cooperatively, namely: optimisation module, prediction (please note we are using terms ‘forecasting’ and ‘prediction’ interchangeably for the same meaning in this thesis) module, and adaptation module. The graphical illustration is shown in figure 1.1 (Modified from Michalewicz 2007.)

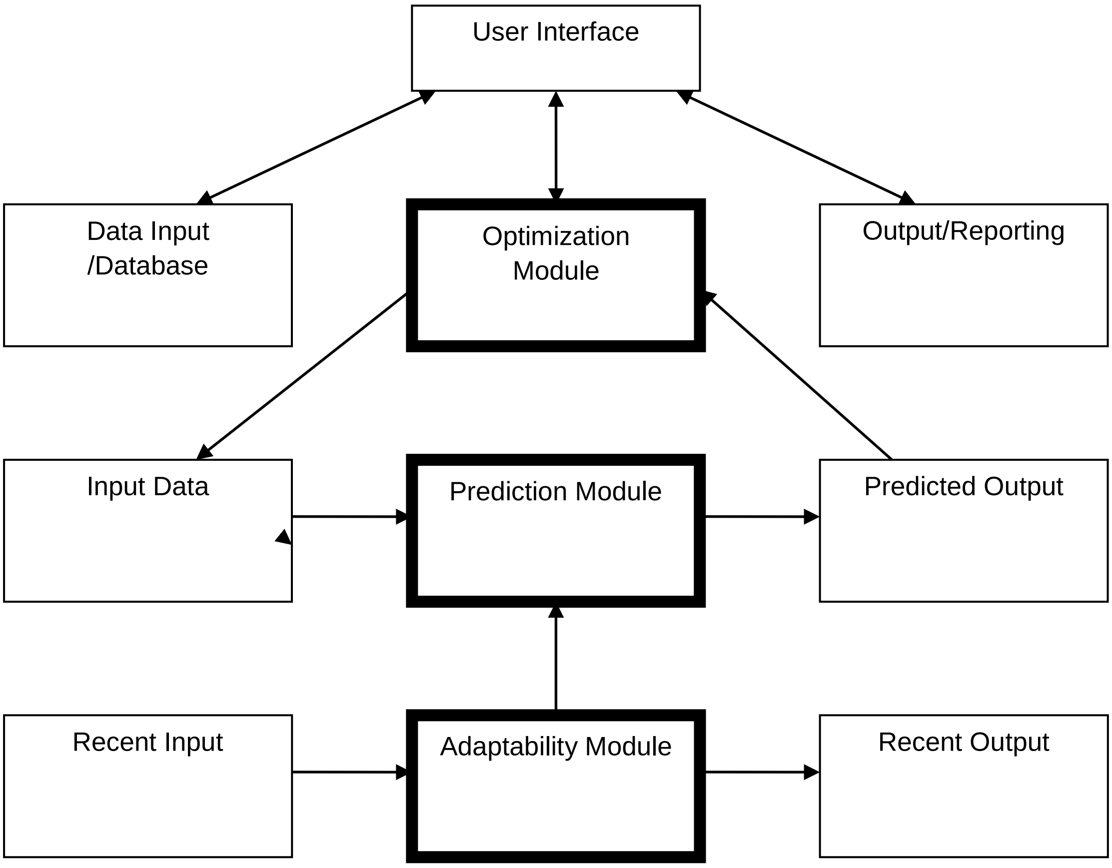


Figure 1.1: Structure of Adaptive Business Intelligence System

Most literature in portfolio optimisation with intelligent systems are largely concentrated in the optimisation module. To build a complete adaptive portfolio optimisation system, we need to do further research in the prediction module and adaptation module as well as improving performance and adaptability of the optimisation module. The research presented in this thesis aims to improve the optimisation module with consideration of prediction and adaptation issues. The research for improving the optimisation module will be directed toward dynamic environment. The techniques analysed for the optimisation module are Multi-objective Evolutionary Algorithms (MOEAs) or hybrid MOEAs. The prediction module is a hybrid system for prediction that consists of many prediction techniques or models. This research is considering time series econometric heuristic estimation, Evolutionary Programming (EP), Genetic Programming (GP) and Artificial Neural Networks techniques with a mechanism of choosing/averaging the results of the models. Figure 1.2 shows the framework diagram of hybrid systems for prediction. The adaptation module is responsible for regularly adjusting the prediction models. It detects and compares errors between the prediction result and the actual result. Then it will tune the prediction module to decrease prediction errors, if errors exist.

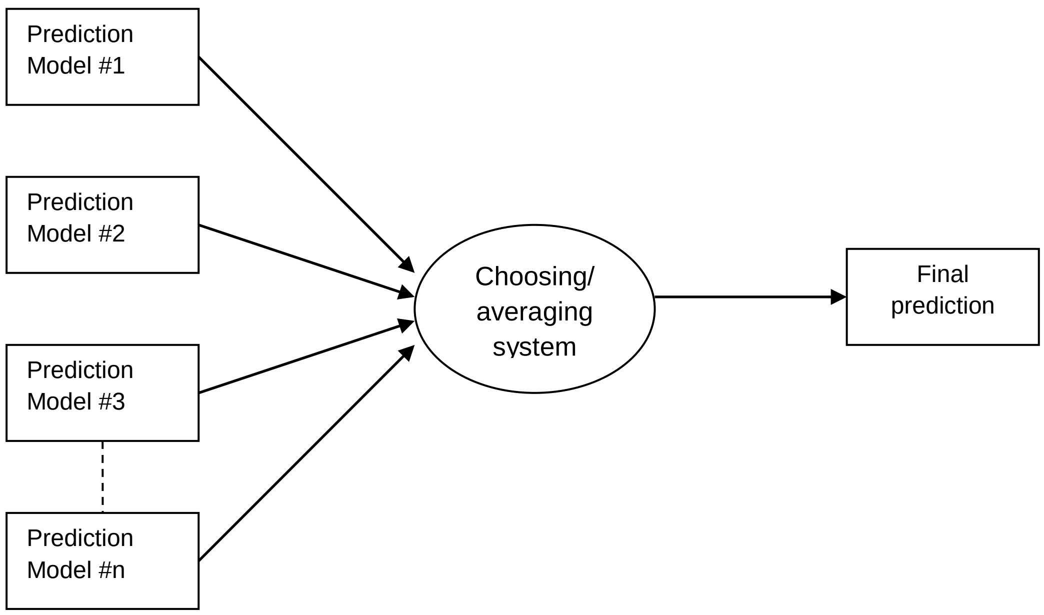


Figure 1.2: Hybrid Systems for Prediction

Models in the diagram above (figure 1.2) may include the same type of techniques or different type of techniques. The selection system may be a simple averaging, weighted averaging, rule based, neural networks, etc. The prediction module is quite important to the adaptive portfolio optimisation system because the inputs of portfolio optimisation models are expected returns (or yields) of all assets included in a portfolio, and variances (standard deviations) or volatilities of the returns, or risk measures (which are based on or can be calculated from variances, e.g. Absolute Variations, Value-at-Risk (VaR), Daily VaR). If the inputs cannot be predicted accurately or close to the actual values, the optimisation will also be inaccurate and leads to wrong decisions. The adaptation module is to update input data from time to time as well as develop and modify the parameters of the prediction models and/or the choosing/averaging system. The adaptability could be: update each individual model, update the choosing/averaging system, and update both of the individual models and choosing/averaging system.

1.1 Research Motivation and Scope

Generally speaking, a workable system of portfolio optimisation comprises at least 2 subsystems or modules, namely an optimisation module and a prediction module. The optimisation module must perform optimisation tasks as efficiently as possible both in terms of accuracy and timeliness, in accordance with the nature of the problem. However, the optimisation module needs inputs from the forecasting module. The forecasting module feeds inputs into the optimisation module. The final outcomes of the whole system therefore depend not only on the performance of the forecasting system, i.e. its accuracy, but also on the performance of the optimisation system, its closeness to the optimal solutions. However, there is abundant research on both optimisation techniques and forecasting models. The scope of the research presented in this thesis is to design an adaptive system to include both optimisation techniques and forecasting models for portfolio optimisation of stock portfolio. The performance of the system will be evaluated by ex-post basis in which a portfolio's actual returns and variances in successive periods are calculated and compared with those expected from previous period data.

As mentioned above, there is a lot research in both forecasting models and optimisation techniques. Doing research in those areas will only add drops of water into a large pool. On the other hand, there is comparatively little research on adaptation module or techniques. Thus, the research presented in this thesis will focus on adaptation techniques which can improve the forecasting part, the optimisation part as well as the system as a whole. However, we need to address and survey the optimisation and forecasting models in order to have a complete view of the whole system. We also put make some improvements to the forecasting and optimisation models. Since the environment is always changing, a single model or algorithm will not be appropriate for all market situations, therefore we emphasise our research on adaptation modules. We believe that effective adaptation modules will improve the outcomes of the system as a whole. In risk management perspective, an adaptive mechanism may be viewed as a mechanism to manage model risk. Model risk occurs when a model does not best represent the true nature of a situation and finding another model which is more appropriate for the situation is a way to reduce model risk.

Portfolio optimisation is designed to manage risk associated with price movements of assets (implied returns of investments). In the original portfolio optimisation presented by Markowitz (1952), the model assumed that we can take the past average of returns of an asset as an accurate prediction of its future returns as well as its past volatility as an accurate prediction of its future volatility. Unfortunately, the assumptions do not hold true in most of the market situations. Most of the time, portfolio managers need to supply the best prediction of both assets' returns and volatilities, in order to find an optimal choice of included assets into portfolios. So far, no model can make accurate predictions for all situations. The problem of portfolio optimisation becomes more complicated as it inherits model risk as well. Little research if any aims to handle this kind of risk in portfolio optimisation. Therefore, we aim to do so believing that if we can reduce model risk, we can improve the actual outcomes of portfolio optimisation significantly.

The scope of the research presented in this thesis, therefore, is to design algorithms aimed to improve the out of sample or ex post performance of the existing portfolio management methods, namely, the performance of the actual results. Also, portfolio optimisation problems need timely and appropriate answers and to properly execute portfolio arrangement and management, thus, the optimisation techniques need to handle applicable realistic constraints in a timely manner. We are also to design adaptation algorithms to handle model risk as a way to improve the actual performance of the whole system.

1.2 Objectives

The aim of this research is to develop a workable adaptive portfolio optimisation system. To achieve the above, the objectives of the research are set as follows:

1. To find and design optimisation modules which can effectively deal with realistic constraints in a timely manner. The optimisation algorithms need to

yield results close enough to the actual portfolio frontier and also have a decent distribution along the actual portfolio frontier. Moreover, the results must be given within an acceptable time

2. To find and improve stock return and volatility forecasting models which can make a good prediction for the next period of time given prevailing information prior to time of making prediction.
3. To design an adaptation algorithm to handle model risk associated with the forecasting models. The algorithm would use rules based on prevailing economic situations to select the best models given an economic situation.
4. To design and improve the optimisation models in such a way that they also can be adaptive and can handle model risk effectively.
5. To integrate the forecasting module and the optimisation module and test for their effectiveness and performances.

1.3 Research Contributions

The main contributions of this thesis can be summarised as follows:

- Multi-objective Genetic Algorithms (MOGAs) are explored and applied for portfolio optimisation with realistic constraints. Some novel mechanisms are added to the existing MOGAs to enhance their performances. (Chapter 4)
- Forecasting models and methodologies are explored and applied for predicting future stock returns and volatilities. A novel Genetic Algorithm (GA) is designed for variable selection into Artificial Neural Network (ANN) models for forecasting. Another novel algorithm is applied for evolving the structure of ANN models for forecasting. (Chapter 5)

- Fuzzy logic is applied for selecting of a pair of forecasting models for the next period stock return and volatility predictions. A novel algorithm is designed and its outcomes are evaluated. (Chapter 6)
- A number of MOGAs with good performance are modified to include another objective to handle model risk (as represented by expected forecasting errors). The prediction outcomes from chapter 6 are used as inputs. The algorithms are evaluated and tested for their performances. (Chapter 7)
- In summary, this thesis presents a number of novel approaches to portfolio optimisation. Firstly, some Multi-Objective Evolutionary Algorithms are modified and applied to solve portfolio optimisation problems with a number of constraints arising from practicality. Secondly, an Evolutionary ANN is proposed and used to forecast stock returns. Thirdly, a fuzzy-like rule system is proposed to select forecasting models which are appropriate for prevailing economic situations in particular periods of time. Lastly, MOGA and SPEA2 are modified to include a third objective as a way to manage risk associated with estimation errors which often arise from model risk. This is where we integrate the three modules together comprising the whole portfolio optimisation system. The final results are validated statistically.

1.4 Publications

The research works of the thesis have also been reported in the publications listed below.

1. Skolpadungket, P.; Dahal, K.; Harnpornchai, N., “A Survey on Portfolio Optimisation with Metaheuristics”, International Conference on Software, Knowledge, Information Management and Applications 2006 (SKIMA 2006), Chiang Mai, Thailand.
2. Skolpadungket, P.; Dahal, K.; Harnpornchai, N., “Portfolio optimisation using multi-objective genetic algorithms”, IEEE Congress on Evolutionary Computation 2007 (CEC 2007), Singapore.
3. Skolpadungket, P.; Dahal, K.; Harnpornchai, N., “Forecasting Stock Returns using Evolutionary Artificial Neural Networks”, 7th International Conference on Computational Intelligence in Economics and Finance (CIEF 2008), Taoyuan, China ROC (Taiwan).
4. Skolpadungket, P.; Dahal, K.; Harnpornchai, N., “Forecasting Stock Returns using Variable Selections with Genetic Algorithm and Artificial Neural – Networks”, 2009 Asia-Pacific Conference on Computational Intelligence and Industrial Applications (PACIIA 2009), Wuhan, China.

1.5 Thesis Outline

The thesis is structured as follows:

This chapter (Chapter 1) explains the background and motivation for the proposed research work and also lists the research contributions made.

Chapter 2 presents the review of the reported researches related to portfolio optimisation models. Those of classical models and post-classical models are reviewed.

Solving methods for portfolio optimisations are covered in Chapter 3. Together with Chapter 2, they comprise a review of relevant literature for this thesis.

Multi-objective Genetic Algorithms for portfolio optimisation are explored in Chapter 4. Some modifications are proposed to enhance the performances of the existing MOGAs. The newly modified MOGAs are also evaluated.

Chapter 5 surveys some selected forecasting methodologies which will be used in later chapters. A couple of novel techniques are proposed to enhance ANN forecasting models. The first algorithm is designed to let an ANN's structure evolve over time and reach its optimal form. The other algorithm is designed for automatic selection of most relevant variables for the forecasting ANN. The results are also examined and evaluated.

In Chapter 6, Fuzzy logic algorithms are designed to select a pair of forecasting models (one for stock return and one for stock volatility) from two groups of eight different kinds of models. The Fuzzy logic rules are based on their past performances and prevailing economic situations. The results are also examined and analysed.

Chapter 7 describes model risk and how to handle model risk within the framework of MOGA portfolio optimisation models. A modification to include a third objective into the two objective MOGAs in Chapter 4 is proposed. The forecasting outcome of the Fuzzy selection algorithms in Chapter 6 are used as inputs for the newly proposed MOGAs constituting a completed adaptive portfolio optimisation system aimed for this thesis. Finally, the results are evaluated and analysed.

Chapter 8 presents the conclusion of this thesis and the future works that can be extended from this work.

1.6 Terminologies and Abbreviations

Portfolio	A collection of financial assets, e.g., stocks, bonds, options, gold, with purpose obtain return for investment over period of time.
Return	Stock return or asset return is the yield for investment of a stock or asset over time. It may be in the form of price appreciation or dividend paid.
Volatility	Stock volatility or asset volatility is a variation of return of a stock or an asset occurred over a period of time. It is usually measured by either standard deviation or variance of its return for the period of time.
AC	Ant Colony
AR	Auto-Regressive
ARMA	Auto-Regressive Moving Average
ANN	Artificial Neural Network
APT	Arbitrage Pricing Theory
AVAR	Average Variable Average Rule
BPN	Back Propagation Network

CAPM	Capital Asset Pricing Model
CEC	Conference on Evolutionary Computation
CFS	Combined Fuzzy Score
CPU	Central Processing Unit
CV	Cross Validation
CVaR	Conditional Value-at-Risk
FSGA	Feature Selected Genetic Algorithm
EA	Evolutionary Algorithm
EANN	Evolutionary Artificial Neural Network
ES	Evolutionary Strategy
GA	Genetic Algorithms
GD	Generational Distance
GNN	Genetic Neural Network
KGA	Knapsack Genetic Algorithm
IAPM	International Asset Pricing Model
LS	Least Square Model
LSM	Local Search Metaheuristic
MA	Moving Average
MVBR	Multiple Variable Best Rule

MCV	Multi-fold Cross Validation
MOEA	Multi-Objective Evolutionary Algorithm
MOGA	Multi-objective Genetic Algorithm
NC1R	1-Norm-Constrained minimum-variance portfolio with δ calibrated by maximising portfolio Return in previous period
NC1V	1-Norm-Constrained minimum-variance portfolio with δ calibrated using cross-Validation over portfolio variances,
NC2R	2-Norm-Constrained minimum-variance portfolio with δ calibrated by maximising portfolio Return in previous period
NC2V	2-Norm-Constrained minimum-variance portfolio with δ calibrated using cross-Validation over portfolio variances
NCFR	$\mathbf{\Sigma}_F$-Norm-Constrained minimum-variance portfolio with δ calibrated by maximising portfolio Return in previous period
NCFV	$\mathbf{\Sigma}_F$-Norm-Constrained minimum-variance portfolio with δ calibrated using cross-Validation over portfolio variances
NPGA	Niche Pareto Genetic Algorithm
NSGA	Non-dominated Sorting Genetic Algorithm
OLS	Ordinary Least Square
PAES	Pareto Archive Evolutionary Strategy

PARR	Partial minimum-variance portfolio with k calibrated by maximising portfolio return in previous period
PARV	Partial minimum-variance portfolio with k calibrated using cross-validation over portfolio variances
SA	Simulated Annealing
SBVBR	Single Best Variable Best Rule
SR	Sharpe Ratio
SPEA	Strength Pareto Evolutionary Algorithm
TOGA	Target Objectives Genetic Algorithm
TS	Tabu Search
US	United States
VaR	Value-at-Risk
VEGA	Vector Evaluated Genetic Algorithm

Chapter 2

Review of Portfolio Optimisation Models

2.1 Chapter Overview

Markowitz's seminal paper in 1952 was the first work on Modern Finance in general and Modern Portfolio Theory in particular. Markowitz himself had long been regarded as the father of Modern Finance even before he was honored with Nobel Prize in Economics in 1989. Moreover, the model was the first that defined the concepts of risk and risk management related to financial and investment management. The concept of risk in the model is defined as the volatility of asset prices which is usually represented by the variance or standard deviation (square root of variance) of asset's prices under consideration (even though there are other definitions of risk such as Value-at-Risk or VaR). The risk management as far as the model is concerned is not to eliminate the risk associated with portfolio management but to choose the combination of assets that yields the lowest overall portfolio's variance given an expected portfolio return. The optimum portfolio management by minimising portfolio's variance is equivalent to optimum portfolio by maximising portfolio's return given a portfolio's variance.

This chapter reviews the portfolio optimisation problem models reported in the literature. The challenges and recent developments in the area of realistic portfolio optimisation are reviewed. The structure of this chapter is as follows: Section 2.1 is for the background of the so-called classical Portfolio Optimisation models and their limitations; Section 2.2 discusses on some recent Portfolio Optimisation models and techniques aimed at overcoming the classical Portfolio Optimisation models; Section 2.3 discusses on some Portfolio Optimisation models with alternative objectives. Section 2.4 covers the problems of estimation errors and model risk and their remedies. In Section, 2.5, we make some conclusions of the recent stage of research and current research gaps and challenges.

2.2 The Classical Portfolio Optimisation Models

One of the practical problems in asset management is how to allocate money to invest in different assets in order to achieve the investors risk appetites and return objectives (Markowitz 1992). An investor is assumed to be a rational economic agent who is risk averse. Given an objective at a level of return, an investor tries to reduce risk as much as possible. To construct a portfolio of assets, a portfolio manager, who acts in the best interest of the investors, adds a number of assets to form a new asset portfolio that has a different risk-return characteristic than those of individual assets. Choices and quantities of different assets that should be included into the portfolio are the outcomes of portfolio selection process. A set of portfolio of feasible set of assets that has a minimum risk level and a maximum return level is called an optimal or efficient portfolio. However, we cannot simply choose an asset which exhibits minimal variance and maximal return because there are correlations between pairs of assets which can further reduce the overall portfolio's variances. We have to consider all of the assets' risks and returns simultaneously. Based upon the aforementioned ideas, portfolio selections should follow Modern Portfolio Theory (MPT) originated in a paper by Harry M. Markowitz in

1952 (Markowitz 1952 cited in Markowitz 1992 and in Elton 1997). The theory states that an investor should not select assets due only to characteristics that are particular to the assets but she/he needs to consider how each asset co-moves with all other assets. Moreover, by taking into account of these co-movements, an investor can construct a portfolio that has less risk given the same expected return than a portfolio constructed by ignoring the interaction between securities (Elton 1997). However, the original MPT is based on a number of unrealistic assumptions, namely, no short selling, infinitesimal dividing of investment assets, observed means and variance of assets represent true means and variance of assets overtime. The Markowitz' version of MPT cannot have a closed form solution but can be easily solved by quadratic programming method.

2.1.1 The Markowitz model

The Markowitz model assumes that investors make their decision in portfolio construction by choosing assets that maximise their portfolio returns at the end of investment period (expected returns). By assuming that investors are risk averse, the simplest model with a number of unrealistic assumptions, namely perfect market without taxes, no transaction costs, no short sales, assets are infinitely divisible, the Markowitz portfolio optimisation can be stated mathematically as follows:

$$Min_{w_i} \sigma_p^2 \tag{2.1}$$

$Subject\ to$

$$\sigma_p^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} w_i\, w_j \sigma_{ij} \tag{2.2}$$

$$r_p = r^* \tag{2.3}$$

$$r_p = \sum_{i=1}^{N} w_i r_i \tag{2.4}$$

$$\sum_{i=1}^{N} w_i = 1 \qquad (2.5)$$

$$0 \leq w_i \leq 1 \quad \forall i \qquad (2.6)$$

Where, σ_{ij} is covariance between asset i and j, if i = j, it is variance of asset i.

σ^2_P is variance of the portfolio of assets,

r_i is expected return of asset i,

r_P is the expected return of the portfolio,

r^* is a predefined level of return,

w_i is weight or proportion of asset i in the portfolio p.

The no-short sell constraint in Equation (2.6) makes the model having no closed form solution and also NP-hard. In the computational aspect, allowing short-selling would make the solution of the problems (i.e. proportion of each asset) a set of all real numbers rather than merely zero and positive real numbers, therefore a closed form solution could be obtained. In this sense, we can think of the no short selling condition as a constraint. However, for a small number of assets to be included in the portfolio, the model can be solved numerically by quadratic optimisation method with a reasonable time (Maringer 2005).

However, later Markowitz has modified the original model to relax some of the aforementioned assumptions such as taxation, transaction costs, etc. The readers are referred to (Markowitz 1992) for detailed working on the modified model and solving methods.

2.1.2. Black's Modification

Black (Black 1972) modified the Markowitz model to allow short-selling (allowing negative weights of assets). The model has a closed form solution. By removing some realistic assumptions such as the non-negativity constraints (i.e. no short sell on any assets are allowed), weights of any assets in the portfolio can be any real number given the sum of them satisfies Equation (2.5), Equation (2.6) is now modified as

$$-\infty \leq w_i \leq +\infty \quad \forall\, i \tag{2.7}$$

The model, thus, has a closed form or exact solution that only the assets' expected returns, the variance and covariance of the assets are parameters which can be solved by Lagrange Methods for constraint optimisation (Amenc 2003). The closed form solution for asset weights for a given level of return, $r_p = r^*$ thus can be determined by

$$\sigma_p^2 = \begin{pmatrix} r_p & 1 \end{pmatrix} A^{-1} \binom{r_p}{1} = \frac{a - 2rp + c(r_p)^2}{ac - b^2} \tag{2.8}$$

with

$$\boldsymbol{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \binom{r'}{I'} \boldsymbol{\Sigma^{-1}} (\boldsymbol{r} \quad \boldsymbol{I}) \tag{2.9}$$

and

$$\boldsymbol{w} = \boldsymbol{\Sigma^{-1}} (\boldsymbol{r} \quad \boldsymbol{I}) \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{\boldsymbol{-1}} \binom{r_p}{1} \tag{2.10}$$

where, **r** and **r**' are the vector of r_i and its transpose respectively,

$\mathbf{\Sigma}$ is the variance-covariance matrix of all (NxN) assets,

$\boldsymbol{I}$ is the unity vector.

The Black model comes with some convenience properties including having a closed form solution. In fact, it can be represented by a linear combination of any two efficient portfolios along the portfolio efficient frontier. Therefore, knowing just only a couple of efficient portfolios enables us to replicate any efficient portfolio. However, the solutions of efficient portfolios may be impracticable due to their short sale positions (i.e. negative weights of assets). Some assets can be short sold but in a very limited way both in name and quantity. Compromising the solutions by ignoring any negative weights, setting them to zero, and adjusting the remaining asset weights accordingly so they sum up to one might result in the solution becoming inefficient (Maringer 2005).

2.1.3 Tobin Model

By assuming that there is a riskless asset, the efficient line is no longer a parabola curve but a straight line or a linear relationship between return and volatility. This model is called the Tobin model as it was proposed by Tobin (1965). If all funds are invested in a riskless asset s with a risk free return (r_S) and some risky portfolio T with a return (r_T) then the fund portfolio P will have portfolio return (r_P) as

$$r_P = \alpha r_s + (1-\alpha) r_T \tag{2.11}$$

Since s is risk-free then its volatility (variance or standard deviation) is also zero. The portfolio volatility (σ_P) is

$$\sigma_P = (1-\alpha)\sigma_T \tag{2.12}$$

Where σ_T is the volatility of the risky portfolio T.

From the equation (2.12), we can solve for α such that

$$\alpha = 1 - \frac{\sigma_P}{\sigma_T} \tag{2.13}$$

Substitute (2.13) into (2.11), we can have a linear relationship between the portfolio's return and volatility:

$$r_P = r_s + (r_T - r_s)\frac{\sigma_P}{\sigma_T} \tag{2.14}$$

The Tobin model has a far reaching implication. If we can find a riskless asset and an efficient portfolio according Markowitz's or Black's efficient frontier, we will be able to construct a portfolio of any return levels which is more than or equal to the original frontier. The new frontier is a linear combination between the efficient portfolio and the riskless asset which is named "Capital Market Line". Now, an investor needs only a riskless asset and an efficient portfolio to invest and satisfy his risk-return appetite. In other words, the optimal portfolio T can be separated from the investment decision. This is called "portfolio separation theorem". However, according to this model, although an investor can choose any risk-return level of investment by combining a riskless asset and an efficient portfolio, the efficient portfolio must be estimated since it is not given.

2.1.4. The Capital Asset Pricing Model (CAPM)

Portfolio Separation theory directly influences Sharpe (1964), Lintner (1965), and Mossin (1966) who independently developed the Capital Asset Pricing Model (CAPM). By asserting that the existing efficient portfolio which is represented by the market portfolio (i.e. portfolio in which consists of all assets in the market) and a risk-free asset, and now the new efficient frontier is a linear combination, CAPM states that the prices of any asset in the market will be adjusted accordingly so that their returns and volatilities align along the linear combination line which is so called “Security Market Line.” Now, according to this model, any asset’s returns can be estimated by its relative correlation with the market portfolio which is given by a symbol β (beta). The beta is defined by

$$\beta_i = \frac{\sigma_{iM}}{\sigma_M^2} = \frac{\sigma_i \rho_{iM} \sigma_M}{\sigma_M \sigma_M} = \frac{\sigma_i \rho_{iM}}{\sigma_M} \tag{2.15}$$

Where, σ_M and σ_M^2 are standard deviation and variance of the market portfolio,

σ_{iM} is the covariance between the market portfolio and asset i,

ρ_{iM} is the correlation coefficient between the market portfolio and asset i,

σ_i is the standard deviation of asset i.

By knowing the beta, we can estimate the return of any asset i by the following equation:

$$r_i = r_s + (r_M - r_s)\beta_i \tag{2.16}$$

Where, r_i is the theoretical return of asset i,

r_s is the observed return of the risk-free asset,

r_M is the (observed) return of the market portfolio.

There are many researchers who have tried to test empirically the validity of CAPM. The results have been mixed. A number of earlier empirical tests, notably by Black (1972), Fama (1973), and Blume (1975) confirmed the theory. On the contrary, some later empirical tests, e.g. Cheng (1980) and Gibbsons (1982), rejected it.

Aforementioned important developments have high impacts on practical implications. The portfolio separation theorem leads to mutual fund theorem. The theory states that if an investor has access to a risk free asset or a comparatively riskless asset, the investor's optimal portfolio is independent of his preferences for expected return and volatility. The investor then can construct an optimal portfolio at any expected return level from only a few elements, namely a risk free asset and one or few market portfolios or mutual funds. This radically simplified investment decision making. The important outcomes of this strand are the Capital Asset Pricing Model (CAPM) (in the seminal papers, Sharp 1964, Lintner 1965, and Mossin 1966 as cited in Elton 1997) and the Arbitrage Pricing Theory (APT) (in the seminal papers, Ross 1967 and Roll 1980 as cited in Elton 1997). However, empirical researches are still inconclusive whether the models actually work in reality. Above all, due to practical and cost matters, portfolio managers sometime have still to construct market portfolios or mutual funds from a subset of all available assets. Another strand has been to extend the Markowitz model into multi-period and dynamic models. A comprehensive treatment of this strand of development can be found in (Merton 1992). More recently, many researchers have extended the model into stochastic forms (e.g. Gulpinar 2003 and Fernholz 1982, 2002, 2003). However, besides Markowitz mean-variance model, CAPM and APT, other models still have lesser practical applications in the investment management industry (Anenc 2003).

2.1.5. Problems and Limitations of the Classical Models

Primarily, there are three problematic aspects of the classical portfolio optimisation models, namely, the problems of the model's objectives, the problems of the model's constraints, and the problem about estimation errors of the inputs of the models. The first aspect is somewhat theoretical, like those of Economic theories in which we observe whether an economic agent has rational behaviors by always maximising profits and utilities in certain ways. In portfolio optimisation models, this aspect concerning what should be representations of risk and what an investor tries to maximise and to minimise in the context of the models. There are a lot of criticisms concerning using mean-variance as the objectives of portfolio optimisation models recently (Gilli 2001). The second aspect has been addressed by a large number of researches. In order to solve for solutions of the models with mathematical or numerical ease, Markowitz (1952) and later Black (1972) simplify the reality of the market practices by excluding a number of realistic constraints which would have made the optimisation problem become discrete so that the optimisation problems are based on differentiable functions and constraints. By introducing realistic discrete constraints into portfolio optimisation, the problems become combinatory and NP complete (Jobst 2001). The third aspect is the most serious problem: Jorion (1992) claims that the estimation errors of the expected returns and the expected volatilities as "inputs" of the portfolio optimisation models make them to be under-utilised by most investment practitioners. The estimation errors of the input may result from inaccurate expert judgments, in case of prediction by experts, or model risk, in case of prediction by forecasting models. The classical portfolio optimisaton models take the accuracies of asset returns and volatilities (historical means as expected return and historical variances as expected volatilities in the Markowitz model) for granted. Optimal portfolios as outcomes are sensitive to the inputs. Wrongly estimating expected asset returns and volatilities will cause the outcomes to be suboptimal. The optimal portfolios, which use expected returns and volatilities forecasted by models based on historical samples, usually perform poorly out of sample (DeMiguel 2009).

2.2 Post-Classical Portfolio Optimisation Models with Realistic Constraints

Recent developments of portfolio optimisation modeling have primarily tackled the problems and limitations of the classical models described in the previous sections. Introducing realistic constraints within the Markowitz model, which includes only the non-negativity constraint, makes portfolio optimisation models become discrete choice and NP complete problems.

The Markowitz model is a simplified model to focus only on a theoretical point of view. In the practicality of investment management, portfolio managers face a number of realistic constraints arising from normal business practices, practical matters and industry regulations. The realistic constraints that are of practical importance include (not exhaustively) integer constraints, cardinality constraints, floor and ceiling constraints, turnover constraints, trading constraints, buy-in threshold, and transaction cost inclusions.

Integer constraints require that the number of any asset included in the portfolio must be an integer or indivisible (i.e. cannot be in any fraction of normal trading lot). The integer constraints can be expressed as

$$w_i = \frac{n_i}{\sum_{i=1}^{N} n_i} \tag{2.22}$$

And

$$n_i \; mod \; l_i = 0 \qquad \forall i \tag{2.23}$$

Where n_i is number of unit of asset (share),

l_i is trading lot of the asset i,

N is the total number of assets and

mod is the modulo operator.

Cardinality constraints are the maximum number and minimum number of assets that a portfolio manager wishes to include in the portfolio due to monitoring reasons or diversification reasons or transaction cost control reasons (Stein 2005). The constraints can be expressed as follow

$$C_l \leq \sum_{i=1}^{N} b_i \leq C_u \tag{2.24}$$

Where $b_i = 1$ if $w_i > 0$, otherwise $b_i = 0$ and

C_l and C_u are the lowest and the highest number of assets to include in a portfolio respectively.

Floor and ceiling constraints define lower and upper limits on the proportion of each asset, which can be held in a portfolio. These constraints may result from institutional policy in order to diversify portfolio and to rule out negligible holding of assets for ease of control (Crama 2003). They can be expressed mathematically as follow

$$f_i \leq w_i \leq c_i \quad \forall i. \qquad (2.25)$$

Where f_i and c_i are the lowest and the highest proportion that asset i can be held in the portfolio respectively.

Turnover constraints impose upper bound for variations of the asset holding from one period to the next. The constraints are a mean to curb the transaction costs therefore they can be modeled indirectly by incorporating transaction costs and read as follows (Maringer 2005)

$$\max(w_i - w_i^0, 0) \leq B_{iu} \quad \forall i \qquad (2.26)$$

$$\max(w_i^0 - w_i, 0) \leq S_{iu} \quad \forall i. \qquad (2.27)$$

Where w^0_i is the holding proportion of asset i in the initial portfolio,

B_{iu} is the maximum purchase of asset i during the current holding period, and

S_{iu} is the maxiimum sales of asset *i* during the current holding period.

On the other hand, trading constraints impose limits on buying and selling tiny quantities of assets due to practical reasons and can be stated as follows (Maringer 2005)

$$(w_i = w_i^0) \lor (w_i \geq (w_i^0 + B_{il})) \lor (w_i \leq (w_i^0 + S_{il})) \quad \forall i. \qquad (2.28)$$

Where B_{il} denotes the minimum purchase of asset i during the current holding period and S_{il} denotes the minimum sale of asset *i* during the current holding period.

The business of asset trading, stock brokers, bond dealers, etc. is related to buying and selling with money. Therefore, transaction costs associated with purchases and sales of assets are inevitable and should be incorporated in the realistic models (Maringer 2005). Transaction costs have many forms as follows:

Fixed fee per transaction:

$$T_i = T_f \qquad (2.29)$$

Variable fee per (dollar) amount:

$$T_i = t_p n_i S_i^0 \qquad (2.30)$$

Variable fee per (dollar) amount with minimum charge:

$$T_i = \max\{t_f, t_p n_i S_i^0\} \qquad (2.31)$$

Variable fee per (dollar) amount plus fixed charge:

$$T_i = T_f + t_p n_i S_i^0 \qquad (2.32)$$

Where T_i is the transaction cost of inclusion of asset *i* into the portfolio,

T_f is the fixed fee per transaction of purchase or sale asset *i,*

t_f is the minimum (floor) fee per-transaction,

t_p is variable fee per amount (dollars) of purchase or sale of asset i,

n_i is the number of purchased or sold share of asset i, and

S_i^0 is current trading or market price of asset *i* (at the time of purchase or sale).

The transaction costs affect the fund that can be invested in all assets. Let V_0 be the initial endowment the portfolio manager is entrusted to construct a portfolio. And n_i is the amount of asset i (assuming the integer constraints hold) then the amount of fund that can be invested in the portfolio will not equal V_0 but will equal (assuming the fund is invested completely on assets).

$$V_0 - \sum_{i=1}^{N} T_i^0 = \sum_{i=1}^{N} n_i S_i^0 \tag{2.33}$$

Where N is the number of asset held in the portfolio.

And if there is no trading during the holding period then the expected portfolio return (r_p) on the initial endowment for the holding period will be (Maringer 2005).

$$r_p = \frac{\sum_{i=1}^{N}(n_i S_i^0 (1+r_i)) - \sum_{i=1}^{N}(T_i^0 + T_i^1)}{V_0} - 1 \tag{2.34}$$

Where T_i^0 is the transaction (purchase) cost of asset i at the beginning of the holding period and

T_i^1 are transaction (sale) cost at the end of the holding period (to convert all assets to cash).

2.3. Models with Alternative Objectives

From the invention of simple single period of Markowitz's portfolio optimisation, there have been many strands of developments. Earlier, investor's utility functions had been taken into consideration as well as additional moments rather than variance such as skewness (Tobin 1958, Lee 1977, Kraus 1976, Fama 1965, and Elton 1974 cited in Elton 1997). The original mean-variance model is based on the assumptions that investors are risk averse and their utility is a quadratic function of the rate of return. Moreover, the distribution of the rate of return must be multivariate normal. But these assumptions do not believe that they hold in realistic setting, especially since many researches indicated that the distribution does not follow a multivariate normal distribution (Yilmas 2010). Since the distribution is not normal, it exhibits non-zero third and fourth (possibly beyond fourth) moments which are skewness and kurtosis respectively.

Jurczenko et al (Jurczenko 2006) constructs a portfolio optimisation model with the first four moments, namely mean, variance, skewness and kurtosis. The model is imposed a no short-sale constraint (non-negativity constraint). The set of feasible portfolio represented by the fraction of proportion vector $\mathbf{w_p}$ must conform to the following restrictions

$$\mathbf{w_p} \in \mathbf{IR^N} \tag{2.35}$$

$$\mathbf{w_p'1} \in \mathbf{1} \tag{2.36}$$

$$\mathbf{w_p} \geq \mathbf{0} \tag{2.37}$$

Where, $\mathbf{w_p}$ is the **(1xN)** vector of the investor's holding of assets;

$\mathbf{w_p}'$ is the transpose of $\mathbf{w_p}$;

1 is the (**Nx1**) unitary vector.

The mean (E (R_p)), variance ($\sigma^2(R_p)$), skewness ($s^3(R_p)$) and kurtosis ($k^4(R_p)$) of portfolio return of a feasible portfolio p are given respectively by:

$$E\left(R_p\right) = E(\textstyle\sum_{i=1}^{N} w_{pi} R_i) = \mathbf{w_p'}\,\mathbf{E} \qquad (2.38)$$

$$\sigma^2\left(R_p\right) = E\{(R_p - E(R_p))^2\} = \textstyle\sum_{i=1}^{N}\sum_{j=1}^{N} w_{pi}\, w_{pj}\sigma_{ij} = \mathbf{w}_p'\boldsymbol{\Omega}\mathbf{w_p} \qquad (2.39)$$

$$s^3\left(R_p\right) = E\{(R_p - E(R_p))^3\} = \textstyle\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} w_{pi}\, w_{pj} w_{pk} s_{ijk} = \mathbf{w}_p'\boldsymbol{\Sigma}\left(\mathbf{w_p}\otimes\mathbf{w_p}\right) \qquad (2.40)$$

$$k^4\left(R_p\right) = E\left\{\left[R_p - E(R_p)\right]^4\right\} = \textstyle\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N} w_{pi}\, w_{pj} w_{pk} w_{pl} k_{ijkl} = \mathbf{w}_p'\boldsymbol{\Gamma}\left(\mathbf{w_p}\otimes\mathbf{w_p}\otimes\mathbf{w_p}\right) \qquad (2.41)$$

Where, w_{pi}, w_{pj}, w_{pk} and w_{pl} are weights of assets i^{th}, j^{th}, k^{th} and l^{th} respectively;

Ω is the non-singular (N x N) variance-covariance matrix of the assets;

$\boldsymbol{\Sigma}$ is the (N x N^2) skewness-coskewness matrix;

$\boldsymbol{\Gamma}$ is the (N x N^3) kurtosis-cokurtosis matrix and

The sign $'\otimes'$stands for the Kronecker product.

The portfolio optimisation in this case can use the shortage function to find the efficient frontier (Briec 2007). However, the finding of such efficient frontier by this method can only guarantee weak efficiency for the mean-variance-skewness-kurtosis space (Jurczenko 2006). The shortage function for the true unknown four moments (i.e. mean, variance, skewness and kurtosis) can be computed by solving the following quadratic optimisation programming (ibid).

$$\mathbf{w_p^*} = \mathrm{Arg}_{\mathrm{w_p}}\{\mathrm{Max}\,\delta\} \tag{2.42}$$

Subject to

$$E(R_k) + \delta g_E \leqq E(R_p)$$

$$\sigma^2(R_k) - \delta g_\sigma \geqq \sigma^2(R_p)$$

$$s^3(R_k) + \delta g_s \leqq s^3(R_p)\; k^4(R_k) + \delta g_k \geqq k^4(R_p)$$

$$\mathbf{w_p'1} = 1$$

$$\mathbf{w_p} \geq \mathbf{0}$$

Where $\mathbf{w_p}^*$ is the(N x **1**) efficient portfolio weight vector that maximises the return's mean, variance, skewness and kurtosis relative improvement over the evaluated portfolio in the direction of the vector **g;**

$\{g_E, g_\sigma, g_s, g_k\}$ are the elements of (1x4) direction vector covering the four-dimensional space of (the return's) mean, variance, skewness and kurtosis that point to instantaneous optimal direction of any optimizing portfolio k (used in quadratic optimisation programming).

However, the shortage function is still needed to include vector projections on the vertical and horizontal part of the non-convex portfolio. Also, more research is still needed in a more robust non-parametric multi-moment efficient portfolio frontier. There are also not many empirical researches concerning validity of this model.

Davies et al (Davies 2009) used the four moment model to study hedge fund allocation problem. In this study, they were to select a number of hedge funds into a portfolio (i.e. the portfolio is a fund of funds). Hedge funds were classified into groups of strategies namely, Long/Short Equity, Equity Market Neutral, Convertible Arbitrage, Distressed Securities, Merger Arbitrage, Global Macro and Emerging Markets. The data used in their study was obtained from Tremont TASS which is one of the largest hedge fund databases. They applied an algorithm called Polynomial

Goal Programming (PGP) to solve the multi-objective optimisation problem. They found that, firstly, hedge fund return statistics (1st moment) tended to trade off against each other funds. Secondly, inclusion skewness and kurtosis in portfolio optimisation problem resulted in huge different from the only mean and variance portfolio seemingly from various trading-off effects. Thirdly, there were two groups of hedge funds, namely equity market neutral funds and global macro funds that had crucial roles in optimal hedge fund portfolios. Due to their attractive co-variance, co-skewness and co-kurtosis properties, equity market neutral funds played roles of volatility and kurtosis reducers, while global macro fund played a role of skewness enhancers. Lastly, hedge funds and stocks should not be mixed together in the same portfolio but each of them should be combined with bonds to yield a more four-moment efficient portfolio, especially for skewness.

Hochreiter (2007) proposed an evolutionary computation approach for portfolio optimisation with general risk measures. He used a standard single objective GA to solve portfolio optimisation with expected return instead of mean and three risk measures (Standard Deviation, Value at Risk (VaR) and Conditional Value at Risk (CVaR)). The author reformulated the two objectives into single criteria for the GA as follow

$$Max_w \qquad E(l_w) - \kappa\rho(l_w) \tag{2.43}$$

Where, l_w is the distribution of discrete profit and loss of portfolio w,

$E(l_w)$ is expected return/profit of portfolio w,

$\rho(l_w)$ is the risk dimension of portfolio w,

κ is an additional risk aversion parameter.

Portfolio's Standard Deviation is defined in the context of (2.79) as follow

$$\rho = \sigma = \sqrt{\sum_{i \in S} p_i (l_i - E(l))^2} \tag{2.44}$$

Where, l_i is portfolio return that would occur in the scenario i.

S is the scenario set,

p_i is the probability in which scenario i would occur.

Portfolio's Value at Risk is defined as follow

$$\rho = VaR_\alpha = \inf\{k \in \mathbb{R} : P(k > l) \leq 1 - \alpha\} \tag{2.45}$$

Where, l is the portfolios return,

1-α is a predefined level of VaR,

P is the cumulative probability.

Portfolio's Conditional Value at Risk is defined as follows

$$\rho = CVaR_\alpha = E(l \mid l \leq VaR_\alpha) \tag{2.46}$$

The author experimented portfolio optimisation problems with the three risk measures in (2.80), (2.81) and (2.82) by the standard GA with population size of 500. The parameter α was set at 0.9. The result showed that portfolio combinations were quite different from each other regarding the risk measure used. However, no ex-post comparison had been made.

Gaivoronski and Pflug (Gaivoronski 2004) proposed a method of calculating mean-VaR portfolio optimal frontier. They formulated mean-VaR portfolio optimisation problem based on sampling data (historical or simulated) as follows

$$Min_{w \in W} VaR(w) \tag{2.47}$$

$Subject\ to$

$$VaR(w) = Min^{\lfloor \alpha N \rfloor + 1} \{w^T \xi^1, \ldots, w^T \xi^N\} \tag{2.48}$$

$$W = \{w \geq 0;\ w^T \boldsymbol{e} \geq \mu;\ w^T \mathbf{1} = 1\} \tag{2.49}$$

$$\boldsymbol{e} = \frac{1}{N}\sum_{i=1}^{N}\xi^i \quad (2.50)$$

Where, $\lfloor \alpha N \rfloor$ is the largest integer not exceeding αN,

N is the number of (time series) observation,

α is the significant level of VaR,

ξ^i is the portfolio return of period i,

e is the average return vector.

The authors solved the problem by using a numerical method. First, they smoothed out the local noisy and extract good global component of VaR function. Then, the problem in (2.47) – (2.50) was solved by standard off-the-shelf software for solution of non-linear programming problems. They found that the mean - VaR feasible set and its efficient portfolio frontier are not convex but the distance from the convex frontier is quite small which could be considered approximated convex. Mean-VaR efficient portfolio is substantially different from those of mean-CVaR and mean-variance. Therefore, mean-CVaR and mean-variance frontier give only a poor approximate of mean-VaR frontier. However, mean-CVaR and mean-variance could be an approximation of each other better than those of mean-VaR. Finally, all frontiers approximate each other better for high risk portfolios than those of medium and low risk because portfolios with high return are likely to consist of only one asset with high return.

Another approach for alternative optimising objectives is an MCDM approach to portfolio optimisation proposed by Ehrgott et al (Ehrgott 2004). The authors explained the motivations behind the development of this model were to address the criticism of the Markowitz model such that disregarded individual investors' preferences and the observation that most investors did not actually hold the efficient portfolio but an inferior portfolio. Therefore, based on several discussions with investors and investment analysts from Standard & Poor's Funds Services GmbH, Germany, they came up with their proposed model. The model was composed of five objectives covering the proxies of portfolio's expected return and of risk. The proposed objectives were 12-month performance (relative change of portfolio's value over 12 month), 3-year performance (relative change of the portfolio's values over 36 month), annual dividend (sum for all i of proportion of dividend of asset i paid by the highest price of the asset i in the last year), the Standard & poor's star ranking (German S&P fund performance raking) and volatility (portfolio variance). The five objectives were then combined into a single criteria optimisation problem as follows

$$Max_{w_i} \sum_{q=1}^{5} h_q u_q(f_q(\boldsymbol{w}) \qquad (2.51)$$

$$Subject\ to$$

$$\sum_{i=1}^{N} w_i = 1 \qquad (2.52)$$

$$w_i \geq 0 \quad \forall i = 1, \dots, M \qquad (2.53)$$

$$w_i - l_i y_i \geq 0 \quad \forall i = 1, \dots, M \qquad (2.54)$$

$$w_i - r_i y_i \leq 0 \quad \forall i = 1, \dots, M \qquad (2.55)$$

$$\sum_{i=1}^{N} y_i = m \qquad (2.56)$$

$$y_i \in \{0, 1\} \quad \forall i = 1, \dots, M \qquad (2.57)$$

Where, w_q is a positive weight for five decision maker specific utility,

$U_q(f_q(\mathbf{w}))$ is the q specific utility function of the decision maker (investor),

l_i is the minimum proportion of asset i in the portfolio,

r_i is the maximum proportion of asset i in the portfolio,

y_i = 1 if asset i include in the portfolio and =0 otherwise.

Note that relations in (2.54) and (2.55) are floor (minimum) and ceiling (maximum) constraints respectively. The authors proposed a novel local search algorithm called Two Phase Local Search and three meta-heuristics, namely, Simulated Annealing (SA), Tabu Search (TS) and Genetic Algorithm (GA) to solve the problem. In the proposed local search, they used two neighbourhood structures instead of one. The neighbourhood structures consisted of the member neighbourhood which were portfolios of the same percentage distribution of assets differing in at most one asset (this is a neighbourhood based on the 0-1 variable y_i) and the percentage neighbourhood structure which were portfolios of the same assets differing in at most two percent percentage values of assets (this is a neighbourhood based on changing the variable w_i). However, in their paper, the authors aimed to test performances and complexities of their proposed algorithms with the other algorithms, no out of sample or ex-post tests were made.

The authors found and reported that for the partially approximated large real case in which consisted of 190 funds and used approximate covariance values, the Two Phase Local Search had a good and stable performance (which are maximising utility value and minimising standard deviation) but slightly inferior to that of SA while the SA was inferior to that of GA. However, the Two Phase Local Search used a shortest CPU time. On the other hand, for the small real case which consisted of 40 funds with actual past covariance values, the two phase local search was the best both in the terms of performance and CPU time. For the randomly generated instances (simulations), the Two Phase Local Search had the best in both the terms of performance and CPU time.

2.4. Estimation Errors and Model Risk

A crucial part of the problem of the Markowitz model of portfolio optimisation lies on the estimation error of the necessary inputs. Typically, we assume that expected returns, risks, and correlations of assets to be included in a portfolio are known exactly and mostly are measured from historical data. The inputs are usually fed into the optimisation algorithm without recognising the uncertainty of their estimations. The estimations are made by various techniques. The simplest one is just plain historical averages (means) for the expected returns, historical variances for the risks, and historical covariance for the covariance matrices. More sophisticated estimations may be made by using some non-linear models. This likely invites model risks. Due to this estimation risk, portfolio optimisation in whatever choice of objective functions and optimising algorithms can only make a crude approximation of a true optimal portfolio. This makes portfolio optimisation models perform badly, especially for ex-post or out-of-sample results. Without the information about the estimation risk and how to manage it effectively, it is likely that professional investment managers often disregard to base their investment decision on the portfolio optimisation models we have discussed so far (Jorion 1992).

Jorion (1992) proposed a simulation approach in order to have information about the effects of estimation errors to ex-post performance of the Markowitz portfolio optimisation and to answer the question whether an optimal portfolio allowed holding of global assets statistically significantly improved performance over the prohibited one. The proposed simulation procedure was as follows:

Step 1: Compute the means and covariance matrix from the historical returns. Let T be the sample size and N be the number of assets. Perform portfolio optimisation.

Step 2: Assuming that the results from Step 1 have multivariate standard normal distribution, draw one random sample of N joint returns T times. These are T periods simulated returns.

Step 3: Estimate from the results in Step 3 a new set of means and a new set of variance-covariance matrix. Run an optimisation from the new inputs. The simulated optimal portfolio gives one observation in the distribution of the original portfolio.

Step 4: Repeat Step 2 and Step 3 until the approximation of optimal portfolio's distribution is at the predefined precision.

Lastly, by choosing a cut-off probability (e.g. 1%, 5% or10%) and removing the lowest return-to-risk ratios by the chosen cut-off percentage, we then have got a statistically equivalent portfolio.

The author used the simulation method to select a global bond portfolio. He chose portfolio based only estimated variances and co-variances but not estimated return. He asserted that doing so because variances and co-variances can be estimated with much more precision than expected returns. The selected optimal portfolio's ex-post Sharp Ratio (mean/standard deviation) was compared to those of US bond index and World bond index over the period of 1978-1988. F-test statistics were performed accordingly. He concluded that there was no evidence that the chosen portfolio outperformed the world index. On the other hand, the chosen portfolio performed significantly better than the US index.

Ong et al (Ong 2005) also recognised that the performance of mean-variance portfolio optimisation depended on the accuracies of forecasts of the return rate and the variance-covariance matrix. Their approach was to make better forecasts of those values. Instead of using just historical means and variance-covariance or regression techniques, they used the grey prediction model which incorporated the sequence curve under the small sample (Wang 2002) to predict the future return rate while used the probabilistic regression (Tanaka 2001) to predict the uncertainty risk. The proposed model was modified to have three objectives, namely to maximise Expected Return (predicting by the grey regression, to minimise Uncertainty Risk (predicting by the probabilistic regression) and Relation Risk (the sum of covariance matrix elements). The authors believed that the uses of the grey and probabilistic regressions were more appropriate to a small sample situation than those of ordinary regression. Since there were to optimise three objective simultaneously, they used a simple Pareto based MOEA to solve the problem. The approach was employed to select a portfolio which was composed of at most 6 stocks. There was only 6 period data to forecast the inputs from. They concluded that the proposed method provided more accurate and flexible results. An investor could select his alternative optimal portfolio based on the results of Pareto set.

An approach based on imposing some adding constraints to the selection process. DeMiguel et al (DeMiguel 2009) also well recognised it was much more difficult to accurately estimate means than that of covariance of asset returns, and the estimation errors of asset means (which represented asset expected returns) had more impact on portfolio weights and, therefore, aimed to minimise variance portfolio which relied on estimation of variances and co-variances. They proposed a general framework to find portfolio that perform well out-of-sample (ex-post) in the presence of estimation errors. The general framework relied on solving the traditional portfolio optimisation problem by minimising portfolio variance based on the sample variance-covariance matrix. The authors developed a new approach for determining portfolio weights in the presence of estimation errors by imposing constraints on the weights (shrinking portfolio weight vector). The constraints were imposed such that the norm of the portfolio weight vector was less than a given value (called a threshold). They used two kinds of portfolio weight vector, which are the 1-norm denoted by $\|\mathbf{w}\|_1$and the A-norm denoted by $\|\mathbf{w}\|_A$, given respectively

$$\|\boldsymbol{w}\|_1 = \sum_{i=1}^{N} |w_i| \tag{2.58}$$

$$\|\boldsymbol{w}\|_A = (\boldsymbol{w}^T \boldsymbol{A} \boldsymbol{w})^{1/2} \tag{2.59}$$

Where, **w** is portfolio weight vector,

$\mathbf{w}^T$is the transposeof **w,**

w_i is the i^{th} element of **w** as weight of asset i in the portfolio and

A is an n by n positive-definite matrix.

The authors formulated the proposed norm constrained minimum-variance portfolio as

$$Min_w \boldsymbol{w}^T \hat{\boldsymbol{\Sigma}} \boldsymbol{w} \tag{2.60}$$

$Subject\ to$

$$\|\boldsymbol{w}\|_x \leq \delta$$

Where, $\|\mathbf{w}\|_x$ is any norm (x-norm which x can be 1 or A or any subscription,

δ is a (constrained) threshold,

w_i is the i^{th} element of **w or** the weight of asset i in the portfolio and

A is an n by n positive-definite matrix.

They developed 8 portfolio strategies as follows:

1. 1-norm-constrained minimum-variance portfolio with δ calibrated using cross-validation over portfolio variances, denoted by NC1V.
2. 1-norm-constrained minimum-variance portfolio with δ calibrated by maximising portfolio return in previous period, NC1R.
3. 2-norm-constrained minimum-variance portfolio with δ calibrated using cross-validation over portfolio variances, denoted by NC2V.
4. 2-norm-constrained minimum-variance portfolio with δ calibrated by maximising portfolio return in previous period, NC2R.
5. $\mathbf{\Sigma}_F$-norm-constrained minimum-variance portfolio with δ calibrated using cross-validation over portfolio variances, denoted by NCFV.
6. $\mathbf{\Sigma}_F$-norm-constrained minimum-variance portfolio with δ calibrated by maximising portfolio return in previous period, NCFR.
7. Partial minimum-variance portfolio with k calibrated using cross-validation over portfolio variances, denoted by PARV.
8. Partial minimum-variance portfolio with k calibrated by maximising portfolio return in previous period, PARR.

Note that 2-norm and $\mathbf{\Sigma}_F$-norm are given respectively as follows

$$\|\boldsymbol{w}\|_2 = \sum_{i=1}^{N} w_i^2 \tag{2.61}$$

$$\|\boldsymbol{w}\|_{\Sigma_F} = (\boldsymbol{w}^T \boldsymbol{\Sigma_F} \boldsymbol{w})^{1/2} \tag{2.62}$$

Where, $\boldsymbol{\Sigma}_F$ is the covariance estimator obtained from a 1-factor model.

The partial minimum variance portfolio is an optimal portfolio obtained by applying the classical conjugate-gradient method. The author can prove that the 2-norm of the k^{th} partial minimum-variance portfolio is smaller than or equal to 2 norm of the short-sale-unconstrained minimum-variance portfolio for $k \leq N-1$ where N is the number of selectable assets.

The authors used 5 sets of data from 07/1963-04/2005 to test their proposed models against other 10 minimum-variance portfolio models from various researchers as follows

1. Equally-weighted (1/N) portfolio
2. Value-weighted (market) portfolio
3. Short-sales unconstrained mean-variance portfolio with risk aversion parameter $\gamma = 5$
4. Short-sales unconstrained Bayesian mean-variance portfolio with risk aversion parameter $\gamma = 5$
5. Minimum-variance portfolio with short-sales unconstrained
6. Minimum-variance portfolio with short-sales unconstrained from (Jagannathan 2003)
7. Minimum-variance portfolio with market as the single factor
8. Weighted average of sample covariance and identity matrix
9. Weighted average of sample covariance and 1-factor matrix

10. Parametric portfolio with a risk-aversion parameter of γ = 5 using the factors size, book-to-market and momentum.

They performed monthly out-of-sample tests for portfolio variances and portfolio Sharpe ratios by comparing all of the proposed models and the 10 models above. They found that the norm-constrained portfolio calibrated using cross validation over the return variances (NC1V, Nc2V, NCFV and PARV) was similar across 5 data sets and lower for those calibrated using criterion of maximum return of the portfolio in the past period (NC1R, NC2R, NCFR and PARR). They concluded that the variances of the norm constrained portfolio often had lower variances than those augmented models. And also the proposed 8 portfolio strategies mostly had higher Sharp ratios and lower portfolio variances than those of 10 comparing portfolio strategies.

Lutgens and Schotman (Lutgens 2010) dealt directly with estimation errors. They considered portfolio choice that was robust to the advice of multiple experts who employed different return models and proposed that a robust optimal portfolio should have maximised the performance over the least favourable return model. Their “robust’ approach assumed no preference or weights on the alternative asset return model but treated each of the model as equally plausible and fully taken into account each one for decision making. They considered two types of optimisation models namely, the optimisation model with point estimates of asset returns and the optimisation model with estimation uncertainty (range estimates). Mathematically speaking, for the point estimate optimisation, a robust investor will consider the worst case and maximise it as

$$Max_w \, Min_j \left(\boldsymbol{\mu}_j'\boldsymbol{w} - \frac{1}{2}\gamma\boldsymbol{w}'\boldsymbol{\Sigma}_j\boldsymbol{w}\right) \tag{2.63}$$

Where, $\boldsymbol{\mu}_j$ is the vector of estimated return of N assets (to be selected) by expert j, and j = 0 to J,

w is the vector of asset weights in the portfolio, $\boldsymbol{\Sigma}_j$ is the variance-covariance matrix of assets as estimated by expert j,

ϒ is the risk aversion parameter.

For the range estimate optimisation, a robust investor will consider the worst case and maximise it as

$$Max_w \, Min_j Min_{\mu \epsilon U_j} \left(\boldsymbol{\mu}_j'\boldsymbol{w} - \frac{1}{2}\gamma\boldsymbol{w}'\boldsymbol{\Sigma}_j\boldsymbol{w}\right). \tag{2.64}$$

Where, $\mathbf{U}_j$ is the uncertainty set of a particular expert j which contains range or parameter of the plausible values of $\boldsymbol{\mu}_j$.

The authors performed empirical tests by assuming that there were 5 experts (J = 4): Expert u forecasted based on past sample means; Expert c forecasted based on CAPM; Expert d forecasted based on CAPM with book value to market value; Expert e forecasted based on IAPM which was an international CAPM; And Expert f forecasted based on all factors from all of the aforementioned models. They ran optimisations by a bootstrap experiment. The results showed that for the point estimation case, the robust investor had a probability that his ex-post portfolio Sharp-ratio was more than or equal to the ex-ante by 56.6% and that his ex-post utility was more than or equal to the ex-ante utility by 55.7%. And for the range estimation case, the robust investor had a probability that his ex-post portfolio Sharp-ratio his ex-post utility were more than or equal to those of ex-ante by 98.5% and 99.0% respectively.

DeMiguel et al (2012) used information implied in prices of stock options to improve estimates of stock-return volatilities and correlations in order to improve the out-of-sample performance of portfolios which were measured in terms of portfolio volatility, Sharpe ratio, certainty-equivalent return and turnover, with the benchmarks being the 1/N portfolio (DeMiguel 2009) and four types of minimum-variance portfolios based on historical data. They found that using option-implied correlations and option-implied volatilities did not significantly improve portfolio performance due to high variability and instability of the estimates of implied volatilities and implied correlations. They also investigated the effects of adjusting the estimates of historical volatilities of stock returns using two sources of option-implied information. The first source uses the volatility risk premium of each stock. The empirical evidence showed that the portfolios with scaled volatilities using the volatility risk premium outperform the traditional portfolios in terms of Sharpe ratio and certainty-equivalent return, but with an increase in turnover. The second source uses the model-free option-implied skewness to scale volatilities in the same manner as for the volatility risk premium. The empirical evidence showed that portfolios based on implied skewness outperform the traditional portfolios in terms of Sharpe ratio and certainty-equivalent return more strongly, but this increase is also accompanied by an increase in turnover and portfolio volatility. Based on the empirical analysis, they concluded that prices of stock options had information that could be used to improve the out-of-sample performance of portfolios.

2.5 Chapter Summary

This chapter reviews literature related portfolio optimisation under realistic situations as development from the classical models proposed by Markowitz and

Black (collectively called the classical models) in the mid of 20th century. There are three main problems recognised by academic and financial communities. Firstly, the classical models' solution methods do not well handle some realistic constraints caused by the prevailing market customs and investment management practices. These constraints usually make the portfolio optimisation problems become NP-hard problems and can only be solved by approximate search algorithms.

Secondly, variances of returns may not be the best representation of risk especially for individual investors' utility functions. There have been other proposed candidates, e.g. expected- shortfall, Value at Risk, kurtosis, etc. However, this is only a matter theoretical perspective. And any of these functions can be handled effectively by meta-heuristics. Lastly, the inputs of the classical models have been historical means and variances, presuming that assets' returns are normally distributed with a constant means. Therefore, their expected returns and expected variances are the same as their historical means and variances. However, this assumption is not true especially in the short-run and causes "estimation errors". The estimation errors eventually make the ex-post outcomes of portfolio optimisation become suboptimal. There are recently a number of researches that have attempted to address these problems as we have discussed and that have resulted in better ex-post performances.

Chapter 3

Review of Solution Methods

3.1. Chapter overview

Most realistic portfolio optimisation models have no closed form solution. Inclusion of typically realistic constraints, such as cardinality constraints, round-lot constraints, etc., makes portfolio optimisation become a combinatory problem. Although they can be solved by exact solution methods, they need exhaustive searches and become NP-hard problems. If the complexity or the number of feasible choices is so great, the problems will not be able to be solved in a reasonable time. In such circumstance, we prefer approximate solutions to exact solutions.

This chapter reviews relevant Artificial Intelligent techniques and their application to solution methodologies of portfolio optimisation reported in the literature. The structure of this chapter is as follows: Section 3.2 is for the algorithms for exact solutions; Section 3.3 discusses on some heuristic methods; Section 3.4, Metaheuristic methods are reviewed; Local Search metaheuristics are briefly covered in Section 3.5; We then go on to cover Genetic Algorithms for portfolio optimisation in Section 3.6. In section 3.7; we discuss Multi-Objective Genetic Algorithm methodologies which are used extensively in this thesis; Lastly, in section 3.8, summary and research gaps are presented.

3.2 Algorithms for Exact Solutions

The standard Markowitz models with non-negativity constraints are NP-hard, only with small problem size, i.e. the small number of assets (N), can be solved within reasonable time for an exact solution using standard optimisation software (quadratic optimisation tools) (Maringer 2005, Wolfe 1959 cited in Crama 2003). In Jagannathan 2003 (cited in Maringer 2005), the non-negativity constraints are incorporated by modifying the covariance matrix without reducing complexity. Some ad hoc methods take advantage of the special structure of the covariance matrix (see Perold 1984 and Bienstock 1996 both cited in Crama 2001). Other researchers investigated some techniques that can be solved only by models with only a subset of constraints, e.g. Dembo 1989 using network flow models, Konno 1991 using linear programming models with embedded absolute deviation approach to measure risk, Takehara 1993 (cited in Crama 2003) using an interior point algorithm, Bienstock 1996 (ibid.), and Horniman 2001 (cited in Stein 2005) using branch and cut approaches (as a subset of Mixed Integer Quadratic Programming – MIQP).

Quadratic Programming (QP) problems consist of an objective function that allows for an additional quadratic form of the decision variable, and a number of constraints which are expressed in linear equality or inequality with respect to the decision variables. A general problem form can be stated as

$$min_x \boldsymbol{f}'\boldsymbol{x} + \tfrac{1}{2}\boldsymbol{x}'\boldsymbol{H}\boldsymbol{x} \qquad (3.1)$$

Subject to

$$\boldsymbol{Ax} = \boldsymbol{a}$$

$$\boldsymbol{Bx} \leq \boldsymbol{b}.$$

Where ***f*** is a 1 by N vector,

H is an N by N matrix,

x is an N by 1 vector of decision variables,

A is an M (any number) by N matrix,

B is an M by N matrix,

a is a vector and

b is a vector.

This can be applied to the Markowitz model (only non-negativity constraint imposed) as follows

$$min_w \ \lambda \boldsymbol{r}'\boldsymbol{w} + \tfrac{1}{2}\boldsymbol{w}'2(1+\lambda)\boldsymbol{\Sigma w} \quad (3.2)$$

$subject\ to$

$$\mathbf{1}\boldsymbol{w} = 1$$

$$-\boldsymbol{I}\boldsymbol{w} \leq \mathbf{0}.$$

Where λ is the parameter which measures trade-off between return and risk,

I is an N by N identity matrix and

1 is a 1 by N unity vector.

If we set λ = 0 then the result will be the minimum variance portfolio. On the other hand, if we set λ = 1 thus put all weight to maximization of expected return, the result will be the portfolio with the highest possible yield. In this case, the Markowitz model disallows negative weights without upper limit of the asset weights. This will result in a portfolio with single asset which has the highest expected return. To construct the entire efficient portfolio, we would increase λ in sufficiently small steps from zero to one and solve the quadratic optimisation for these values (Maringer 2005).

Although the model with non-negativity constraint cannot have a closed form solution, it can be solved efficiently by specialised algorithms and other ad hoc methods. A popular method for Portfolio Optimisation with non-negativity constraint is Quadratic Programming (QP) which is an iterative numerical method (Scherer 2005). However, imposing other constraints (e.g. the integer constraint or maximum number of asset constraint) will increase the problem complexity and size making them difficult to solve by Quadratic Programming, mixed integer non-linear programming or other exact solution algorithms, within a reasonable time (Busetti 2000). The portfolio optimisation problems with realistic constraints are NP hard problems, especially for those of exact solutions. The methods for exact solutions require complete enumerations where all possible and valid values for the decision variables are tested. The problem will be to select *k* out of *N* assets and optimize their portfolio weights. Only the complexity of selection of asset alone is O (C (N, k)) e.g. selecting 10 out of 100 assets come with C (100, 10) or 1.73×10^{13} alternatives. Moreover, for each of the alternatives, the optimal weights must be determined. If we assume that the weight must be zero or multiples of 0.1, we will have 10^{10} possible weight structures for each alternative. Obviously, the problem size can quickly get out of our present computational power. There are two ways to cope with the NP hard problems. First, realistic models of portfolio optimisation are solved by approximate algorithms, which do not guarantee finding the optimal solution, but search for good enough solutions in a significantly reduced computational time (Blum

2003). Second, approximately simplified unrealistic portfolio optimisation models are constructed instead so that they can be solved by standard methods or algorithms.

3.3. Heuristic Methods

In the process of investment decision, portfolio managers usually face an abundance of choices of investment assets. Also, they may need to make timely decisions in a rapidly changing financial market. This represents a tough optimisation problem, which continues to present a challenge for efficient optimisation solution techniques (Maringer 2005). A variety of different techniques have been employed to solve the portfolio optimisation problem. The main drawback of techniques for exact solution is that the number of combinations of states that must be searched increases exponentially with the size of problem and becomes computationally prohibitive (Crama 2003). Furthermore, these techniques are poor in handling the nonlinear objective and constraint functions, and several assumptions are generally required to make the problem solvable using reasonable computational resources (Maringer 2005). Alternatively, some heuristic-based techniques use algorithms to find approximate solutions for problem instances of NP-hard problems in a reasonable time (Blum 2001). By using heuristics, the optimisation problems can be tackled in polynomial time with a trade-off for their optimality. In some circumstances of the practicality, the speed to reach the acceptable approximate solutions is very critical. Feasible near-optimum solutions are acceptable but untimely ones are not. The simple heuristic solution approaches are based on specialised techniques that work particularly well for a given problem but are only of limited applicability to other problems (Blum 2003). Furthermore simple heuristics, based on greedy search algorithms, tend to stop in inferior local optima.

Heuristics are approximate algorithms. Basic or classical heuristics are greedy algorithms. In combinatorial optimisations, the basic heuristics include local search algorithms and constructive algorithms. Local search algorithms start from initial solutions and repeatedly try to substitute the current solution with a better one in an appropriate vicinity or neighbourhood of the current solutions. Constructive algorithms generate a solution from an initially empty solution by adding components until a solution is completed.

An example of a heuristic approach is proposed by Elton and Gruber (Elton 1997) for determining optimal portfolio which in fact approximates the solution of the Markowitz model by using CAPM as a tool of selecting assets. The algorithm can be described as follow:

Step1: For all asset I, calculating a ratio based on CAPM, which is later also known as "Treynor Ratio" and given by

$$TR_i = \frac{E(r_i) - r_s}{\beta_i} \qquad (3.3)$$

Where, E (r_i) denotes the expected return of asset

In order to select any asset to be included in the optimal portfolio, we need to calculate for all potential assets.

Step 2: The results are ordered from the highest value to the lowest in which is preferred more to less.

Step 3: Estimating a regression from a time series data of the empirical market model given as follows

$$r_{it} = \alpha_{it} + \beta_i R_{Mt} + \varepsilon_{it} \quad (3.4)$$

Whereε_{it} is the specific return on asset i which is the residual of the regression.

Now, we have and keep the value of ε_I to be used in Step 5.

Step 4: Add the highest ranking asset into the portfolio.

Step 5: Try to add another asset which has the highest value of Treynor Ratio and has not yet included into the portfolio. Then, calculate the threshold denoted by C_i which is expressed as follows

$$C_i = \frac{\sigma_M^2 \sum_{j=1}^{i} \frac{(E(r_i) - r_S)\beta_j}{vat(\varepsilon_j)}}{I + \sigma_M^2 \sum_{j=1}^{i} \frac{\beta_j^2}{vat(\varepsilon_j)}} \quad (3.5)$$

Step 6: Compare C_i with TR_i, if C_i<= TR_i , including the asset i into the portfolio. Let C^* = C_i, and then, go to Step 5. Else exclude the asset i, end the selection and then proceed to the next step (Step 7).

Step 7: With the list of included assets (let's assume that we have p assets), we calculate the proportion (weight) of asset i to be held in the portfolio by

$$w_i = \frac{Z_i}{\sum_{j=1}^{P} Z_j} \quad (3.6)$$

Where

$$Z_i = \frac{\beta_i}{var(\varepsilon_i)} \left(\frac{E(r_i) - r_s}{\beta_i} - C^* \right) \quad (3.7)$$

In most combinatorial optimisations, heuristic methods can reduce the computation complexity to at most polynomial time (but yield solutions that are not sure to be global optima). A drawback of the basic heuristics is that they tend to be trapped with local optima far inferior to the true or global optima and, for most of time, is not considered good enough. Metaheuristics are to combine basic heuristic methods with some Intelligence or guided strategies aimed to avoid the traps.

3.4. Metaheuristic Methods

In order to overcome the above limitations, researchers in the last decade have focused much attention on metaheuristic solution techniques (Blum 2003 and Yilmaz 2011). Metaheuristics are general intelligent searches that could find the ways out of local optima. Despite their intelligence and generality, performances of metaheuristics depend on problem settings. Over the last decade a number of research publications have been reported on applications of metaheuristic approaches addressing some of portfolio management and optimisation issues.

Essentially, metaheuristics are algorithms for exploring search spaces by using guided strategies that have a dynamic balance between the exploitation of the accumulated search experience (i.e. intensification) and the exploration of search space (i.e. diversification) (Blum 2001). Metaheuristics can be classified into two categories, namely local search metaheuristics (LSMs) and evolutionary algorithms (EAs). LSMs begin with a single solution that is subsequently replaced by another (often but not always the best) solution found in the neighbourhood. They are called exploit-oriented (or intensification) methods because they are often allowed to find a local optima solution. However, they are different form local search algorithms of basic heuristics in such a way that they have some mechanism to strategically guide the search away from trapped local optima. Conversely, EAs make use of a randomly generated population of solutions. The initial population is improved through natural evolution/selection processes. In the processes, the whole or part of population is replaced by newly generated offspring (often the most suitable ones). As a result, EAs are often called exploration-oriented method. LSMs can be categorised into various models, based on guiding techniques, which include Simulated Annealing (SA) (Kirkpatrick 1983), Tabu Search (TS) (Glover 1990), Greedy Randomised Adaptive Search Procedure (GRASP) (Feo 1995), and Variable Neighbourhood Search (VNS) (Mladenovic 1997). On the other hand, EAs include Genetic Algorithms (GA) (Goldberg 1988), Evolution Strategies (ES) (Rechenberg 1989), Genetic Programming (GP) (Koza 1999), Ant Colonies (Dorigo 1997), Estimation of Distribution Algorithms (Larranaga 2002), and Scatter Search (Glover 2000). Since there is active research around the world to find new heuristic techniques, the lists are by no mean exhaustive. Moreover, there are also hybrids and other metaheuristics that can fall into neither category (Alba 2005).

Even though metaheuristics are not problem specific but in order to reach good solutions, they often need to make use of detailed knowledge of the problem domains. As a result, most efficient metaheuristics are not reusable for different problems or even different instances of the same problem, without redevelopment or in some cases, adjustment of relevant parameters.

3.5. Applications of Local Search Metaheuritics

The main focus of this research is portfolio optimisation using multi-objective EAs, however for completeness we briefly discuss here some of the applications of popular LSMs, namely Ant Colony (AC), Tabu Search (TS) and Simulated Annealing. The readers are referred to the references (Dorigo 1997, Glover 1990 and Kirkpatrick 1983) for the detailed working principle of these algorithms.

Ant Colony Search is the imitation of behavior of ants that enable them to find shortest path between food sources and their nest. While moving to/from the nest ants deposit pheromone on the ground. Using the concentration of pheromone, they decide the likelihood of which direction to go. The shortest paths are likely to be used more often within the same period of time. They are likely to have more amount of pheromone on the trail than those used less. This reinforced strategy enables ants to find shortest paths between the nest and the food source (Blum 2003).

Maringer (2005) applied ant colony algorithms to solve optimisation problems in small portfolios (with cardinality constraints). He asserts that the problems can be regard as Knapsack problems with some modifications. In order to make the problem have only a single objective function, he set the objective function to be the proportion of risk premium over the risk-free asset and the standard deviation of portfolio return which is called the “Sharp Ratio” denoted by SR_P. Assuming that there exists a market M with N assets (M is a set of assets or M = {1,…,N}) and the investor will choose k assets to be included in the portfolio P.

The author tested for two cases of choosing three stocks (k = 3) and ten (k = 10) stocks respectively from S&P 100 (with 161,700 and 1.73×10^{13} alternatives respectively). In the case of standard parameter setting (no evaporation of the pheromone) for k = 3 only 16 % of all run in which the global optimum was found and for k = 10 in just 2 out of the 1,000 independent runs that the global optimal was found. However, when the parameters were set to the appropriate values (with some evaporation), for k = 3, about two third of the run that global optima were found, and for k = 10, from half and two third of the run that global optima were found (Maringer 2005). Thus, Ant Colony Search is not quite efficient for portfolio optimisation problems, especially in a large scale. Also, it is not flexible to incorporate more than one objective into the problem.

Simulated Annealing (SA) is the oldest among the metaheuristics. It allows moves toward worse solutions in order to escape from local optima. The probability of such worse moves is diminished in a similar way to molecules slowing down when the system's temperature is cooling down (which the name is derived from) (Blum 2003).

Crama and Schyns (Crama 2003) applied simulated annealing to solve complex portfolio selection problems with floor, ceiling, turnover, trading and cardinality constraints. They encoded a solution of the problem as an n-dimensional vector X whose element x_i represents the holding of asset i in the portfolio. The quality of a solution was measured by the variance of the portfolio. They used specific approaches to handle specific classes of constraint either by explicitly restricting the solutions to be in feasible region or by penalizing infeasible solutions. They found that the algorithm could approximate the optimal portfolio frontier for medium size problem (151 assets) within acceptable computing time and could handle more classes of constraints than those of classical approaches. Also it was quite versatile to apply to different measures of risk other than variance as well as different covariance matrix properties. However, the algorithm still needed to customise and to delicately fine tune parameters to account for different classes of constraints.

Maringer (2005) adapted SA to solve portfolio optimisation problems with transaction costs. The results showed that in presence of transaction costs the optimal portfolio structures were severely affected. Fixed costs could lead to a substantial reduction of different assets which should have been included in a portfolio as well as for proportional cost and compound cost schemes (Maringer 2005). Chang (2000) found that, in terms of mean percentage errors, on average SA performed better than TS but worse than GA.

Tabu Search (TS) is a local search heuristic. The concept is a systematic operator that, given a single starting solution, generates other possible solutions which are the neighbourhood of the single starting solution. The solution may not be feasible in this stage. The best solution may be the very first that improves the solution in the neighbourhood or may be the last one upon completing enumeration. The best solution is chosen to become a new starting solution for the next round as the process is repeated.

Chang et al (Chang 2000) applied TS to cardinality constrained portfolio optimisation (with also minimum portion constraint). They solved the problem by three meta-heuristic methods, namely Genetic Algorithm (GA), Tabu Search (TS), and Simulated Annealing (SA). SA was the next best. TA reported the highest mean percentage errors. Busetti (Busetti 2000) used Tabu Search/Scatter Search tools in a decision support system developed by Decisioneering Incorporated to handle portfolio optimisations with cardinality constraints. The results were then compared with those of Genetic Algorithms (GA). The authors found that Tabu/scatter search method was unsuitable for optimisation portfolio with cardinality constraints (of the size of 40 assets). Therefore, he concluded that GA was better than Tabu/scatter search for this application and problem size. Moreover, the GA applied to portfolio optimisation was effective and robust with respect to quality of solution and speed of convergence. It was also more versatile by not relying on restrictive properties of the model, by ease of new constraint addition and by ease of the objective function's modifications. Contrary to other metaheuristic methods, the needs for tailoring, customising, and fine-tuning are not an issue for GA, even though these may improve performance of the model to an extent but not necessarily.

3.6. Evolutionary Algorithms

Evolutionary Algorithms (EA) are population based heuristic algorithms. Genetic algorithm (GA) is the foundation of EA as most of the other evolutionary algorithms can be viewed as variations of GA. In GA, solutions are represented as chromosomes to be bred by crossover, or modified by mutation. Selection processes are used to find optima solutions imitating the natural selection of survival of the fittest (Maringer 2005). During the evolution process, the populations change and evolve through the natural selection. Individual chromosomes which are more successful adapting to the environment will have a better chance to survive and thus to breed. The less fit individuals, however, are eliminated from the gene pool. Thus, the fit genes will spread and increase from generation to generation. The good characters from successful parents may produce even better offspring which become increasingly adapted to the environment. The basic step of a simple GA is shown in Figure 3.1.

```
Generate an initial population
Evaluate fitness of individuals in the population
REPEAT
        Select parents from the population
        Mating (Crossover) parents to produce offspring
        Evaluate fitness of the offspring
        Replace some or all of the (previous) population by the offspring individuals
UNTIL  a satisfactory solution has been found (or a number of generation have been reached)
```

Figure 3.1: Genetic Algorithm

Busetti (2000) compared GA with Tabu search and found that GA performs better for portfolio optimisation problems in the problem setting. Chang (2000) applied stocks in Hang Seng (31 names), DAX (85 names), FTSE (89 names), S&P (98 names), and Nikkei (225 names), and found that on average GA is the best in terms of mean percentage errors compared to those of SA and TA. Streichart et al. (Streichart 2004a) apply the Multi-Objective Evolutionary Algorithm (MOEA) to solve portfolio optimisation problem. However, they were not the first group to applied Evolutionary Algorithms to solve the problem. Tettamanzi et al. (cited in Arnone 1993, and Loraschi 1995a and 1995b) transformed the multi objective optimisation problem into a single-objective problem by using a trade-off function (therefore not a true multi-objective). In their paper, they compared the performances of different GA representations of portfolio optimisations with several combinations of the constraints on the Hang Seng data set with 31 assets. The representations were binary bit-string based genotypes or gray-code encoding and real-valued genotype. They also investigated the size of the bit string from 32 bit 'continuous' representation to 7 bit 'discrete' representation. They also compared GA with and without Lamarckism (hybrid GA in which the genotype can be modified, not only being removed from the population) and Knapsack GA (KGA) with and without Larmarckism. The constraints imposed on the optimisation problems were cardinality and integer (discrete) constraints. The results showed that KGA produced better results, as well as converged faster than ordinary GA due to more efficient removal of surplus assets. This conclusion was drawn from the fact that, in the problems without constraints, both GA and KGA performed almost the same. They also found that GAs without Larmarckism tended to prematurely converge because the neutrality of the search space, i.e. the space is relatively even space or gradually changed in values in all directions, caused the GAs to be trapped in the sub-optimal search space. KGAs, even without Larmarckism, did not have this tendency. For the bit strings, on average the real value coding performed worst in all problem instances. Also, the differences were not much if a constraint was not added. The discrete 7-bit string performed better than the 'continuous' 32-bit string because the mutation and crossover operators were more effective. Thus, the hybrid KGA with 7 - bit gray-coding and Larmarckism was the best in the most problem instances with real-world constraints. In another paper (Streichert 2005), the same group of authors introduced an alternative hybrid encoding for evolutionary algorithms, which combined both 'continuous' real value and 'discrete' binary value together. The algorithm then was compared with the different EA representations. When the algorithm and the other EAs without Larmarckism were applied on the problem with only cardinality constraints, the algorithms performed better than those of standard EAs. However, after introducing Larmarckism into the algorithms as well as into all standard EAs, the algorithm's performance was only comparable to other standard EAs. Similar to the results in the previous paper, in problems with many constraints, the algorithms with Larmarckism were among the best of all due to their ability to remove some of the neutrality in the search space. In other words, Larmarckism expands the search space by allowing self-modification of genetic strings in addition to altering in crossing-over operations.

Gomez et al (Gomez 2006) proposed a Hybrid Evolutionary Algorithm which was in fact an evolutionary Simulated Annealing. They used population of SA instead of just a single solution. The best of the population's solutions were selected and retained. The worst were discarded and replaced by a clone of the best ones or some new solutions with properties of the best solutions. The hybrid approach was used to solve portfolio optimisations with budget constraints, floor and ceiling constraints, purchase constraints, sale constraints, trading constraints and cardinality constraints. They found that increasing the number of constraints would also increase the time to find the optimal portfolio and would cause more noise on the efficient frontier. By comparing to those of the standard SA, the hybrid algorithm's was better. It increased the average of return by 1.7% and reduced the average variance by 41%. The curve of the portfolio frontier was also smoother.

3.7. Multi-Objective Evolutionary Algorithms

Most of meta-heuristics and standard Evolutionary Algorithms are for single objective optimisation. In the case of portfolio optimisation which usually has two objectives (to maximise mean and to minimise variance), we need to modify the objectives so as to reduce to a single objective, e.g. Sharp ratio, mean minus variance, etc. However, it becomes very inconvenient to handle optimisation with more than two objectives. Even though many objectives can be combined into one objective by assigning appropriate weights to the objectives, determining appropriate weights is not an easy task.

Multi-Objective Evolutionary Algorithms (MOEA) are based on the concept of Pareto optimality. Two solutions may not have the same bi-objective values but they may be as good as one another in the sense of Pareto dominance in which they both are non-dominant solution. Generally, a Multi-Objective Optimisation problem can be stated as follows

$$\boldsymbol{Minimize} \qquad \boldsymbol{T}(\boldsymbol{x}) = (\boldsymbol{f_1}(\boldsymbol{x}), \boldsymbol{f_2}(\boldsymbol{x}), \ldots \ldots, \boldsymbol{f_m}(\boldsymbol{x})) \tag{3.8}$$

$\boldsymbol{subject\ to}$

$$\boldsymbol{e}(\boldsymbol{x}) = (\boldsymbol{e_1}(\boldsymbol{x}), \boldsymbol{e_2}(\boldsymbol{x}), \ldots \ldots, \boldsymbol{e_n}(\boldsymbol{x})) \tag{3.9}$$

Where, T **(x)** is a vector of m objectives,

f_1 to f_m are the m objective functions (also elements of vector T(x)),

x is a vector of decision variables,

e **(x)** is a set of n constraints,

e_1 to e_n are the n constraints, $e_j(x) \geq 0$ for all j.

A Pareto optimal set of solutions is composed of non-dominant solutions. Mathematically, a decision vector **u** is said to strictly dominate another **v** if

$$f_i(\boldsymbol{u}) \leq f_i(\boldsymbol{v}) \forall_i = 1, \ldots \ldots, m \tag{3.10}$$

And

$$f_i(\boldsymbol{u}) < f_i(\boldsymbol{v}) \quad \exists_i = 1, \ldots \ldots, m \tag{3.11}$$

The algorithmic procedure of MOEA is the same as that of GA (or general Evolutionary Algorithms) except when it comes to evaluating the fitness function. A pseudo code for general MOEA for portfolio optimisation is shown in Figure 3.2 which is adapted from Fieldsand (2004).

```
Let T := maximum number of interation
Let H is set of sets of portfolios defining the c different estimated frontier
Let t:= 0, H^t_k = { }    ∀k = 1,….., a
H^t_{k,1} := random_portfolio (k)    ∀k = 1,……,a
While (t < T)
        k = U (1,a)
        w:= Select (H^t, k)                    // w is a asset weight vector
        w  := adjust (w)
        y  := evaluate(w)
        H^{t+1} = check_insert_remove (H^t ,w, y)
        t := t + 1
END
```

Figure 3.2: Pseudo-code for General Multi-objective Evolutionary Algorithm for Portfolio Optimisation Problems

So far, there have been a good number of varieties of MOEA that have been applied to many science, engineering, and business problems. Brief descriptions of some well-known MOEA algorithms are as follows.

MOGA was proposed by Fonceca and Fleming (1993). In this algorithm, each individual is given a rank according to the number of individuals in the population which it dominates, thus the non-dominated individuals have the lowest rank of 1. The population is then sorted according to each individual's rank. The fitness is determined by interpolating from the best rank to the worst. The individuals with the same rank will be sampled at the same rate by having their fitness averaged.

NPGA (Niche Pareto Genetic Algorithm) was proposed by Horn (Horn 1994). It uses a tournament selection scheme based on Pareto dominance. Instead of simple comparing two individuals in the population, the algorithm draws two individuals and then compares them with a random sample of the population. If one of them is non-dominated and the other is dominated with respect to the random sample, the non-dominated will be selected. On the other hand, if the both are either non-dominated or dominated, the tournament will be decided through fitness sharing in the objective domain. Because the decision of Pareto ranking is based on a segment of the population, the algorithm is quite fast compared to MOGA.

NSGA (Non-dominated Sorting Genetic Algorithm) was proposed by Srinivas and Deb (Srinivas 1994). In this algorithm, each individual in the population is ranked using Pareto-dominance. The non-dominated individual are then classified in to one category and assigned a dummy fitness value proportional to the size of population. The category is separated from the population. Another search for non-dominated individuals is performed until all of the population is ranked. In order to maintain population's diversity, individuals in each category share their dummy fitness value. Its computational complexity is $O(mN^3)$, where m = the number of category (niche count) and N is the population's size.

PAES (Pareto Archived Evolutionary Strategy) was proposed by Knowles et al (Knowles 1999). It is not a kind of Genetic Algorithm (GA) but a kind of Evolutionary Strategy (ES) in which one parent reproduces one child. The parent and its child are compared after the child has been produced. The child is accepted to the population if it dominates its parent else it is discarded and a new child will be produced. However, if no one can dominate the other, the one that is more diverse among the obtained solutions will be preferred. This can be achieved by comparing the child to the non-dominated solution archive. And if it is not dominated by any of the member in the archive, it and its parent will be checked to find which one resides in the less crowded region of the parameter space. This diversity maintenance is achieved by dividing the objective space up into grids. Solutions are put to the grids according to their objective values. A crowding measure based on the number of solution in and around each grid location is used to maintain diversity. The algorithm has a lower computational complexity compare to MOGA, NPGA and NSGA because every solution in the population is only compared to its parent and all member of the archive.

SPEA (Strength Pareto Evolutionary Algorithm) was introduced by Zitzler and Thiele (Zitzler 1999). It employs an external achieve to preserve the non-dominated solution (which is considered as an elitism mechanism). The external archive then collects all no-dominated individuals for each generation. The individuals in the archive are to be tested for dominated individuals and deleted if found. If the number of individuals in the archive exceeds the predefined upper limit, some of non-dominated individuals will be deleted based on their phenotypic closeness to other members of the group. This is a mechanism to preserve diversity. The fitness of an individual in the population is calculated by summing up the strength values, which are the number of the other individuals in the population which a particular individual has dominated, divided by the population size plus one (ranged from zero to one) of all archive members it dominates. In the reproduction, both current population and the archive population are combined together to form one population. Then the parents are selected and mated to generate the offspring population. Its computational complexity is O (N^2), where N is the population's size.

NSGA2 is an improvement to the NSGA algorithm also proposed by Deb et al (Deb 2001). The main criticism of NSGA is the computational complexity. The NSGA2 is composed of two main loops. First, for each individual i, two values are computed, namely the number of individuals which dominates it (denoted by n_i) and the number of individuals which it dominates (denoted by s_i). Then, the non-dominated individuals are put in a separate list called the current front. Second, the current front from the first loop is traversed for every individual. For each individual on each member j, when it is visited and its own n_j is decreased by one. If n_j becomes zero, it will be put in a separate list. When all individuals of the current front have been visited the new separate list becomes the current front. As a result, NSGA2 requires only O (N^2) computation while NSGA requires O (mN^3). The fitness assignment, reproduction, and selection remain the same as for NSGA.

SPEA2 is also an improvement of SPEA. It was designed by Zitzler et al (Zitzler 2002). It is different from SPEA in many respects. First, its archive has a fixed size. The archive may fill with some dominated individuals if the number of non-dominated is less than that of dominated individuals. And if the number of non-dominated ones is greater than the archive's size some of non-dominated individuals will be excluded by truncation. Second, the truncation method truncates individuals which have the minimum distance to another individual. If more than one individual have the same distance, the second smallest distance will be considered. Third, the fitness assignment in SPEA2 is defined to take in to account both dominated and dominating individuals. Individuals in both archive and the population are assigned a strength value representing the number of individuals. The raw fitness value is calculated as the sum of the strengths of the dominating individuals in both archive and the population. Thus, the raw fitness is to be minimised. Fourth, Density information is included to the fitness function by computing the inverse of the distance to the k^{th} nearest neighbor. It then is used to further discriminate between individuals. Fifth, only members of the archive are allowed to mate and to be selected.

Zitzler et al (Zitzler 1999) compared a number of MOEA by using metrics of performance which capture three aspects of performance, namely the distance of the non-dominated set (should be minimised), the distribution of solutions (should be maximised), and the range of values covered on each objective (should be maximised). Using two objective problems as test problems which were followed a guideline proposed by Deb (1998), they found that among the other algorithms SPEA was the best in the terms of distance from Pareto optimal front. The next best was NSGA. NPGA was the worst of the mentioned three algorithms. According to the test problems, multimodality and deception seemed to cause the most difficulty for MOEA. Moreover, they also found that elitism, which was incorporated into SPEA, was a crucial factor contributing to a MOEA's performance. By inclusion of elitism to other MOEA, it was improved significantly.

3.8 Fuzzy Logic

Fuzzy Logic can be viewed as an extension of multi-value logic to include probabilistic theory. Fuzzy logic is concerned with approximate modes of reasoning rather than precise ones as for classical logic (or crisp logic). In fuzzy logic, truth is described as a matter of degrees. In other words, truth in fuzzy world is revealed as grey scales rather than black or white as in the world of classical logic. An advantage of fuzzy logic in the realm of modeling is that it allows computing with linguistic variables. Fuzzy logic also can make inferences from qualified propositions, especially from probability-qualified propositions. This is crucial for managing uncertainty in expert systems and in the formalization of common sense reasoning. The values of a linguistic variable can be derived from a primary term such as “hot” and its antonym such as “cold” (as well as a collection of modifiers such as “not”, “very”, “more or less”, etc.), and the connectives “and” and “or.” The values can be generated by a context-free grammar. Moreover, each value of a linguistic variable represents a possibility distribution which can be calculated from the predefined possibility distributions of the primary term and its antonym through the use of attributed grammar techniques (Zadeh 1988).

Contrary to the classical set theory, in the fuzzy set theory, membership of fuzzy set A of universe *X* is defined by the membership function (of A) denoted by (Negnevitsky 2005)

$$\mu_A(x): X \rightarrow (0,1) \tag{3.12}$$

Where

$\mu_A(x) = 1\ if\ x\ is\ totally\ in\ A$;

$\mu_A(x) = 0\ if\ x\ is\ not\ in\ A$;

$0 < \mu_A(x) < 1\ if\ x\ is\ partly\ in\ A.$

For an example, according to our sensory feeling and knowledge of temperature, we can produce fuzzy sets of hot, warm and cold. Thus, the universe of discourse of temperature in degree Celsius consists of three sets: cold, warm and hot. The fuzzy sets can be graphically drawn as Figure 3.3. In this instance, a temperature of 20 degree Celsius is a member of Cold temperature with a degree of membership 0.3 and at the same time it is also a member of Warm temperature with a degree of 0.3. This means that the temperature of 20 degree Celsius has a partial membership in multiple sets.

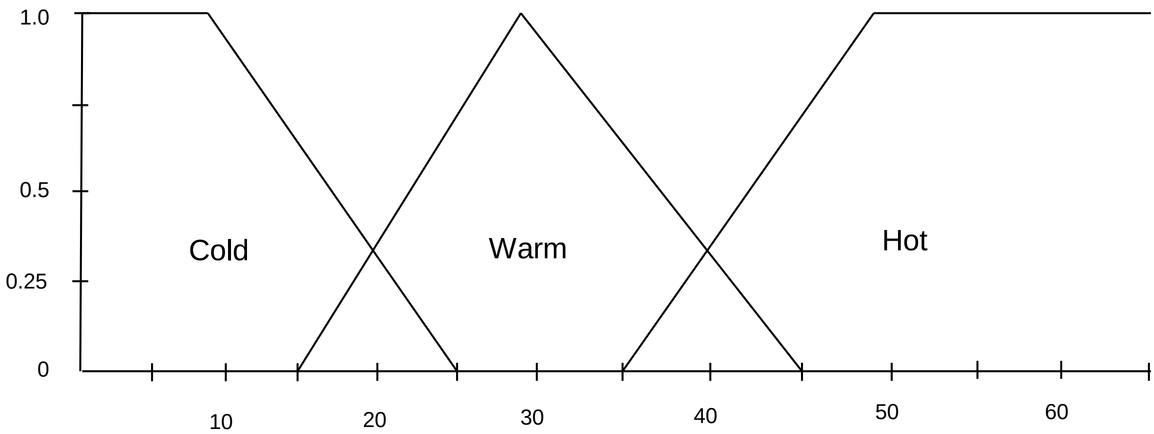


Figure 3.3 Fuzzy sets of Cold, Warm and Hot

Fuzzy logic has been applied extensively in engineering and in social science. For engineering applications, it is successfully used as control logic in various control systems as well as decision mechanism in a number of expert systems (Zadeh 1988). Also it has widely been applied to various fields of social science such as Economics, Political Science, etc. There are at least five reasons that fuzzy logic has been used in social science research. Firstly, it can handle vagueness systematically. Secondly, many variables in social science are categorical variables which are often turned out to be matters of degree. Thirdly, it can analyse multivariate relationships beyond conditional means and the general linear model. Fourthly, it is more suitable to social theoretical concepts which are frequently expressed in logical and set-wise terms than those of statistical models which rely on continuous variables. Lastly, fuzzy logic can rigorously merge set-wise thinking and continuous variables (Smithson 2006).

Fuzzy set theory is applied to construct an intelligent system with fuzzy inference. In a complicated system dedicated to a realm of knowledge, the intelligent system may sometimes be called an expert system. Fuzzy inference is a process of mapping a given input to an output by applying the fuzzy set theory. The most popular fuzzy inference technique is Mamdani method which consists of 4 steps (Negnevitsky 2005). First, fuzzification is to determine the fuzzy membership degree of the crisp inputs to each applicable fuzzy set. Second, rule evaluation is to take fuzzified inputs and then apply to the antecedents of the relevant fuzzy rules. Third, aggregation of the rule outputs is the unification procedure of the outputs of all rules. In this step, we take the membership functions of all rule consequents which are previously clipped and blend them together into a single fuzzy set. Fourth, the last step is defuzzification. In this step, fuzziness is aggregated into a single crisp number. The most popular method is the centriod technique. The techniques is to find centre of gravity (COG) to represent the aggregate output membership function (as a crisp number) which can be expressed mathematically over a sample of points as (Negnevitsky 2005)

$$COG = \frac{\sum_{x=a}^{b} \mu_A(x)x}{\sum_{x=a}^{b} \mu_A(x)} \ , \qquad (3.13)$$

or over a continuum of points as (ibid)

$$COG = \frac{\int_{x=a}^{b} \mu_A(x)x}{\int_{x=a}^{b} \mu_A(x)} \ . \qquad (3.14)$$

Where a is the lowest value of the lowest fuzzy set,

b is the highest values of the highest fuzzy set.

The readers are referred to (Mendel 2001) and (Negnetivitsky 2005) for the detailed working principles of the fuzzy systems.

3.9. Summary and Research Gap

There are a number of researchers that have attempted to find and develop the most suitable solving algorithms, and most of the researchers turned to metaheuristics. Metaheuristics have showed that they can solve the problems effectively. Genetic Algorithms have an advantage due to their simplicity, adaptability and flexibility. Moreover, they are quite effective especially for large scale portfolio optimisation problems. They are also extendable to MOEAs which can be effectively used to solve multi-objective problems. Among the MOEAs, according to the previous research, SPEA2's performance is likely to be the best both in the terms of the closeness to the real Pareto front and the diversity of solutions. Secondly, the classical models' objective functions may not reflect the real investors' utility functions or reflect their intentions. Since the utility functions or the intentions are hidden inside the investors and are not likely to be homogeneous, the proposed novel objective functions are only from theoretical points of views, if not purely conjectures. There are also no convincing proofs of the models' ability to explain the reality of the markets or the actuality of investors' behaviours. Thirdly, the classical models are prone to estimation errors or model risk of the inputs. Originally the historical means and variances of assets were used as the expected returns and the expected variance of assets. However, these values are poor estimation of the actual returns and variances especially in the short runs and medium runs. A small error of assets' expected returns usually have a huge effect on portfolio choices and optimality. A number of researchers coped with this problem by various approaches. Almost all were indirectly dealing with the estimation of inputs. They circumvented the problems by imposing more constraints into portfolio choices, using simulations or modifying the MOEA such that it can handle dynamic environments well. Some direct methods were using more accurate forecasting models or choosing the most pessimistic in order to be better than the forecasts in ex-post.

There is a gap for short-term portfolio optimisation in realistic settings. The estimation errors or forecasting models risk which are among the most serious problems to put the portfolio optimisation to be effectively used in practices should be handled systematically and effectively. There are no forecasting models or experts that can be accurate most of the time especially in predictions of the future. However, the expected errors can be managed in an efficient manner. Also if we can estimate the risk of various models and map their suitability for prevailing economic situations, we could mitigate the model risk by properly choosing the best, fittest model for the situations. This is the gap which this research attempts to reduce.

Chapter 4 Portfolio Optimisation Using Multi-Objective Genetic Algorithms

4.1. Chapter Overview

Different techniques have been used to solve portfolio optimisation problems. For those techniques for exact solution, they usually involve the number of combinations of states that must be searched which increases exponentially with the size of problem and becomes computationally intractable (Crama 2003). Furthermore, these techniques are inept at handling the nonlinear objective and constraint functions, and several assumptions are generally required to make the problem solvable using reasonable computational resources (Maringer 2005). Alternatively, some heuristic-based and evolutionary techniques can approximate solutions for problem instances of NP-hard problems in a reasonable time (Blum 2001). By using those techniques, the optimisation problems can be tackled in polynomial time with a trade-off in their optimality. In some circumstances of the practicality, the speed to reach the acceptable approximate solutions is very critical. Feasible near-optimum solutions are acceptable but untimely ones are not. Simple heuristics, based on greedy search algorithms, tend to stop in inferior local optima. In this chapter, we will use Multi-Objective Genetic Algorithm (MOGAs) to solve portfolio optimisation problem with typical realistic constraints. Although their solutions are approximate ones, they can be reached in acceptable time. We will also evaluate the quality of the solutions in the aspect of their closeness to the efficient portfolio frontiers. The best performing algorithms will be used in the later chapters as the main methodologies for the optimisation module. Even though the two objectives of portfolio optimization, i.e. minimizing portfolio volatility and maximizing portfolio return, can be collapsed into a single value (e.g. Sharpe ratio), this requires an assumption about utility function of the investor which defines the level of acceptable risk. In the case of using the Sharpe ratio, the investor's risk-return trading-off should be closed to linearity (Elton 1997).

4.2 Related work

The background and general working principles of the popular GA-based solution approaches for multi-objective optimisation are already discussed in details in section 3.7. In this section we will review their applications to portfolio optimisation problems.

Subbu et al (Subbu 2005) presented a new hybrid evolutionary multi-objective portfolio optimisation algorithm called Pareto Sorting Evolutionary Algorithm (PSEA) that integrated evolutionary computation with linear programming. Unlike Streichert el al (2004b), they applied the algorithm to a portfolio optimisation with more than 2 objectives. The problem can be stated as

$$Max_{w_i} BYld_p = \frac{\sum_i (BV_i \; x \; BYld_i)}{\sum_i BV_i} \quad (4.1)$$

$$Min_{wi} {\sigma_p}^2 \quad (4.2)$$

$$Min_{wi} \;\; VaR_p \quad (4.3)$$

$Subject\ to$

$$D_A - D_L \leq T1 \quad (4.4)$$

$$C_A - C_L \leq T2 \quad (4.5)$$

$$\boldsymbol{Ax} = \boldsymbol{a} \quad (4.6)$$

$$\boldsymbol{Bx} \leq \boldsymbol{b} \quad (4.7)$$

where, $BYld_p$ is book yield of the portfolio;

$BYld_i$ is book yield of asset i,

BV_i is book value of asset i,

VaR_p is Value at Risk of the portfolio,

D_A and D_L are durations of portfolio assets and of portfolio liabilities respectively,

C_A and C_L are convexities of portfolio assets and of portfolio liabilities respectively,

N is the number of assets in the portfolio,

x is an N by 1 vector of decision variables,

A is an M (any number) by N matrix,

B is an M by N matrix,

a is a vector.

Equations (4.4) and (4.7) represent sets of linear equality constraints and linear inequality constraints respectively. The authors propose a quite complicated algorithm to solve the problem. Firstly, they use apply Randomised Linear Programming to generate the portfolio population which satisfy the linear constraints. Secondly, the constrained population are preliminarily optimised in parallel by both of the Pareto Sorting Evolutionary Algorithm (PSEA) and the Target Objective Genetic

Algorithm (TOGA) (Subbu 2005). Thirdly, the efficient frontier portfolios selected by the both algorithms are filtered by the Fast Dominance Filter Algorithm. Fourthly, the results are visualised and the user sets targets. Fifthly, the selected portfolios are optimised by TOGA again. Lastly, the portfolio frontier is shown on the screen and the user may finally make a down selection.

The PSEA is good for small population size. Like all of meta-heuristic algorithms, there is no guarantee that a solution found at certain generation would be a globally solution.

The authors also use TOGA which is a non-Pareto and non-aggregating function approach to multi-objective optimisation as augmented optimizing algorithm. The algorithm borrows ideas from goal programming and Vector Evaluated Genetic Algorithm (VEGA). Contrary to the PSEA, TOGA is not based on the concept of dominance but is based on finding solutions of a predefined target for one or many criteria (Subbu 2005).

Hassan (2010) attempted to tackle the problem from a different angle. He recognised that SPEA2 was the best algorithm to solve portfolio problems so far and its main limitation lied in sub-optimality of out-of-sample results. He showed that the capital market was always changing thus the problems were becoming a multi-objective optimisation in dynamic environment. In other words, the suboptimal problem was not caused by estimation errors as such but by changing of the environment. He proposed a novel MOEA algorithm called R-SPEA2 to deal with the changing environment. He incorporated three strategies into the standard SPEA2 namely, ranked-based selection bias, diversity enhancement and mating restriction. In rank based selection bias, he added an extra routine after the Pareto front had been identified to calculate the robustness value based on the ranking of robustness which was defined as a solution's ranking preservation in the new environment. The robustness value was then added to the solution's original fitness value. In diversity enhancement, he implemented a couple of techniques to achieve. First, the pointed mutation was set to probability of 0.3 (30%) for entire evolutionary cycle. Second, the duplicate individuals were removed from the SPEA2 archive before selection so that the archive only contained unique genes. For mating restriction, he proposed a cluster-based mating restriction which prioritised mating between parents from the same cluster in order to preserve heterogeneity level of entire population. He also tested the proposed algorithms against the original SPEA2 with the Markowitz portfolio optimisation problem (no-short-sale). The data was drawn from 25 UK stocks listed in FTSE100 with 80 monthly observation sets. The time series were divided up to 60 months for training (48 months from May 1999 to April 2003 for in-sample training, and 12 months from May 2003 to April 2004 for validation) and the last 20 months from May 2004 to December 2005 for out-of-sample testing. All experiments had a population size of 500, archive size of 200 and repeated for 35 generations. He concluded from the result that R-SPEA2 had a statistically highly significant improvement in the mean number of cluster changes by individual solution between a training environment and a validation environment. Also with the diversity enhancement technique, the diversity of the population was indeed increased. He found that the cluster-based mating restriction combined provided the best robustness results with greatly enhancing the quality of solution and also increased the diversity of the solution population.

In this thesis (and also in the paper present to CEC 2007 [Skolpadungket 2007]), five GA based MOEA compared, namely VEGA, a fuzzy VEGA, MOGA, NSGA2, and SPEA2 using the dataset of Hong Kong's Hang Seng Index from the

OR-Library. The results of the experiment SPEA2 show that is the best with respects to GD metric and the distribution of solutions along the front (see also Chapter 3). Also, in a recent related work (Anagnostopoulos 2010), NSGA2, PESA and SPEA2 were compared in experimentation for three objectives, namely risk, return, and the number of securities in the portfolio. With three dimensions, the efficient frontier became a surface rather than a line. By visual comparisons, the authors found that SPEA2 was the best with respect to the hyper-volume metric as well as the diversity of the solution. However, it was the slowest with respect to computing time.

4.3 The Portfolio Optimisation Problem Model

The portfolio optimisation problem model can be applied to Multi-Objective Genetic Algorithm since the underlying objectives are to maximise portfolio expected return and minimised expected risk. In the original Markowitz model (see Section 2.1.1 and equations 2.1-2.6) the expected portfolio return is assumed to be based on assets' historical means, and the expected risk is assumed to be based on assets' historical variances and their past correction of assets' returns. So, the model is also called mean-variance analysis model. For Multi-Objective formulation, the problem can be represented as

$$Min_{\,w_i}\sigma_p^2 \ . \tag{4.8}$$

And

$$Max_{w_1} r_p \tag{4.9}$$

Subject to

$$\sum_i w_i = 1 \tag{4.10}$$

$$w_i \geq 0 \quad \forall\, i\,. \tag{4.11}$$

Where, σ^2_P is variance of the portfolio of assets,

r_P is the expected return of the portfolio,

w_i is the weights of asset i in the portfolio.

The Markowitz model is a simplified model to focus only on a theoretical point of view. In investment management, portfolio managers face a number of realistic constraints that arise from normal business practices, practical matters, and industry regulations. The realistic constraints that are of practical importance include (not exhaustively) round-lot constraints, cardinality constraints, floor constraints, turnover constraints, trading constraints, buy-in threshold, and transaction cost inclusions. In the meantime, only round-lot constraints, cardinality constraints, and buy-in (floor) constraints are considered.

Round-lot constraints require that the number of any asset included in the portfolio must be multiples of normal trading lot. The round-lot constraints can be expressed as:

$$w_i = \frac{n_i}{\sum_{i=1}^{N} n_i} \tag{4.12}$$

$$n_i \bmod l_i = 0 \quad \forall i. \tag{4.13}$$

Where n_i is number of unit of asset (share),

l_i is trading lot of the asset I and

N is the number of asset.

Cardinality constraints are the maximum number and minimum number of assets that a portfolio manager wishes to include in the portfolio due to monitoring reasons, or diversification reasons, or transaction cost control reasons (Stein 2005). The constraints can be expressed as follows:

$$C_i \leq \sum_{i=1}^{N} b_i \leq C_u. \tag{4.14}$$

Where, $b_1 = 1$ if $x_1 > 0$, otherwise, $b_1 = 0$,

C_l and C_u are the lowest number of assets and the highest number of assets required to include in a portfolio respectively.

Floor constraints define the lower limits on the proportion of each asset, which can be held in a portfolio. These constraints may result from institutional policy in order to diversify portfolio and to rule out negligible holding of assets for the ease of control (Crama 2003). They can be expressed mathematically as follows;

$$w_i \geq f_i \quad \forall i. \tag{4.15}$$

Where f_i is the lowest proportion and the highest proportion that asset *i* can be held in the portfolio.

4.4 The GA Algorithm Design

A Genetic algorithm is an evolutionary algorithm. The general main routine of GA can be found in Figure 3.1. GA's operation involves the exchanges of chromosomes which represent problem solutions. The good design of a GA is to balance between exploitation and exploration. The exploitation is to keep best solutions and improve them in order to be closer to the exact or ultimate solution. While the exploration is a mechanism to prevent the searches trapped in local optima (Blum 2003).

Genetic Algorithms are versatile search algorithms. They can be adjusted and modified to suit a wide range of problems and various degrees of performances. Design of a GA involves making choices mainly over its problem representation, its variation operator, its population size, its selection mechanism, its initialisation of the population and its fitness function. The key to designing a good GA lies in making the appropriate choices for these concerns with respect to the problem at hand (Michalewicz 2004). There are a number of successful applications of GA to portfolio optimisation which have been discussed in Section 3.6 and 3.7.

4.4.1 Problem Representation

The problem is represented by hybrid encoding (Streichert 2004b, 2005). A pair of genetic strings stands for a particular portfolio (an individual of population) as shown in Figure 4.1. The binary value string represents which stocks (or assets) are included in portfolio (0 stands for not included and 1 stands for included). The real value string represents weights of each stock in portfolio. So the lengths of both strings are equal to the number of stocks in the market (or the stocks of interest). Crossover and mutation operations are performed independently for both strings. But before evaluation both strings need to be combined so that the objective values can be calculated. Although the real value string can adequately represent the problem, the search space in this encoding is expanded and can be abruptly altered due to any changes in the binary value string.

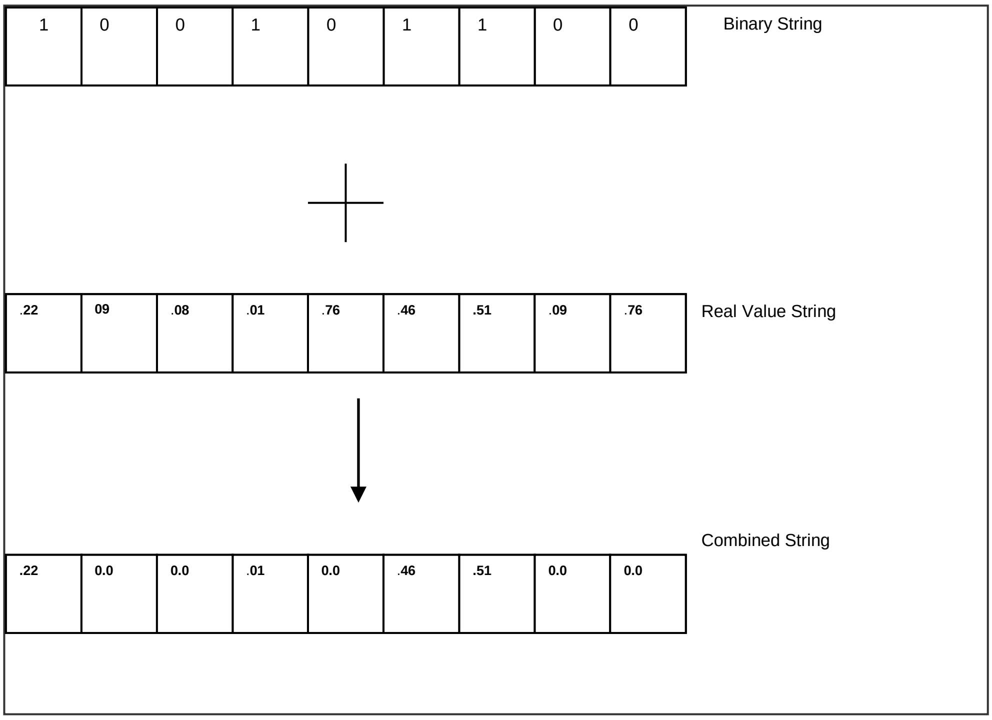


Figure 4.1: Problem Representation: Binary String, Real Value String and Combined String

4.4.2 Repair Algorithms and Constraint Handling

All constraints are handled through repair algorithms. The algorithms were proposed and used in Streichert (2004a, 2004b and 2005). The constraints in this setting are unity constraints (the sum of weights must be equal to one), cardinality constraints, floor (buy-in) constraints and round-lot constraints.

The repair algorithm first handles the cardinality constraints by setting N - K the smallest values of combined string to zero, where N is the number of selectable stocks (equal to the length of the strings) and K is the maximum number of stocks permitted in a portfolio (cardinality constraints). Then, it handles floor constraints (buy-in threshold) by setting stocks whose weights are below the buy-in threshold to zero. Next, it normalises those remaining non-zero weights to make all weights sum to 1 by setting $w_i^{/} = l_i + (w_i - l_i) / \Sigma(w_i - l_i)$, where w_i is non-zero weight of stock *i* and l_i is the buy-in threshold (the minimum weight amount that can be purchased) for stock *i*. Then, the round-lot constraints are handled by rounding the non-zero weights to the next round-lot level such that $w_i^{//} = w_i^{/} - (w_i^{/} \; mod \; c_i)$, where, c_i is the smallest volumes that can be normally purchased from the stock market for stock *i.* The remainder of the rounding process ($\Sigma w_i^{/} \; mod \; c_i$) is allocated in quantity of c_i to $w_i^{//}$ which has the biggest value of $w_i^{/} \; mod \; c_i$ until all of the remainder is depleted.

All pairs of strings are first filled with random numbers, so, they need to be repaired by the repair algorithm. Since crossover and mutation operations cause the string to be deformed, the repair algorithm need to be applied again to preserve the aforementioned constraints before the evaluations and selections.

4.5 Vector Evaluated Genetic Algorithm (VEGA)

The Vector Evaluated Genetic Algorithm is proposed by Schaffer (1985) as an extension of a simple genetic algorithm to handle multiple objectives in a single run. VEGA is a criterion-based fitness assignment, which is filling equal portions of mating pool according to the different objectives (Zitzler 2004). For *m* optimisation objectives and a fixed population size as *P*, VEGA randomly selects *m* subpopulations with the size of *P/m* each so there are *i = 1 to m* subpopulation. Each individual in subpopulation *i* will be evaluated based on the optimisation objective *i*. After probabilistic selections based on relative objective values (two for this experiment, i.e., relative yield and relative variance), the selected individual from each subpopulation is shuffled and pooled together to form a new population of size *P*. The new population is then followed by crossover and mutation operations. The whole process is repeated until the predetermined condition is met (in this case, Nth generation or round has been reached. VEGA usually has O (N) complexity for each round (generation) of population, where, N is the number of generation. Figure 4.2 shows the pseudo code of the VEGA algorithm.

Initialize generation counter: *N = 0*

Create a population, *Pop.*

Repeat while stopping criteria is not met.

  Initialize subpopulation counter: *i = 1.*

Repeat while *i<= m.*

    Generate *i*th subpopulation, *SubPop(i)*, by randomly selecting *P/m* individuals from *Pop*.

    Remove any individuals *SubPop(i)* from *Pop*.

    Generate the*i*th objective function values,**F**(*i*), for individuals in *SubPop(i).*

    Perform genetic selection on *SubPop(i)* based on **F***(i).*

i = i + 1.

End Repeat

Pop = Integrate all *SubPop(i)*.

  Shuffle the individual sequence in *Pop*.

N = N + 1.

End Repeat

Evaluate *Pop* for all objective function values for all **F**(i).

Return (*Pop*, all **F**(i),…)

Figure 4.2: VEGA Main Routine

4.6 Fuzzy VEGA

VEGA tends to converge towards one-objective best solution, thus it is quite incompatible with multi-objective optimisation in which trading off between objectives is required during the search process. Introducing fuzzy logic into VEGA may facilitate the trade-off between objectives. We incorporate a fuzzy decision rule to combine optimisation objectives together. The fuzzy decision rule dictates the probability of selection for each individual. Fuzzy logic theory has been briefly discussed in Section 3.8 of Chapter 3.

Y | Var	**Min**	**Very Lo**	**Low**	**Moderate**	**High**	**Very High**
Max	Certain	Highly Likely	Highly Likely	Likely	Likely	Probably
Very High	Highly Likely	Highly Likely	Likely	Likely	Probably	Probably
High	Highly Likely	Likely	Likely	Likely	Probably	Probably
Moderate	Likely	Likely	Likely	Probably	Unlikely	Highly Unlikely
Low	Likely	Probably	Probably	Unlikely	Unlikely	Highly Unlikely
Very Low	Probably	Probably	Probably	Unlikely	Highly Unlikely	Never

TABLE 4.1 Fuzzy Rules for Fuzzy VEGA (VEGA_Fuz1)

We design two versions of VEGA with fuzzy logic. The first version of fuzzy VEGA (VEGA_Fuz1) is modified form ordinary VEGA using fuzzy combination of both objectives (yield and variance) to determine the probability of selection of particular individuals (see Table 4.1.) for both of subpopulations which are, in the original VEGA, the sub-populations selected by only single objectives (one for yield and another for variance). In table 4.1, Certain, Highly Likely, Likely, Probably, Unlikely, Highly Unlikely and Never represent probabilities of selection. By their linguistic meanings, we set the values of probabilities to 1.0, 0.9, 0.75, 0.5, 0.35, 0.1 and 0.0 accordingly. The second version of fuzzy VEGA (VEGA_Fuz2) is different from VEGA_Fuz1 by combining an additional objective, i.e. sampling distance by randomly selecting 25 different individuals from populations (using sampling instead of plain distance to reduce the complexity of the algorithm) into fuzzy rule. Then, the fuzzy rule has 3 objectives instead of 2. The fuzzy rule prefers more sampling distance to less to correct clustering problems of both VEGA and VEGA_Fuz1. Figure 4.3 demonstrates Fuzzy VEGA main routine in which fuzzy logic is used to combine the two objectives together with fuzzy rules.

Initialize generation counter: *N = 0*

Create a population, *Pop.*

Repeat while stopping criteria is not met.

Repeat while *i*<= *m.*

Generate the*i*th objective function values,**F**(*i*), for individuals in *Pop.*

End Repeat

Fuzzify of the objective function values.

Apply Fuzzy rules to the Fuzzified objectives giving Fuzzy value.

Defuzzify the Fuzzy value to get selecting probability *p.*

Perform genetic selection on *Pop* by probability *p.*

N = N + 1.

End Repeat

Evaluate *Pop* for all objective function values for all **F**(i).

Return (*Pop*, all **F**(i),…)

Figure 4.3: Fuzzy VEGA Main Routine

4.7 MOGA

MOGA in this paper stands for the Multi-Objective Genetic Algorithm proposed by Fonseca and Flemming in 1993 (Fonceca 1993). MOGA uses Pareto rankings to assign the smallest ranking value to all non- dominated individuals. For dominated individuals, they are ranked by how many individuals in the population dominate them. Thus, the raw fitness of an individual is an inverse function of its Pareto rank. In order to distribute the individual in the population evenly along the Pareto front, the overall fitness function is then adjusted by sum of sharing distance. The sharing distance between individuals i and j is given by

$$SF(i,j) = 1 - \left[\frac{d(x_i, x_j)}{\sigma_{share}}\right], \;\; if\ d(x_i, x_j) < \sigma_{share}\,, \;\; else\ \ SF(i,j) = 0 \qquad (4.16)$$

Where, $d(X_i, X_j)$is a metric distance between two individuals in objective domain,

σ_{share} is a predefined sharing distance.

Thus, the overall fitness is defined by

$$Sharing\ fit\ (i) = \frac{fit(i)}{\sum_j SF(i,j)} \qquad (4.17)$$

Where, fit(i) is the inverse of Pareto rank(i) (1/rank(i) in this test).

The overall fitness values of individuals are to be used in the probabilistic selection process by the comparative overall fitness to the individual that has maximum overall fitness. The comparative fitness values are used to compare with a random number. If they exceed the random number, the individual will be selected (roulette selection method). MOGA usually has O (N^2) for a single round, because it needs to compute Pareto ranks and the sharing distance for all individuals.

4.8 Strength Pareto Evolutionary Algorithm 2 (SPEA2)

SPEA2 was proposed by Zitzler et al (2001) as an improvement of the original SPEA. Like SPEA, SPEA2 uses two populations of size P, the first (P) for the population and the second (P’) for the archive. In SPEA, all non-dominated individuals in population P are copied to the archive P’, so the size of the archive P’ is varied from generation to generation. However, in SPEA2, the size of P’ is fixed thus if the number of non-dominated individuals in a generation exceeds the size of P’, the number will be truncated. On the other hand, if the number is less than P’, some dominated individuals need to be added in the archive P’. The truncation and addition of dominated individuals are incorporate density information as a strategy to make the solutions distribute along the Pareto front. The density estimation of an individual i is defined as $D(i) = 1/(d_i+2)$, where d_i is the distance of individual i from the nearest neighbor.

SPEA2 first selects all non-dominated individuals from the population P in the first round and then selects the combined population of P and archive of P’ in the subsequent rounds. Unlike VEGAs and MOGA, the selection is deterministic rather than probabilistic. If the number of non-dominated individuals exceeds the fixed size of the archive P’, the excess individuals will be unselected based on the density estimation. If the number of non-dominated individuals falls short of the size of P’, then the remaining dominated individuals (the next best Pareto front) will be selected until the archive has been filled. (However, if the last selected Pareto front exceeds the size, the same truncation method will be employed.) SPEA2 usually has O (N^2 log N) a single round, due to the density estimation calculation (Zitzler 2004.)

4.9 Experimentation and Results

As our aim in this chapter is to test the performances of a number of MOEAs and comparing them with the benchmark portfolios which are usually referred to by the other researchers in the field, we decide to use a different data set from the rest of this thesis. In this way we will be able to compare the results and reference in future works of portfolio optimisation research. We conducted a number of experiments on data from OR library maintained by Prof. Beasley as a public benchmark data set (derived from Heng Seng data set with 31 stocks.) The data can be found at http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html.

The experiment runs on 2 problem instances; the first case with cardinality constraints where the number of stocks in a portfolio is not being larger than 10 (K = 10) and the second case with cardinality constraints where the number of stocks in a portfolio is not being larger than 5 (K = 5). In each case, the data is run by using VEGA, VEGA_Fuz1, VEGA_Fuz2 and MOGA in which the run lasts for 500, 1000 and 5000 rounds (generations) with the number of population at 400. The performance is assessed through the average of 10 times of tests. For SPEA2, the algorithm runs only for 100, 250 and 500 generations. The performance is measured by Generation Distance (GD) (Tan 2005) for the last generation of the tests. GD is given as follow;

$$GD = \sqrt{\frac{1}{M}\sum_{i=1}^{M} d_i^2}. \qquad (4.18)$$

Where, d_i is the nearest distance between Pareto front of the results (PF known) and Pareto front of the benchmark solutions provided by the OR-library (PF true),

M is the number of solution population.

The graphical results of known Pareto fronts for the algorithms (N = 5000) is shown in Figure 4.4. MOGA seems the best when compared to the true Pareto front (NC Eff Front) both in the terms of the closeness and distribution along the true front. When one looks at SPEA2, they perform much better, even when N is only 100, 250 and 500 rounds.

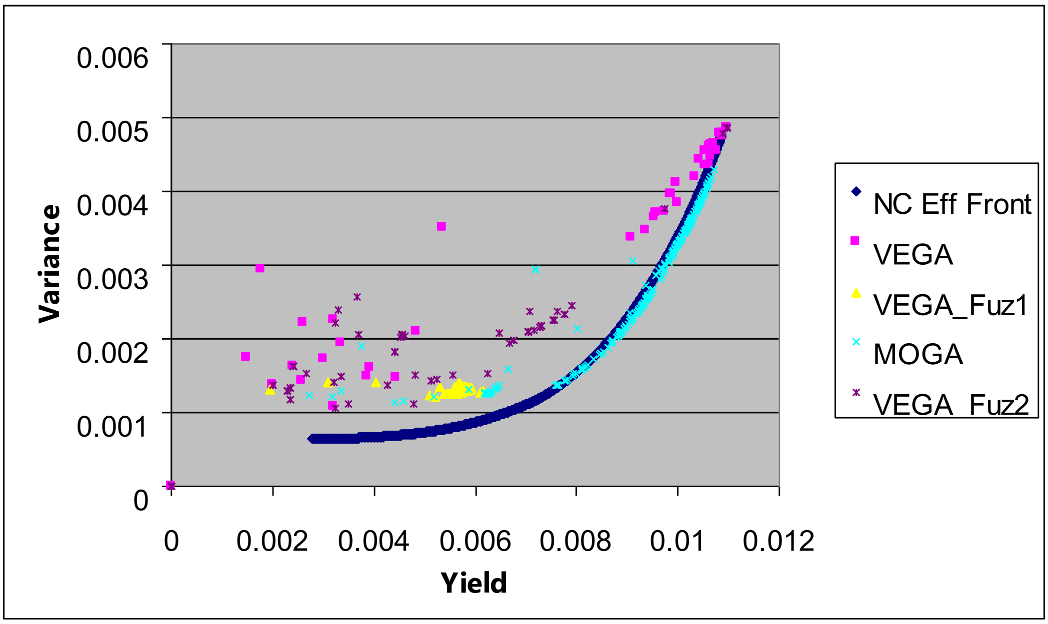


Figure 4.4: The Results for N= 5000

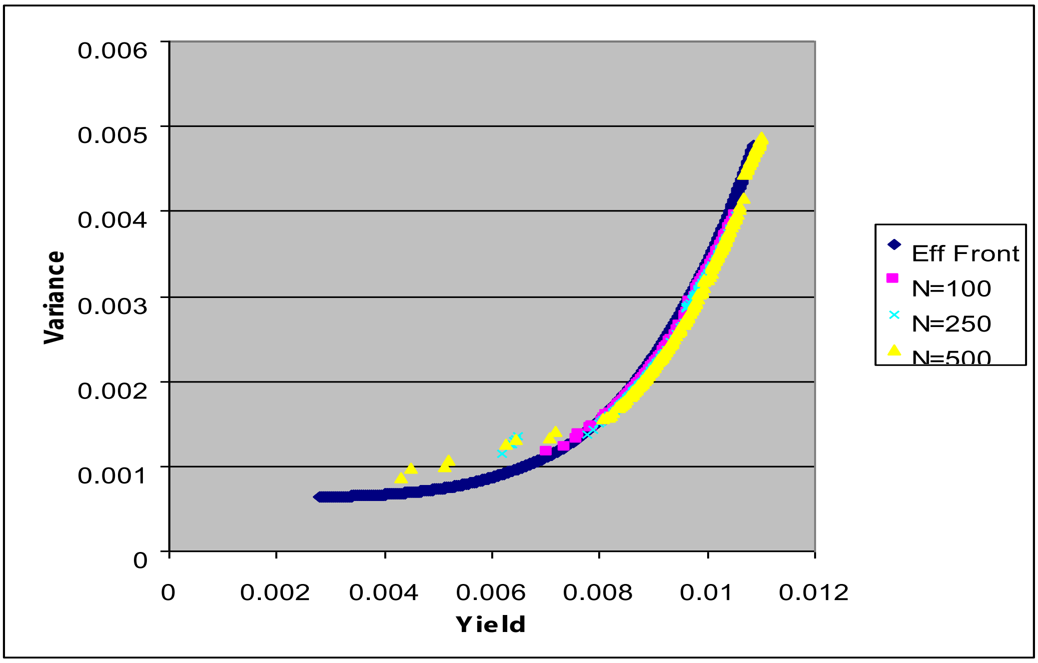
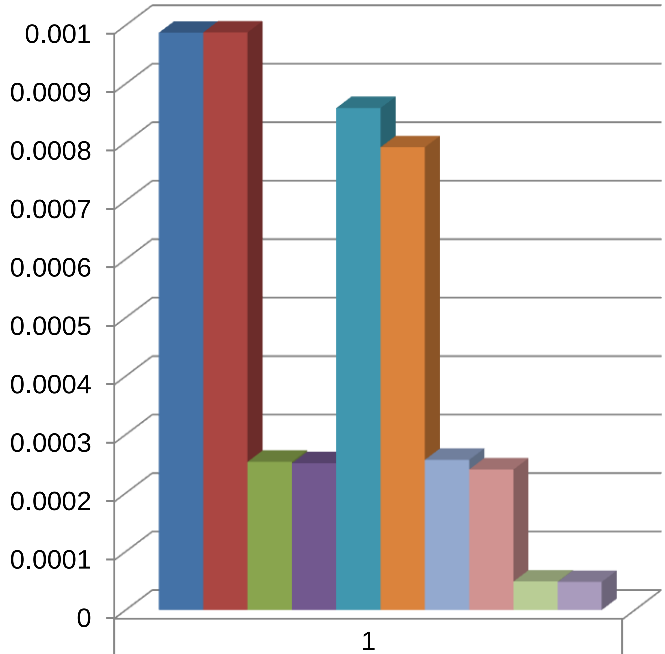


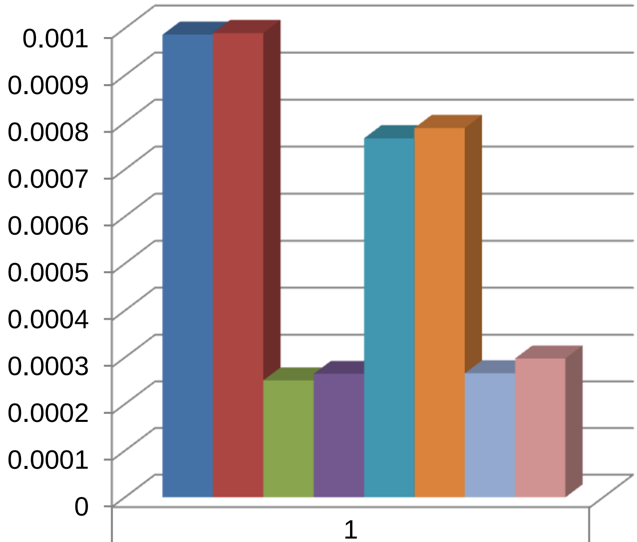
Figure 4.5: SPEA2 with N=100, 250 and 500



	1
VEGA K =10	0.000987345
VEGA K = 5	0.00098776
VEGA_Fuz1 K=10	0.00025351
VEGA_Fuz1 K=5	0.0002511
VEGA_Fuz2 K=10	0.000858425
VEGA_Fuz2 K=5	0.000791422
MOGA K=10	0.000256879
MOGA K=5	0.000240365
SPEA2 K=10	4.8902E-05
SPEA2 K=5	4.80922E-05

Figure 4.6: Generational Distance (GD) for N = 500

Comparing performance by using Generational Distances (GD), one finds that for N= 500, SPEA2 performs the best for both of instances (K = 5 and K =10). MOGA and VEGA_Fuz 1 perform about the same. VEGA_Fuz2 is the second worst and VEGA is the worst. However, Figure 4.1 shows that VEGA_Fuz 1 does not evenly distribute while VEGA_Fuz 2 improves the distributions but has to trade off performance. The results indicate that, as VEGA is not Pareto based MOEAs, we can improve some aspects of its performance by using Fuzzy logic to combine the objective selections but its performance is still unmatched to the performances of those Pareto based MOEAs.



	1
VEGA K=10	0.000986705
VEGA K=5	0.000990245
VEGA_Fuz1 K=10	0.000251035
VEGA_Fuz1 K=5	0.000265086
VEGA_Fuz2 K=10	0.000766337
VEGA_Fuz2 K=5	0.000788418
MOGA K=10	0.000266229
MOGA K=5	0.000297958

Figure 4.7: Generational Distance (GD) for N = 5000

Figure 4.7 shows the GDs for N= 5000, the relative performances for the algorithms are not changed for those of N= 5000 (no SPEA2). We can see that the results GDs are not much different from those of N= 500. It seems that all of the algorithms are rapidly converged. Interestingly, in these cases, those with K= 10 (with 10 assets) on average perform better than those of K = 5. This is contrary to the cases of N = 500. This might be due to the expanded search space for K = 10, which mitigates the possibility of being trapped in local optima. Even so, the performances are only slightly better than those of the counter parts. Also those of VEGA_Fuz1 show superior performance in both cases (K=5 and K=10) to those of VEGA and VEGA_Fuz2. However, in these cases of N=5000, those of VEGA_Fuz1 are even slightly better than those of MOGA. The VEGA_Fuz1's drawback is the solutions tend to concentrate in a certain area behind the efficient frontier, not distributing along it.

4.10 Chapter Summary

In this chapter, various techniques of Multi-Objective Genetic Algorithms are applied to solve portfolio optimisation with some realistic constraints, namely cardinality constraints, floor constraints, and round-lot constraints. We apply fuzzy logic to see whether it can improve performances of the Vector Evaluated Genetic Algorithm (VEGA). The results show that using fuzzy logic to combine optimisation objectives of VEGA (in VEGA_Fuz1) for this problem does improve performances especially in Generation Distance from the true Pareto front, but its solutions tend to cluster around a few points. With additional fuzzy logic to make VEGA solution more distributed, it causes the performance to worsen. MOGA and SPEA2 are more complex algorithms but they perform better. SPEA2 perform the best even in comparatively small N and also has a good distribution along the Pareto front.

Chapter 5 Forecasting Stock Returns and Volatilities Using Evolutionary Artificial Neural Networks

5.1 Chapter overview

This chapter presents ANN and Genetic Algorithms to build stock return forecasting frameworks, namely Evolutionary Artificial Neural Network (EANN). We propose an evolutionary scheme of neural networks with evolving connection weights and "step-up," adding more hidden nodes and layers in order to search for optimal structure. The adaptive EANNs are used to predict individual stock returns based on multivariate time series (AR with state variables) models. Since the input time series are quite limited, Multi-fold Cross Validation methods for the sections of the optimal ANN structures are used to circumvent the data problem. The prediction results have been compared with those of simple regressions (Least Square Estimation) and of simple (non-evolutionary) ANNs (Back Propagation and Elman Recurrent ANNs). The proposed EANNs will be used as forecasting models among the set of selectable forecasting models in the next Chapter.

5.2 Related Works

For stock market investors and portfolio managers, it is crucial to have the most accurate forecast of stock returns. However, stock returns are difficult to forecast accurately. Several models and techniques have been used to forecast stock returns. The relevant models include autoregressive models (AR), autoregressive-moving-average models (ARMA) and AR with state variable (explaining variable) models. It is reported that the AR models with state variables are superior to the rest both in the short-run and the long-run. In developing a model with many state variables as model inputs to forecast stock returns, a crucial part is to identify input variables. There are numerous theories and models that determine the input variables ranging from technical analysis based on trading data to complicated multivariate time series models. The multivariate time series models, which are based on fundamental factors, are considered more theoretically sound than those based on technical factors (e.g. trading volume, price trend, etc.) (Zhou 1996). They imply that stock prices and stock returns can be explained and thus predicted by a number of "fundamental" economic factors as proposed by Capital Market Theory (CAPM-single factor, i.e. stock market index) (Sharp 1964, Lintner 1965) and Asset Pricing Theory (APT – multi-factors) (Ross 1967). The inputs suggested by an empirical research were changing in economic and financial variables, such as changing in inflation, changing in yield spreads, etc. (Roll 1980).

The techniques that have been deployed to forecast stock returns are linear regression (time-series), artificial neural networks (ANNs), decision trees, rule induction, Bayesian belief networks, evolutionary algorithms (EAs), classifier systems and association rules (Kwon 2005). Researchers found that ANNs show better performance than most other techniques, especially linear regressions (Kwon 2007). However, ANNs with sub-optimal initial weights can be trapped in local minima. In dynamic environments, as the nature of learning objects are always changing, the topologies of the ANNs also should be adapted accordingly. Evolutionary Algorithms can be applied to evolve ANNs at many levels, e.g. connection weights, topologies, both the number of hidden nodes as well as the number of hidden layers, and learning rules (Yao 1999). Some previous research has applied Evolutionary Artificial Neural Networks (EANNs) to predict stocks prices (Kwon 2005, 2007). In their research, Kwon et al. (Kwon 2005) aimed to predict stock price trends (only up or down) by using EANNs that could evolve their initial connection weights. Comparing buy-and-hold strategy, Recurrent ANNs (RANNs) and EANNs, they found that their proposed EANNs outperformed Recurrent ANNs, and also EANNs were significantly outperformed buy-and-hold strategies. The same author in another paper (Kwon 2007), attempted to predict stock returns based on stock correlations (with other stocks in the same market). They proposed EANNs (called Feature Selection Genetic Algorithm –FSGA) that could evolve a set of inputs (selections of inputs). By comparing prediction performance (only stock price up or down) of buy-and-hold, Recurrent ANNs and EANNs (FSGAs), they found that the order of performance was the same, i.e. EANNs then RANNs and then buy-and-hold-strtegies. A related research by Armano et al. (Armano 2005) proposed an EANN algorithm called NXCS, essentially a set of genetic classifiers designed to control feed forward ANNs' activation for performing forecasting at different particular local scopes, to predict stock indexes (rather than individual stock prices or returns also up or down only.) The prediction then results from experts' interactions in the population. The research found that the proposed methodology repeatedly outperformed buy-and-hold strategy.

Most previous research has been concentrated on predicting stock indexes or trends rather than individual stock returns (Zhou 1996, Yao 1999). For stock trading, merely prediction on trends of stock prices or indexes would be adequate. But for portfolio optimisation, especially mean-variances analysis (Markowitz) portfolio optimisation model, to construct an efficient portfolio of stocks, a portfolio manager needs to most accurately predict individual stock returns as well as their variances (Oberuc 2004).

5.3 Stock Forecasting Models

Generally, stock returns are the difference in prices from the beginning period (investing time) to the ending period (disinvesting time) plus dividends if any. For convenience and by assuming that either the time is quite short or dividend payouts are always reflected in asset prices, dividends will be disregarded in the models. Models of asset return can be generalised into 3 categories of models. In the simplest AR Model for time-variation expected returns, the expected returns follow auto-regressive (AR) processes. The second category is called the ARMA model. The logarithmic prices of assets have two components, a permanent part and a transitory part. The permanent part follows an AR process. On the other hand, the transitory part follows a moving average (MA) process. The last category is the state variable model. In this kind of model, the transitory component of price not only depends on its own past value but also on state variables (x) which are relevant financial and economic variables. The AR Model of expected returns has considerable capacity to capture the movement of stock returns over short horizons but has a mediocre capacity to predict the expected stock return over longer horizons. On the other hand, the ARMA model does a good job of forecasting long-horizon returns, but has no adequate flexibility to capture the pattern of expected return at short horizons. Meanwhile the State Variable model is the best in at least four aspects, namely using only the recent past returns, parsimonious, good prediction in the short-run and good prediction in the long-run (Zhou 1996).

A State Variable model with K state variables and N time lags can be stated as

$$r_t \quad = \quad \sum_{i=1}^{N} \phi_i \, r_{t-i} + \sum_{k}^{K} \sum_{i}^{N} \gamma_i^k \Delta x_{t-i}^k + e_t . \qquad (5.1)$$

Where,

r_t, , r_{t-i} is stock returns at the time period t and t-i respectively.

ϕ_i is stock return's autoregressive coefficient for time lag i.

x_{t-i}^k is the k[th] state variable at previous i[th] time period,

γ_ι^k is the regression coefficient for the previous i[th] period of the k[th] state variable,

e_τ is the error term.

Factors that have evidences of influencing stock returns and are included in this model are previous stock returns (R), unemployment (U), money supply (M), stock index (SP500), inflation (CPI), default spread (DS), term spread (TS), reference interest rate (FED), industrial production (IP), and January effect (JAN-circumstantial variable) (Oberuc 2004). Some factors which have no monthly data have been excluded (e.g. trade deficits, GDP etc). The model can be stated as follows

$$R(t=0) = R(t=-1 \text{ to } -12) + U(t=-1 \text{ to } -12) + M(t=-1 \text{ to } -12) + SP500(t=-1 \text{ to } -12) + CPI(t=-1 \text{ to } -12) + DS(t=-1 \text{ to } -12) + TS(t=-1 \text{ to } -12) + FED(t=-1 \text{ to } -12) + IP(t=-1 \text{ to } -12) + JAN(t=-1 \text{ to } -12) \quad (5.2)$$

The model in 5.2 constitutes a prediction output and 120 inputs (10 kinds with 12 month lags each.) For January effect (JAN), the input is “1” if the month is January and “0” otherwise. It can be estimated by Ordinary Least Square or Linear Square Regression Techniques (Rachev 2007) as well as making it nonlinear through Artificial Neural Networks.

5.4 Back Propagation and Elman ANNs

An Artificial Neural Networks is a crude approximation of biological neural networks. A neural network is a fundamental unit of human and animal brains. A brain is merely an extremely large collection of connected neurons. Neurons communicate and pass signals through synapses where, in biological neurons, a signal is created by chemical interactions. A neuron typically receives signals through synapses on dendrites and sends signals through synapses on axons. An artificial neuron is a simplified version of the biological neurons. There are three basic elements of the artificial neuron. First, connecting links or synapses, each weighted by positive or negative values. Second, an adder is to sum up the input signal from the weighted synapses. Third, an activation function or a squashing function is to limit the permissible amplitude range of the output signal to a range of finite value. Figure 5.1 shows a nonlinear model of a neuron.

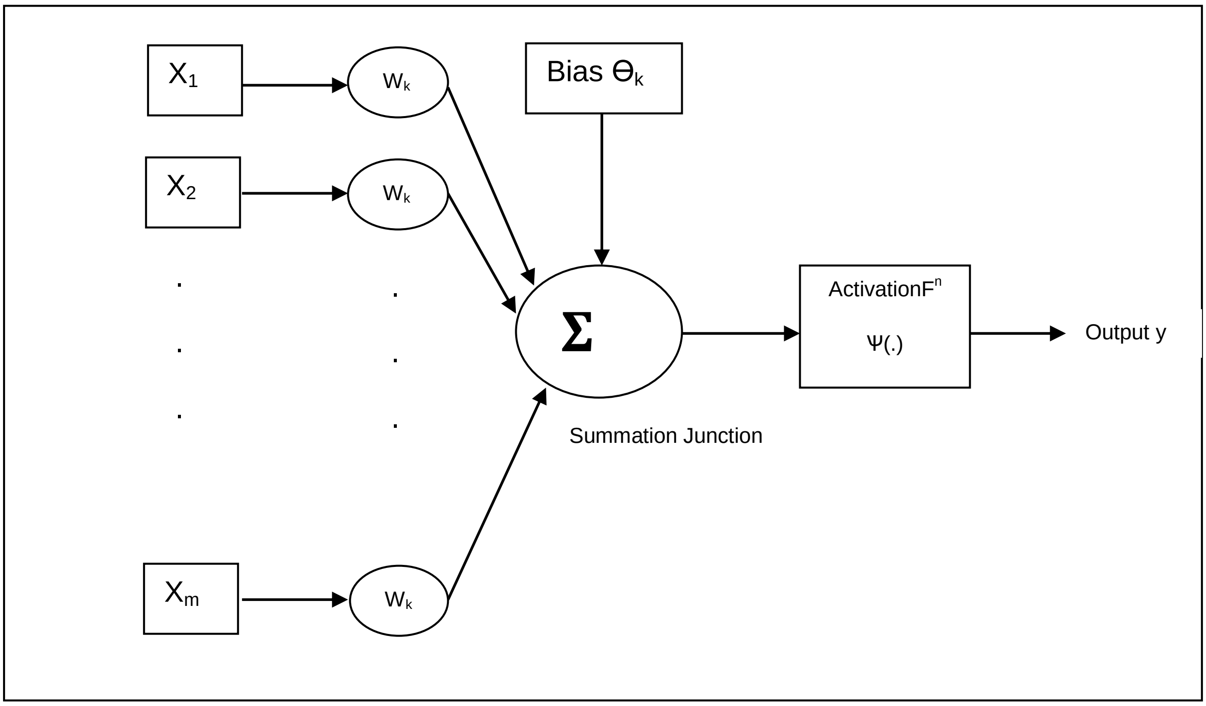


Figure 5.1: A nonlinear model of a neuron

In this chapter, two kinds of Artificial Neuron Networks, namely, Back Propagation Networks and Elman Recurrent Networks have been used in experiment to forecast stock returns. In a Back Propagation Network, there are three or more layers with an input layer, an output layer and one or more hidden layers. Its output is a non-linear function of activation, usually a sigmoid function which can be stated mathematically.

$$y = \frac{1}{1+ e^{-u}} \;; \qquad u = \sum_j w_j \, x_j + \theta \tag{5.3}$$

Where, x_j is j^{th} input,

w_j is weight associated with input $x_{j,}$

y is the output and

Ө is the bias.

The training algorithm is to adjust the weights as to minimise the error e with p sets or periods of data, which is given as,

$$e = \sum_p \sum_j (t_j - y_j)^2 \tag{5.4}$$

Where, y_j is the j^{th} output, and

t_j is the actual value.

For the output layer, the weight updates would be:

$$\Delta w_{jk} = \eta . \sum_p (t_k - y_k). y_k . (1 - y_k). y_j \tag{5.5}$$

Where, η is the learning constant.

Δw_{jk} is the change in weights from node k in the adjacent layer to the output layer,

For the hidden layers, the weight updates would be:

$$\Delta w_{ji} = \eta . \sum_k \sum_p (t_k - y_k) . y_k . (1 - y_k) . w_{kj} . y_j . \left(1 - y_j\right) . y_i \qquad (5.6)$$

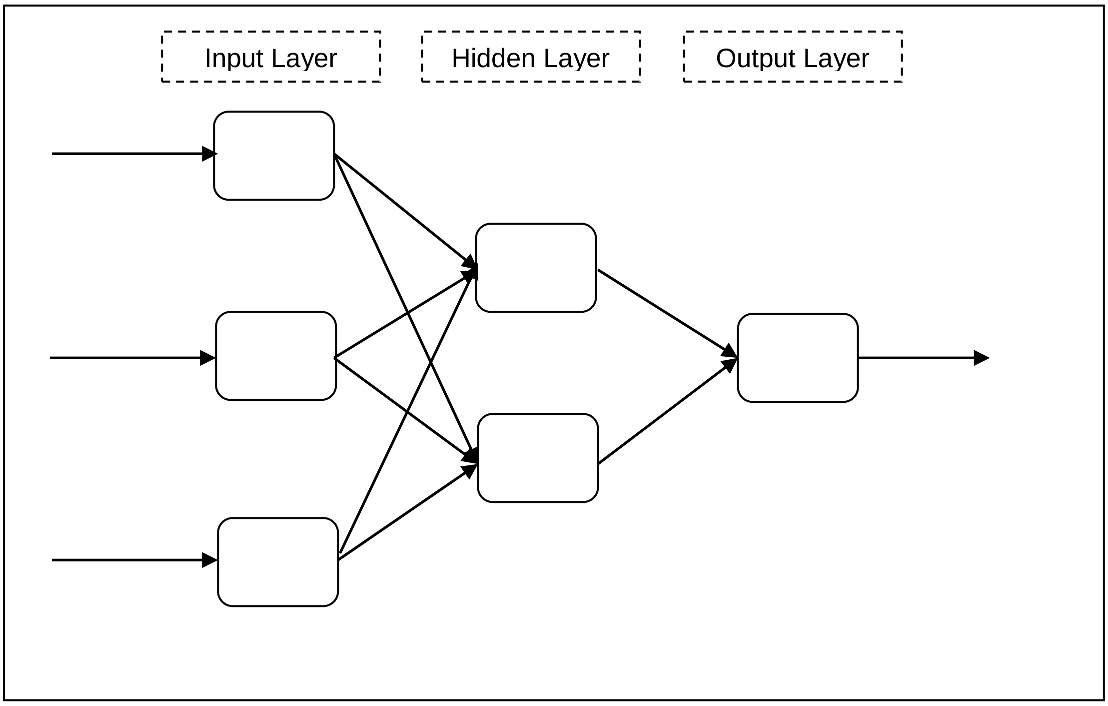


Figure 5.2: Graphical Model of a Typical Back Propagation Neuron Network

Figure 5.2 depicts a graphical model of Back Propagation Network (BPN). An Elman Recurrent Network includes a number of context units to accept signals from the hidden layer and feed them back as inputs to the hidden layer. Figure 5.3 shows the graphical model of an Elman Recurrent Network. BPNs and Elman ANNs are used in the next section to predict stock return for next T period ahead and modified to be evolved by encoded GAs in the next section.

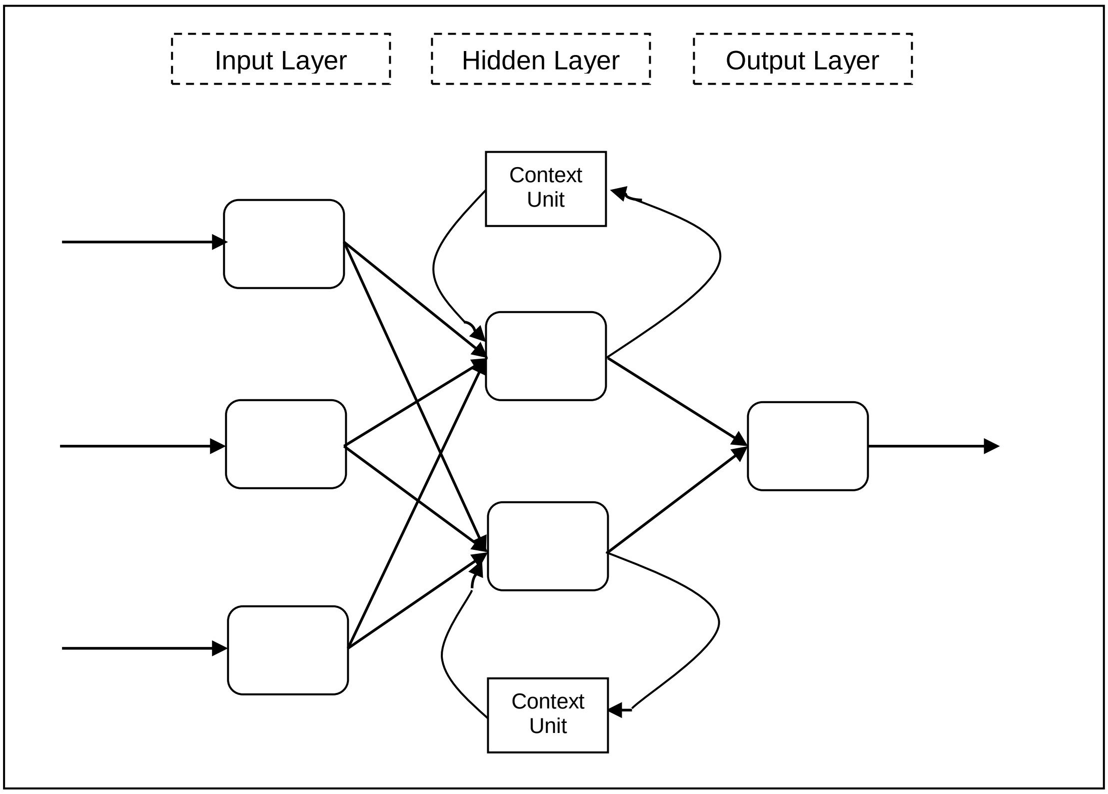


Figure 5.3: Graphical Model of a Typical Elman Recurrent Neuron Network

5.5 Evolutionary ANN Design

5.5.1 The ANN and EA Encoding

The proposed encodings for the BPNs and RANs for evolving proposes use direct encoding for connection weights and indirect encoding for the number of hidden nodes and the number of layers. There are two set of evolving encoding genes, namely weight matrices and layer specification. A weight matrix describes weights of connections for each node (hidden and output nodes) to other hidden nodes in the immediate previous layer (may be the input layer for hidden nodes in the first hidden layer). The weight values are between -1 and 1. In the first generation the weights are randomly set up. Also when the structure of an ANN is changing, i.e. adding a hidden node or adding a hidden layer, the weights are also randomly re-assigned. In evolutionary process, each weight mutates to its current value plus a random number between -0.5 and 0.5. If the mutated value exceeds 1then the weight will be set to 1. If the mutated values is below -1 then the weight will be set to -1. A layer specification is a vector of integer describing the number of hidden nodes in each layer excluding input layers, thus the length of layer specification is equal to the number of hidden node plus one of the output layers. In evolutionary process, after a mutation on connection weights cannot improve the performance of an ANN, a hidden node is added into the first hidden layer with all connection weights randomly reassigned. If the inclusion of a hidden node in the first hidden layer also cannot improve the performance of the ANN, a new hidden layer with 2 hidden nodes is put before the first hidden layer (then become the first hidden layer.)

5.5.2 Evolutionary Algorithm

The proposed Evolutionary Algorithm used in this chapter is a modified EP Net Algorithm from (Sharp 1964). The proposed EA is named Step-up Mutation Evolutionary ANN. As the name suggests, the algorithm begins with a mutation that has least effect on the structure of ANNs then “step-up” to have greater effect if the previous mutation fails to improve the performance of the ANNs. On the other hand, if the mutation can improve the performance, the new network will be selected while its parent will be discarded (dual tournament with its parent) and the loop will continue to the next iteration. The main loop is shown in Figure 5.4.

.

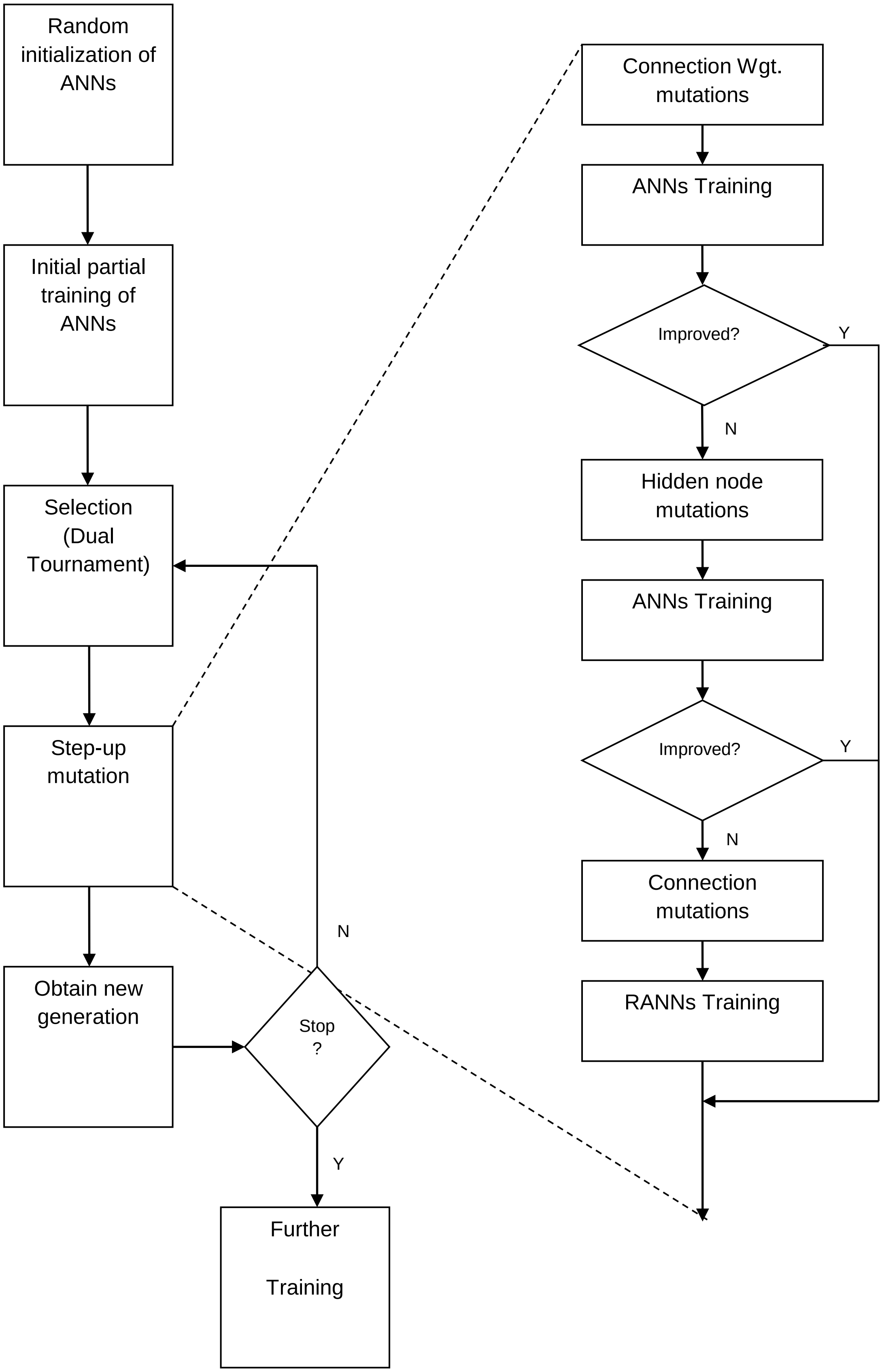


Figure 5.4: The Step Wise Mutation of EANN Algorithm

The algorithm begins with random initialisation of a set of ANN encoding genetic boxes and then the creation of the corresponding ANNs. All of the ANNs are initially trained to collect their preliminary fitness values. The selected ANNs then go under the “step-up” mutation with 3 conditional sub-steps, namely, connection weight mutations, hidden node mutations and connection mutations. The mutations are conditionally stepped-up in such a way that it will step at a sub-step if the trained corresponding ANN’s fitness value (in this case, MCV value as described in the next section) is improved. The process repeats until the pre-specified round count is met. The algorithm does not have a mating operator but bases the evolution solely on the mutation operators.

5.5.3 The EA Objective

Multifold Cross-Validation (MCV) is a method that makes efficient use of the available data. It is a sample re-used method to estimate prediction risk. The EA objective is to minimise prediction risk of the EANN. MCV is essentially a perturbation refinement of Cross-Validation (CV) methods. The method can be described as follows:

Let the data set *D* be divided into m randomly chosen disjoint subsets D_j of roughly equal size.

$$\bigcup_{j=1}^{m} D_j \quad = \quad D, \qquad D_i \cap D_j = \phi \quad for\,\forall i \neq j \tag{5.7}$$

For each disjoint set j, CV is defined as

$$CV_{D_j}(\lambda) \quad = \quad \frac{1}{N_j} \sum_{(x_k,\, t_k) \in D_j} (t_k - \hat{\mu}_\lambda(D_j, x_k))^2 \qquad (5.8)$$

Where,

$\mu_\lambda(D_j , x_k)$ is an estimator trained on all data except $(x, t) \varepsilon D_j$,

t_k is the realized (actual) output,

x_k is the vector of all inputs and

N_j is the number of observations in the subset D_j.

Cross-Validation (CV) for all available data set of an ANN is a non-parametric estimation of the prediction risk.

$$CV(\lambda) \quad = \quad \frac{1}{m} \sum_j CV_{D_j}(\lambda) \qquad (5.9)$$

A refinement is required for CV to become MCV. An ANN is trained on the entire set of data D to obtain estimates $\mu_\lambda(D, x_k)$ with set of weights W_0. The weights W_0 are used as starting point for m-fold cross validation production procedure. Each subset D_j is removed from the training data in turn. The ANN is then retrained using the remaining data (starting at W_0, not random initial weights) assuming that deleting a subset from training data set does not lead to a significant difference in the locally-optima weights. These perturbed retraining from W_0 yields W_i (i = 1 to m.) The MCV error is calculated for each "perturbed model" by the sum $(t_k - \mu_\lambda (D_j, x_k))$ as an estimation of prediction risk of the model with W_0.

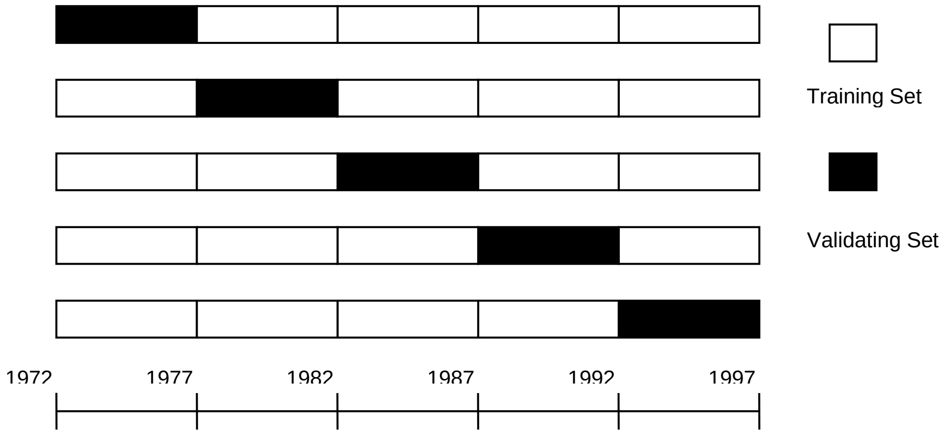


Figure 5.5: Multi-fold Cross Validation for Selection of Optimal ANN Structure

5.6 Experimentation

The forecast ANNs and the experiments were conducted with the Step-up Mutation EANN Algorithm proposed above with BPNs and Elman RANs without evolutions as well as forecasting from Linear Least Square Regression (LS) in order to compare their performances. All forecasting have been trained and tested with monthly dividend and split adjusted return series from 1971 to 2007 on 10 selected stocks in US Stock markets, namely Alcoa (AA), Boeing (BA), Caterpillar (CAT), Dupont (DD), Disney (DIS), General Electric (GE), General Motor (GM), Honeywell (HON), HP (HPQ) and IBM (IBM). The independent variables are 12 month time lag (from lag = -1 to -12) of changing on consumer price index (CPI), default yield spread, term yield spread, fed fund rate, industrial product, money supply (M), S&P 500, unemployment rate, January effect (dummy variable) also stock returns own lags. These variables are selected based on some previous research related to economic factors that have or should have effects on stock prices (Roll 1980 and Oberuc 2004). The variables believe to have direct or indirect effects on stock prices can be described as follows. Change in consumer price index is a measure of inflation and inflation usually causes asset price to rise. Default yield spread, which is a spread between an A rating bond and a BBB rating bond, is observed to be widen with economic downturn and stock prices falling. Term yield spread, which is the spread between a long-term bond (usually 15 year remaining) and a short-term bond (1 year remaining), is observed to be widen when the economy prospers, and narrowed or even negative when the economy takes a downturn. Fed fund rate is usually set high in an economic boom and low when the economy is in recession. Industrial product tends to correlate with the level of economic activity and stock market cycle. Money supply also shows some correlation with economic and stock market cycles. S&P 500 is the stock index itself and according to CAPM, directly affects stock prices. In contrast, unemployment rate usually has negative correlation with the economic and stock market cycles. Lastly, it has been noticed that stock markets are usually driven high at the end of long Christmas and New Year holidays. This might be because some investors return to the markets after long holidays. Therefore, we add this variable as a dummy variable. The sets of data are paired between a dependent variable and a set of time lags independent variables (10 nominal variables with 12 lags, in totality 120 including lags) to form pattern sets (10 stocks in consideration thus 10 pattern sets.)

For training and testing the regression model (LS), BPNs and Elman RNNs, each pattern set is broken into 16 subsets: 8 subsets for training and 8 subsets for testing (1972 – 1999 for training and 2000 for testing, 1973-2000 for training and 2001 for testing correspondingly to the eighth set 1978-2006 for training and 2007 for testing). But for training and testing the Evolutionary ANNs, each pattern set from 1972-1997 is broken into 5 subsets (12 months for 5 years thus 60 patterns each.)

The subsets then form 5 training sets corresponding with a validating set for training and testing to obtain MCVs' value (see Figure 5.5). Then the best group of genes, which has the minimum MVCs of the last generation for each stock, is deployed to structure ANNs. The same pattern sets are used to train and test LS, BPNs and Elman RNNs.

The parameters for all BPNs, Elman RNNs and final training testing for EANNs are as follows: Since the experiment involves both evolutionary phase and training phase which takes a lot of time, epoch limit is 100 and error limit is 0.0. The learning constant is 1. The (initial for EANN) architecture is 1 hidden layer with 2 nodes. The step-up evolution algorithm was run 50 generations on population of 10 ANNs (5 BPNs and 5 Elman RANNs) with conditional mutation probability at 1 (a mutation will affect only if there is an improvement).

5.7 Results Analysis

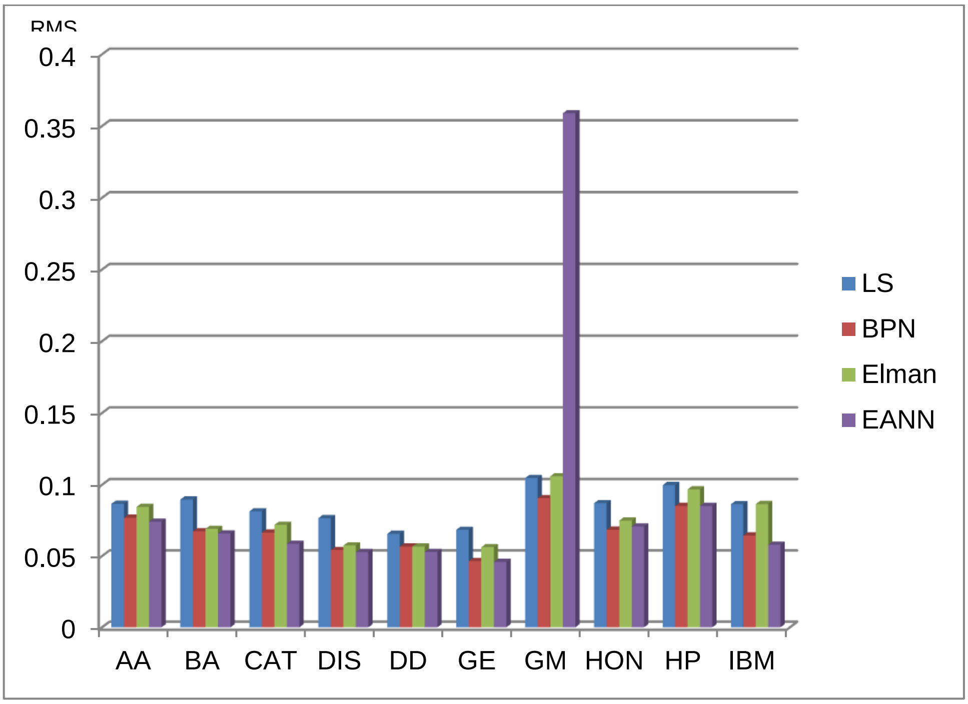


Figure 5.6: Comparing Average MCV values (Root Mean Square Error-RMSE) of Linear Regression (LS), Backpropagation ANN (BPN) and Elman Recurrent ANN (Elman)

Figure 5.6 shows the results of five forecasting models's root mean square errors (RMSEs) in comparison among the others (y-axis is shown average RMSE values of the MCVs) The bar graphs show the corresponding values of LS, BPN, Elman and EANN from left to right. Note that lower RMSE value is better in the term of model's performance (means less error).

In comparison among the four methods of forecasting, namely Linear Regression(LS), Back Propagation ANN, Elman RAN and Evolutionary ANN (EANN) for ten stock returns, the result shows that BPNs have better performances in all stock return forecasts than those of LS and Elman RANs. This result shows that to increase the complexity of ANN by introducing recurrent networks does not always improve forecasting performances. For the EANNs, they are all but one (namely GM's) at least slightly better than those of BPNs. However, the improvements are not substantial. Most of EANNs have evolved only in initial weights of BPNs (AA, BA, CAT, DD and HON) only a few have evolved both initial weights and structures of BPNs (GE, HON and HPQ). Only for GM and IBM, evolved Elman RANs are selected. The experiment shows that optimal structures of ANN for prediction stock returns are not complicated. ANNs with only a single hidden layer mostly outperformed those with more hidden layers, thus they are not evolved to have more hidden layers. Also BPNs, which are comparatively simpler than Elman RANs, are mostly selected.

5.8 Chapter Summary

The experimental results show that ANNs have the potential to make a better forecasting of financial and economics time series. In this research, a step has been taken by automatically evolving both initial and structures (number of hidden nodes and number of hidden layers). Compared to the traditional Linear Regressions, the ANNs show promising results and most of the proposed EANNs can improve the performances of the ANNs as expected even if the improvements are slight. There is ample room for further research. Firstly, the running time is quite long -- about 36 hours for each stock return forecast. Due to limited computer power, one is unable to experiment with many populations and many generations. Running the experiment in parallel high performance computer may improve the results, because we could run an experiment with more population members and more generation. This might render more accurate forecasts. The EANN algorithm also can be modified for further improvement in accuracy, such as introducing more variations of ANNs or selection of inputs. To apply the EANN to other related forecasting problems such as predicting stock volatilities, exchange rates, etc., is quite a natural step to do which we will use the EANN as choices of forecasting models in the next chapter.

Chapter 6 Fuzzy Model Selection

6.1 Chapter Overview

In this chapter, we design and test an algorithm to select the best forecasting model from a number of relevant models as we have surveyed in chapter 4. The idea is that a particular model may be best suited for use in some economic situation but not in others. If we can find rules linking a model and prevailing economic conditions at the time of forecasting, it would be helpful for better selection of forecasting models. If so, the predicting results will be closer to actual outcomes.

There are many classes of models that can be used to forecast stock returns and volatilities. A class of model may perform best in some situations but not for all situations and all time. Its performance may depend on the prevailing economic and market situations. In this chapter, adaptive and dynamic model selection mechanisms are introduced. The selection rules are based on economic and financial variables. Hopefully, pooling all of the forecasting classes of models together will indeed improve the accuracy of the portfolio optimisation inputs and eventually improve the performance of portfolio selections.

6.2 Methodology and Algorithm

Fuzzy logic is a methodology in what is called “soft computing”. It is the main methodology to build expert systems where expert rules are combined and used for a particular knowledge domain. To select the best forecasting model given economic situations, there needs to be some rules, and the rules should not be made of crisp values otherwise they would be model relationships themselves. Thus, we design an automatic selection system based on Fuzzy logic and GA. Fuzzy logic theory has been discussed in Section 3.8, Chapter 3.

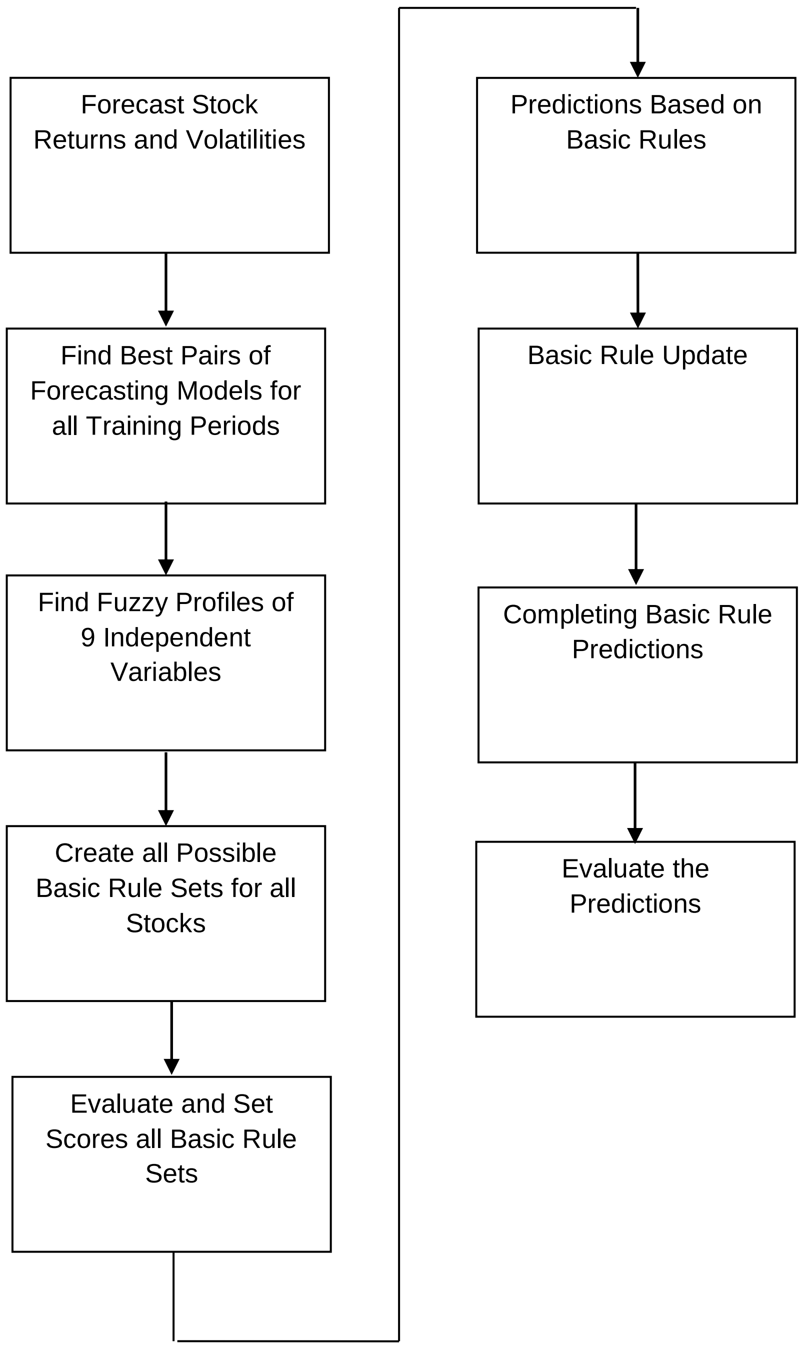


Figure 6.1: Model Selection Algorithm for Predictions of Stock Returns and Volatilities

The methodology can be described into 9 steps. The first step is to make forecasting of the interested values, in this research, stock returns and volatilities. In the second step, appropriate fuzzy profiles for the selecting variables (independent variables or input variables) are determined. Then, the variable data are to be fuzzified. The next step is to find for a given period which pair of models is the best in the term of minimising portfolio sharp ratio error. The best pair of models for entire training periods is determined. The fourth step is to generate all possible single variable rules of prediction, called the basic rule set. After the basic rule sets are formed, in the fifth step, they are individually evaluated based on the fuzzified training data against the best pair of forecasting models. The scores are evaluated and set based on the right predictions adjusted by the degree of fuzzy membership. Then, all of the basic rules are ranked according to their scores. In the sixth step, a number of basic rule set predictions, namely best variable- best rules, average best rules of all variables and average all variable-all rule, are performed. After, making prediction in each period, in order to make the basic rule set adaptive and up to date, it is updated to reflect the newly available data (the seventh step). After all predictions have been complete (the eighth step), its performance is evaluated according their portfolio sharp-ratio errors (the ninth step).

6.3 Forecasting Models

There are nine classes of model which apply to both stock returns and volatilities. The nine classes consist of 1) auto-regressive and moving average models (ARMA), 2) multi-variable ordinary least square fit models (OLS), 3) auto-regressive back propagation neural network models (BPN-AR or BPN1), 4) multi-variable back propagation neural network models (BPN-M or BPN2), 5) auto-regressive Elman neural network models (Elman-AR or ELM1), 6) multi-variable Elman neural network models (Elman-M or ELM2), 7) auto-regressive genetic neural network (GNN-AR or GANN1), 8) multi-variable genetic neural network (GNN-M or GANN2) and 9) historical mean models (MEAN).

Ordinary Least Square (OLS) is a technique most often used in Econometric model estimations for economic analyses and forecasting. OLS models assume linear relationship between the dependent variables and its corresponding independent variables. OLS models are in the form as follow (Greene 2008):

$$y = a + b_1 x_1 + b_2 x_2 + \dots + b_N x_N + e \qquad (6.1)$$

Where,y is the dependent variable of the equation i.e. the variable to be forecasted,

x_1, x_2 and x_N are independent variables,

a is the equation intercept term ,

b_1, b_2 and b_N are coefficients of the variable x_1, x_2 and x_N respectively and

e is the error term.

The actual OLS model is described in Table 6.1.

ARMA is a class of Time Series Regressions which are standard techniques to deal with forecasting stock returns and volatilities given that all stock returns and volatilities are time series in nature. Time series regressions assume linear relationship among independent variables and dependent variables. The differences from other regressions are they also include either autoregressive terms or moving average terms. The simplest kind of time series models is Autoregressive model

(AR) which has only lag-time dependent variables as independent variables. One can describe Autoregressive model as (Rachev 2007):

$$x_t = a + b_1 x_{t-1} + b_2 x_{t-2} + \ldots + b_N x_{t-N} + e \qquad (6.2)$$

Where, $x_{t,}$, $x_{t-1,}$ x_{t-2} and x_{t-N} are the value of interested variable at time t, t-1, t-2 and t-N respectively,

a is the equation intercept term ,

$b_{1,}$ b_2 and b_N are coefficients of the variable x_{t-1}, x_{t-2} and x_{t-N} respectively and

e is the error term.

Moving Average model (MA) is a time series model composed only of moving average. The moving average included in a model may be ranged from previous period lagged term to N period lagged term. One can describe the model as follow (Rachav 2007):

$$x_t = a + c_1 e_{t-1} + c_2 e_{t-2} + \ldots + c_N x_{t-N} + e \qquad (6.3)$$

Where, $x_{t,}$, $x_{t-1,}$ x_{t-2} and x_{t-N} are the value of interested variable at time t, t-1, t-2 and t-N respectively,

a is the equation intercept term ,

$c_{1,}$ c_2 and c_N are coefficients of the variable e_{t-1}, e_{t-2} and e_{t-N} respectively and

e is the error term.

AR(p) is referred to an Autoregressive model which includes lagged terms from t = t-1 to t = t-p. So as MA(q) is referred to a Moving Average model which included lagged terms from t = t-1 to t = t-q.

A model with both moving average and autoregressive terms is called ARMA model which can be specifically stated ARMA (p, q) to inform that how many autoregressive lagged terms and how many moving average lagged terms are included in the models. ARMA (p, q) can be described as follow (Rachav 2007):

$$x_t = a + b_1 x_{t-1} + b_2 x_{t-2} + \dots + b_p x_{t-p} + c_1 e_{t-1} + c_2 e_{t-2} + \dots + c_q x_{t-q} + e \quad (6.4)$$

Where, x_t,, x_{t-1}, x_{t-2} and x_{t-p} are the value of interested variable at time t, t-1, t-2 and t-p respectively,

a is the equation intercept term ,

b_1, b_2 and b_p are coefficients of the variable x_{t-1}, x_{t-2} and x_{t-p} respectively,

c_1, c_2 and c_q are coefficients of the variable e_{t-1}, e_{t-2} and e_{t-q} respectively and

e is the error term.

The actual ARMA model is described in Table 6.1.

Auto-regressive back propagation neural network models (BPN-AR or BPN1), Multi-variable back propagation neural network models (BPN-M or BPN2), Auto-regressive Elman neural network models (Elman-AR or ELM1) and Multi-variable Elman neural network models (Elman-M or ELM2) have been described in Section 5.4, Chapter 5. The actual specifications of these models are given in Table 6.1.

Auto-regressive genetic neural network (GNN-AR or GANN1) and Multi-variable genetic neural network (GNN-M or GANN2) are constructed according to Section 5.5, Chapter 5. The actual specifications of these models are also given in Table 6.1

Historical mean models (MEAN) are simply the average of past values. They can be stated mathematically as follow:

$$x_t = \frac{\sum_{i=1}^{N} x_{t-i}}{N} \qquad (6.5)$$

Where, $x_{t,}$is interested variable at time t (predicted),

x_{t-i} is the past (i^{th} previous period realised value) of interested variables,

N is the number of periods.

For ARMA, OLS, BPN-AR, BPN-M, Elman-AR, Elman-M and MEAN, forecasting models are estimated from the previous 18 years of data (216 month periods) starting from January 1971 (1971 M01) to December 1988 (1988 M12). The forecasting models are re-estimated each year with shifting windows of 18 years of data (216 months) for the subsequent 18 years ending 2006. The estimated models are used to forecast the next 12 months from the ending of the estimating data, e.g. the first estimated models using data from 1971 M1 to 1988 M12 estimating data are used to forecast stock returns and volatilities from 1989 M1 to 1989 M12. The subsequent models are estimated from the next 216 months of data with 12-month shifting window to the last window of estimating data of 1989 M01 to 2006 M12. The last set of forecasting values are from 2007 M1 to 2007 M12. For GNN-AR and GNN-M, the estimation of models needs to be done in two phases. First, the optimal structures of Artificial Neural Networks (i.e. the number of hidden layers and the number of hidden nodes in each layer) must be chosen by means of evolutionary algorithms. Second, the optimally structured ANNs are trained (estimated) to be ready for forecasting. Due to limited data availability, the evolutions of ANN structures use the subset of data of those for ANN training. In the structure evolution phases, by the cross validation method, previous 15 years of data are partitioned into 5 cohorts, of 3 years each. Four cohorts are randomly chosen for training each of the population of ANNs. The remaining cohort is for choosing (for validating) the ANNs (to calculate the value of the objective function i.e. prediction error). After the optimal ANN structure has been chosen, the ANN is trained by previous 18 years of data. The shifting window of data also applied for GNN-AR and GNN-M as in the rests of models described above. The forecasting models produce 9 series of stock return forecasting and 9 series of stock volatility forecasting for all 17 stocks from 1989 M01 to 2007 M12 which are used in the next steps.

Model	Inputs for Return	Inputs for Volatility	Structure	Output
ARMA	return(-1) to return(-6)	volatility(-1) to volatility (-6)	$x_0 = a_1x_{-1} + \ldots + a_6x_{-6} + e$	$return^f(0)$ or $volatility^f(0)$
OLS	return(-1), dCPI(-1), dBTY(-1), dRPM(-1), dFED(-1), dM1(-1), dUM(-1), dSP500(-1) and dIPI(-1)	volatility(-1), dCPI(-1), dBTY(-1), dRPM(-1), dFED(-1), dM1(-1), dUM(-1), dSP500(-1) and dIPI(-1)	$x_0 = a_1x_{-1} + a_2dCPI_{-1} + a_3dBTY_{-1} + a_4dRPM_{-1} + a_5dFED_{-1} + a_6dM1_{-1} + a_7dUM_{-1}\ \ a_8dSP500_{-1} + a_9dIPI_{-1} + e$	$return^f(0)$, $volatility^f(0)$
BPN-AR (BPN1)	return(-1) to return(-6)	volatility(-1) to volatility (-6)	Inputs: 6 Nodes Hidden1: 6 Nodes Output: 1 Node	$return^f(0)$ or $volatility^f(0)$
BPN-M (BPN2)	return(-1), dCPI(-1), dBTY(-1), dRPM(-1), dFED(-1), dM1(-1), dUM(-1), dSP500(-1) and dIPI(-1)	volatility(-1), dCPI(-1), dBTY(-1), dRPM(-1), dFED(-1), dM1(-1), dUM(-1), dSP500(-1) and dIPI(-1)	Inputs: 10 Nodes Hidden1: 10 Nodes Output: 1 Node	$return^f(0)$ or $volatility^f(0)$
Elman-AR (ELM1)	return(-1) to return(-6)	volatility(-1) to volatility (-6)	Inputs: 6 Nodes Hidden1: 6 Nodes Output: 1 Node (Recurrent)	$return^f(0)$ or $volatility^f(0)$
Elman-M (ELM2)	return(-1), dCPI(-1), dBTY(-1), dRPM(-1), dFED(-1), dM1(-1), dUM(-1), dSP500(-1) and dIPI(-1)	volatility(-1), dCPI(-1), dBTY(-1), dRPM(-1), dFED(-1), dM1(-1), dUM(-1), dSP500(-1) and dIPI(-1)	Inputs: 10 Nodes Hidden1: 10 Nodes Output: 1 Node (Recurrent)	$return^f(0)$ or $volatility^f(0)$
GNN-AR (GANN1)	return(-1) to return(-6)	volatility(-1) to volatility (-6)	Inputs: 6 Nodes Hidden Layers: Varied Hidden Nodes in each layer: Varied Output: 1 Node Elman or BPN	$return^f(0)$ or $volatility^f(0)$
GNN-M (GANN2)	return(-1), dCPI(-1), dBTY(-1), dRPM(-1), dFED(-1), dM1(-1), dUM(-1), dSP500(-1) and dIPI(-1)	volatility(-1), dCPI(-1), dBTY(-1), dRPM(-1), dFED(-1), dM1(-1), dUM(-1), dSP500(-1) and dIPI(-1)	Inputs: 10 Nodes Hidden Layers: Varied Hidden Nodes in each layer: Varied Output: 1 Node Elman or BPN	$return^f(0)$ or $volatility^f(0)$
MEAN	return(-1) to return(-12)	volatility(-1) to volatility(-12)	$x_0 = (x_{-1} + \ldots + x_{-12})/12$	$return^f(0)$ or $volatility^f(0)$

Table 6.1 Descriptions of 9 Selectable models

Note that in Table 6.1 dCPI(-1) = change in Consumer Price Index (-1), dBTY(-1) = change in bond yield term spread(-1), dRPM(-1) = change in risk premium(-1), dFED(-1) = change in fed-fund rate(-1), dM1(-1) = change in money supply(-1), dUM(-1) = change in unemployment rate(-1), dSP500 = change in stock index (S&P 500)(-1) and dIPI(-1) = change in industry product index(-1); where (-1) stands for t-1 or the values of the previous period. And (-2) to (-12) are (t-2) to (t-12) or n periods to the past.

6.3 Impact of Input Errors to the Output of Portfolio Optimisation

6.3.1 Portfolio's Sharp Ratio Error

Sharp Ratio is a ratio that measures asset's or portfolio's returns adjusted by risk. Risk in the ratio is measured by standard deviation of asset's or portfolio's returns. Sharpe Ratio is defined as (Sharpe 1964)

$$SR_p = \frac{R_p - R_f}{\sigma_p} \tag{6.6}$$

Where,

SR_p is Sharpe Ratio of a portfolio,

R_p is return of the portfolio,

R_f is the risk-free rate of return,

σ_p is Standard deviation of the portfolio.

Sharpe Ratio is essentially standardised excess return above risk free rate per unit of absolute risk as measured by standard deviation. In a sense, returns of assets or of portfolio of assets cannot be compared directly because they may have different level of inherent risks. However, Sharpe Ratio has adjusted the level of risk so that it represents a standardised return that can be comparable among different assets and portfolios. For summarily comparing two assets or two portfolios, Sharp Ratio is the right tool to do the job. To compare two portfolios, the larger Sharp Ratio portfolio is the better one.

The objective is to find a portfolio that can handle model risk effectively. The proposed algorithm needs prove that it can render better portfolio, i.e. portfolio with larger Sharp Ratio on average and for large number of time periods. Portfolio optimisation is to construct a portfolio that satisfies two objectives simultaneously, i.e. minimisation of risk as measured by the portfolio's standard deviation and maximisation of the portfolio's return. This is actually to maximise Sharpe Ratio at a given level of returns or at a given level of risk. Let consider two slightly different portfolios which have a slightly different proportion of asset w_i. Comparing a forecasted portfolio's Sharpe Ratio and that of an actual portfolio's, the error of Sharpe Ratio (ESR_p) can be calculated as follows:

$$ESR_p = \left(\frac{R'_p - R_f}{\sigma'_p}\right) - \left(\frac{R_p - R_f}{\sigma_p}\right) \quad (6.7)$$

$$\frac{\partial ESR_p}{\partial w_i} = \left(\frac{1}{\sigma'_p}\frac{\partial R'_p}{\partial w_i} - \frac{R'_p}{\sigma'^2_p}\frac{\partial \sigma'_p}{\partial w_i} + \frac{R_f}{\sigma'^2_p}\frac{\partial \sigma'_p}{\partial w_i}\right) - \left(\frac{1}{\sigma_p}\frac{\partial R_p}{\partial w_i} - \frac{R_p}{\sigma^2_p}\frac{\partial \sigma_p}{\partial w_i} + \frac{R_f}{\sigma^2_p}\frac{\partial \sigma_p}{\partial w_i}\right) \quad (6.8)$$

For simplicity, let assume that $R_f = 0$.

$$\frac{\partial ESR_p}{\partial w_i} = \left(\frac{1}{\sigma'_p}\frac{\partial R'_p}{\partial w_i} - \frac{R'_p}{\sigma'^2_p}\frac{\partial \sigma'_p}{\partial w_i}\right) - \left(\frac{1}{\sigma_p}\frac{\partial R_p}{\partial w_i} - \frac{R_p}{\sigma^2_p}\frac{\partial \sigma_p}{\partial w_i}\right) \quad (6.9)$$

Let us consider,

$$\frac{\partial R_p'}{\partial w_i} = \frac{\partial \sum w_i R_i'}{\partial w_i} = R_i' \tag{6.10}$$

$$\frac{\partial R_p}{\partial w_i} = \frac{\partial \sum w_i R_i}{\partial w_i} = R_i \tag{6.11}$$

$$\frac{\partial \sigma_p'}{\partial w_i} = \frac{\partial \sum_i \sum_j w_i w_j \rho_{ij} \sigma_i' \sigma_j'}{\partial w_j} = \frac{1}{2\sqrt{\sigma_p'}} \left(\sum_{j \neq i} w_j \rho_{ij} \sigma_i' \sigma_j' + 2 w_i \sigma_i'^2 \right) \tag{6.12}$$

$$\frac{\partial \sigma_p}{\partial w_i} = \frac{\partial \sum_i \sum_j w_i w \rho_{ij} \sigma_i \sigma_j}{\partial w_j} = \frac{1}{2\sqrt{\sigma_p}} \left(\sum_{j \neq i} w_j \rho_{ij} \sigma_i \sigma_j + 2 w_i \sigma_i^2 \right) \tag{6.13}$$

Substitute (6.10), (6.11), (6.12) and (6.13) in (6.9),

$$\frac{\partial ESR_p}{\partial w_i} = \left(\frac{R_i'}{\sigma_p'} - \frac{R_p'}{2\sigma_p'^{5/2}} \left(\sum_{j \neq i} w_j \rho_{ij} \sigma_i' \sigma_j' + 2 w_i \sigma_i'^2 \right) \right) - \left(\frac{R_i}{\sigma_p} - \frac{R_p}{2\sigma_p^{5/2}} \left(\sum_{j \neq i} w_j \rho_{ij} \sigma_i \sigma_j + 2 w_i \sigma_i^2 \right) \right) \tag{6.14}$$

By rearranging terms, one has got,

$$\frac{\partial ESR_p}{\partial w_i} = \left(\frac{R_i'}{\sigma_p'} - \frac{R_i}{\sigma_p} \right) + \left(\frac{R_p}{\sigma_p^{\frac{5}{2}}} w_i \sigma_i^2 - \frac{R_p'}{\sigma_p'^{\frac{5}{2}}} w_i \sigma_i'^2 \right) + \left\{ \frac{R_p}{2\sigma_p^{\frac{5}{2}}} \left(\sum_{j \neq i} w_j \rho_{ij} \sigma_i \sigma_j \right) - \frac{R_p'}{2\sigma_p'^{\frac{5}{2}}} \left(\sum_{j \neq i} w_j \rho_{ij} \sigma_i' \sigma_j' \right) \right\} \tag{6.15}$$

Let us consider equation (6.15), according to the assumptions, one assumes that the original portfolio is optimal and the second portfolio is only infinitesimally changed from the original, thus we can approximate that

$$R_p' \approx R_p \qquad (6.16)$$

And,

$$\sigma_p' \approx \sigma_p \qquad (6.17)$$

Terms inside the bracket {} are correlations of the assets. Since correlations between pair of assets are assumed to be the same from both cases, thus it can be approximated as follow:

$$\sum_{j \neq i} w_j \rho_{ij} \sigma_i \sigma_j \approx \sum_{j \neq i} w_j \rho_{ij} \sigma_i' \sigma_j' \qquad (6.18)$$

From the equations (6.17) and (6.18), the third term is vanished and substituted in both equations into (6.15) given:

$$\frac{\partial ESR_p}{\partial w_i} = \frac{R_p}{\sigma_p^{\frac{5}{2}}} w_i \sigma_i^2 - \frac{R_p'}{\sigma_p^{\prime\frac{5}{2}}} w_i \sigma_i'^2 \qquad (6.19)$$

6.4 Best Pairs of Prediction Models

There are 9 classes of models for predicting stock returns and another 9 classes of models for predicting stock volatilities. There are 17 stocks to predict and 216 monthly periods to forecast. The best class of model for each prediction is the one which has minimum error in each period. However, in this research, the best pair of prediction models which yield minimum error for portfolio selection are to be determined. Thus, the best pair should neither be select based on their return model nor their volatility model alone but they should be selected based on a combination of both. An approximation of error for portfolio selection are used to evaluate and also to be the objective function for both Basic Fuzzy Rule set selection and the complex Fuzzy Rule Set selection by GA. By rearranging the terms of equation (6.19), the approximation of a unit change in Sharp Ratio Error is given as follows:

$$\Delta ESP_p = \frac{1}{\sigma_p}\left(R_i^f - R_i\right) + \frac{R_p}{\sigma_p^{\frac{3}{2}}}\left(w_i\sigma_i^2 - w_i\sigma_i^{f2}\right) \qquad (6.20)$$

Where, R_i^f and σ_i^f are forecasted return and forecasted volatility from a return prediction model and from volatility prediction model for stock i^{th} at a given period respectively.

R_i and σ_i are actual return and actual volatility respectively for stock i^{th} at a given period respectively.

R_p and σ_p are approximated portfolio return and volatility respectively. Since no specific portfolio is predetermined, the market portfolio (i.e. S&P 500 index) is assumed. It has average return in the period from 1980 to 1999 of 0.011874 and volatility of S&P 500 index in the same periods has the average of 0.043027.

w_i is an approximated portion of stock i^{th} in the portfolio. The portfolio of equal weighted stock is taken as general case. So, each stock has 1/17 weight in the portfolio i.e. 0.0588.

In each period from 1989 M1 to 2007 M12, ΔESP_p is calculated for all possible combinations of the return models and the volatility models (9x9 = 81 combinations) and find their minimum values. The return model and the volatility model pair which produce the minimum value are set as the best pair of forecasting models in that period. Note that less Sharp Ratio error is preferred to more, regardless of its absolute value. Neither an absolute operator nor a square is appropriate here.

6.5 Fuzzy Profiles

Before fuzzification of the independent (predicting) variables can be made, the fuzzy profiles are to be determined. As a rule of thumb, the number of fuzzy profiles should be set as minimum but also be able to capture the distinction nature of value in question. Because fuzzy profiles are set based upon the linguistic meaning of ordinal classification which we tend to measure things in just only a few orders such as very hot, hot, warm, cold, very cold. Therefore, in this case, it is deemed appropriate that five profiles for each of those variables should suffice. The profiles are Very Low (VL), Low (L), Medium (M), High (H) and Very High (VH). The shape of the profiles is set to be linear for Very High profile, inverted linear for Very Low and spike profiles for those of Low, Medium and High. For the lowest points (fuzzy membership degree = 1) of the Very Low linear profile and the highest points of the Very High inverted linear profile are set to the minimum values and maximum values in the ranges of data of the predicting variables from estimating period (1971 M1 to 1988 M12). For those of spike profiles, the middle points of Low, Medium and High are set by the Fuzzy C-Means Algorithm (Yang 1993). To satisfy the condition that degrees of fuzzy membership are always in unity, the lowest points of the Very Low profiles are the middle points of Low profiles and the lowest points of Very High profiles are the middle points of High profiles. And the middle points of the Low profiles are the lowest points of the Medium profiles so as the middle points of the

High profiles are the highest point of the Medium profiles. Thus, all 5 profiles of a variable can be described by a vector of 5 numbers. Figure 6.2 shows the fuzzy membership profiles.

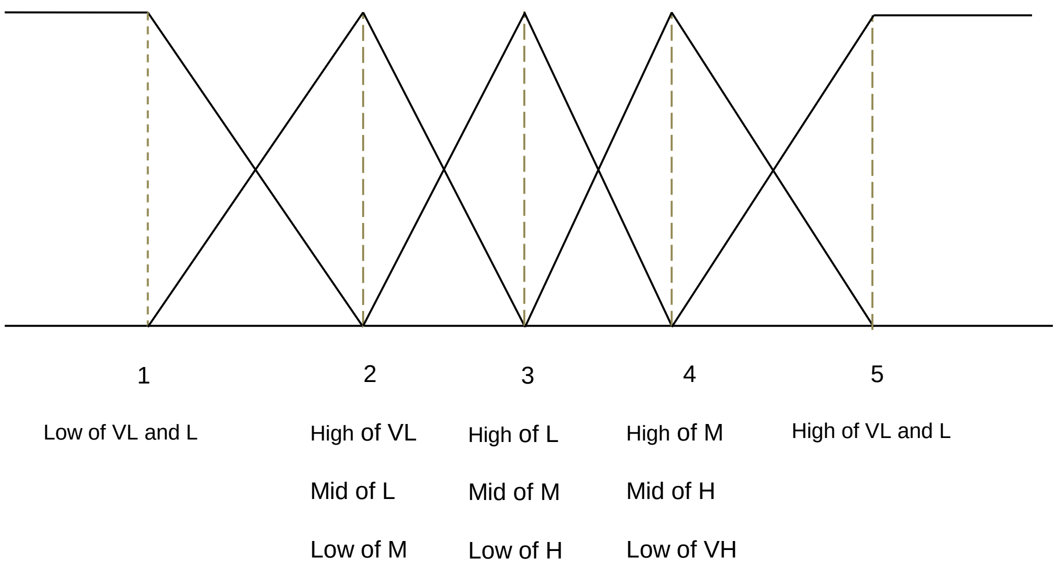


Figure 6.2: Fuzzy Membership Profiles of the Predicting Variables

For each of the predicting variables, point 1 and point 5 have been set to their minimum values and maximum values in the range of the estimating period as mentioned above. However, points 2, 3 and 4 are needed to determine the Fuzzy C-Means (FCM) Clustering Algorithm. It is an algorithm for clustering which determines one piece of data to belong to two or more clusters. The FCM is an algorithm to maximise the following objective function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \left\| x_i - c_j \right\|^2 \tag{6.21}$$

Where *m* is any real number greater than 1, in this experiment, it is set to be 2.00, in order to be consistent with the term of $||x_i\text{-}c_j||$;

u_{ij} is the degree of membership of x_i in the cluster j;

x_i is the i^{th} of d-dimensional measure data;

c_j is the d-dimension center of the cluster and

||*|| is any norm expressing the similarity between any measure data and the centre.

The algorithm is an iterative optimisation of the objective function shown in (6.21). It consists of the following steps:

Step 1: Initialise $U_{ij} = (u_{ij})$ matrix as $U(0)$ which is derived from equal partitioning from the maximum values (points 1) and the minimum values (points 5) of points 2, 3 and 4 of the vector C.

Step 2: At k-step, calculate the centres of vectors C (k) = (c_j) with $U(k)$ using the following equation:

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \tag{6.22}$$

Step 3: Update $U(k)$ and $U(k+1)$

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{\|x_i - c_j\|}{\|x_i - c_j\|}\right)^{\frac{2}{m-1}}} \tag{6.23}$$

If $\|U(k+1) - U(k)\| < \varepsilon$ then STOP, else return to Step 2 (ε is set to 0.0000001).[i]

The algorithm yields Fuzzy C-Means of the data of three partitioned intervals between the maximum values and the minimum values of each predicting variables which are the points 2, 3 and 4 for creating its respective fuzzy profiles. The fuzzy profiles will be used to fuzzify the predicting variables both in finding the best basic rule set and the optimal complex rule set. They will also used in the forecasting periods.

6.6 Fuzzy Basic Rule Set

A Fuzzy basic rule consists of a causal part and a result part. In this setting, an example of Fuzzy basic rule is "If Change in inflation (dCPI) is high, then the best pair of models are return model 1 and volatility model 4". It can be seen that there are 9 predicting variables and each of them can be in 5 Fuzzy membership profiles or sets (a realised value of any variable is usually in 2 Fuzzy membership sets at the same time). So, there are 9x5 or 45 possible causal units to form a causal part. For the result part, there are 9 return prediction models and 9 volatility prediction models or 81 possible combinations. Thus, in totality, there are 45x81 or 3645 Fuzzy basic rules. Obviously, the Fuzzy basic rules here are not truly Fuzzy rules because their result parts are not fuzzified values but pairs of models or pairs of objects which point to pairs of predefined crisp values. Since there are a large number of rules, they are organised in a particular order for convenience of scoring and utilisation. The Fuzzy basic rule sets are collected firstly as groups of rules for particular stocks, then as groups for the predicting variables and lastly as groups for Fuzzy profiles, as shown in Figure 6.3.

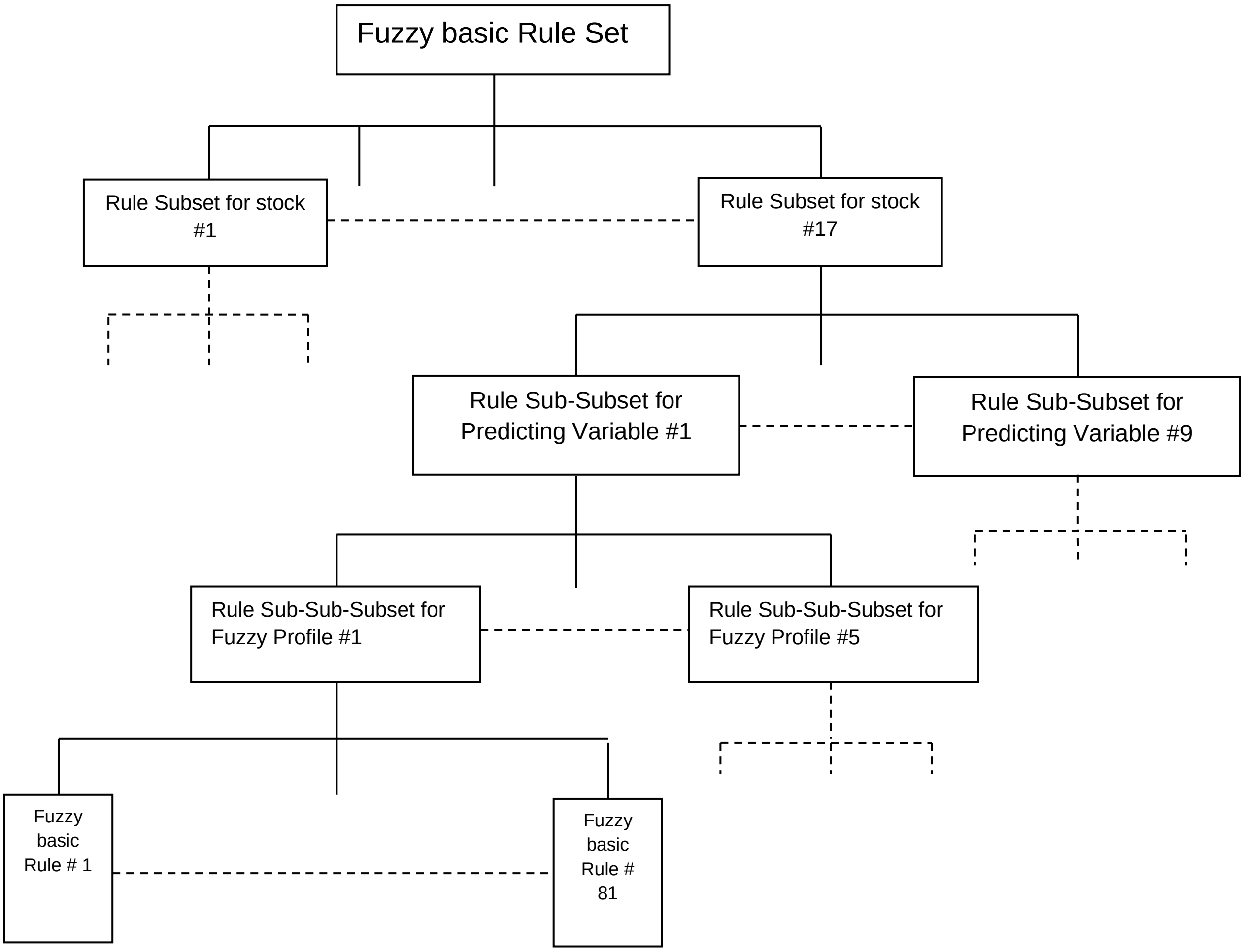


Figure 6.3: Organisation of Fuzzy Basic Rule Set

6.7 Evaluation of the Fuzzy Basic Rules

All possible Fuzzy basic rules are evaluated based on their predictive ability. If a Fuzzy basic rule can correctly predict the best pair of forecasting models using just the previous value of the predicting variable, it is given a credit weighted the predictive variable's degree of Fuzzy membership according to the Fuzzy profile stated in the causal part of the rule. Summing up credits for entire of the training period, each of the Fuzzy basic rules has its own score. A score for a Fuzzy basic rule of predicting variable i^{th} with Fuzzy profile j^{th} and result combination k^{th}, accumulated through period 0 to period T (S_0^T), can be stated as follows:

$$S_0^T(i,j,k) = \sum_{t=0}^{T} M_t\,(i,j)C_t(k) \tag{6.24}$$

Where M_t *(i,j)* is Fuzzy membership degree of the realised value of predicting variable i^{th} of Fuzzy profile (set) j^{th} at time period t;

C_t *(k)* is credit given to the combination pair of prediction models. If k refers to prediction models (e.g. return model #4 and volatility model #8) which match the best pair of prediction models for that time period t, it will have value equal to one, otherwise its value is zero.

After all Fuzzy basic rules are completely scored, those contained in each group of sub-sub-subset for Fuzzy profile, which have 81 rules, are sorted by descending order according to their scores. This sorted Fuzzy basic rule set will be used for predictions by model selections on step 7 (see Figure 5.1).

6.8 Single Best Variable Best Rule (Basic 1)

There are three ways to use Fuzzy basic rule set to predict stock returns and stock volatilities. First, the values are predicted from the highest scored Fuzzy rule from all predicting variables. This method can be called as “Single Best Variable Best Rule” or SBVBR or Basic Rule Prediction 1. For each stock s, it can be stated mathematically for each time period predictions of stock return (r_s^e) and of stock volatility (σ_s^e) respectively as follows:

$$r_s^e = Max(CFS_v^{max})_{\forall v} \sum_{i=1}^{5} r_{max}^e M(v,i)S(v,i,max) \Big/ CFS_v \tag{6.25}$$

$$\sigma_s^e = Max(CFS_v^{max})_{\forall v} \sum_{i=1}^{5} \sigma_{max}^e M(v,i) S(v,i,max) \Big/ CFS_v \quad (6.26)$$

$$\mathrm{CFS}_v^{max} = \sum_{i=1}^{5} M(v,i) S(v,i,max). \quad (6.27)$$

Where *M (v, i)* is Fuzzy membership degree of the realised value of predicting variable v of Fuzzy profile (set) i^{th} for the predicting time period.

S (v, i, max) is a cumulative score prior to predicting time period according to (6.24) given to the combination pair of prediction models which have the maximum score of Fuzzy profile i of subset of all k (or all combinations of pairs of predicting models) of variable v.

Max (CFS_v^{max}) is the maximum value of CFS_v^{max} (6.27) selected from all predicting variable v.

6.9 Multiple Variable Best Rule (Basic 2)

The second method is called “Multiple Variable Best Rule” or MVBR or Basic Rule 2. This method is different from SBVBR in such a way that the predictions are weighted averages of all predictions from the predicting variables. The weighted averages are the products of corresponding Fuzzy membership degrees of the particular predicting variables and rule scores for the combination pair of prediction models which have the maximum score of the applicable Fuzzy profile of subset of the predicting variable or CFS_v^{max} according to (6.27). The predictions of the stock return and volatility for a time period are;

$$r_s^{e2} = (\sum_{v=1}^{9} r_s^e \, . \, CFS_v^{max}) / \, (\sum_{v=1}^{9} CFS_v^{max}) \tag{6.28}$$

$$\sigma_s^{e2} = (\sum_{v=1}^{9} \sigma_s^e \, . \, CFS_v^{max}) / \, (\sum_{v=1}^{9} CFS_v^{max}). \tag{6.29}$$

6.10 All Variables All Rules (Basic 3)

The third method is called “All Variables All Rules” or AVAR or Basic Rule 3. This method uses all of the rules (which are equal to the combination of predicting models) of all Fuzzy profile sets of all predicting variables, thus 81 for each of Fuzzy profile set for each predicting variables. The predictions of stock return and volatility of a particular period are the results of weighted average values based on each of current individual rule score and the corresponding Fuzzy membership degrees of the particular predicting variables. The weighted factor or CFS of a particular Fuzzy rule is given as follows:

$$\text{CFS (v, i, k)} = \quad M(v,i)S(v,i,k) \tag{6.30}$$

Where *M (v, i)* is Fuzzy membership degree of the realised value of predicting variable v of Fuzzy profile (set) i for the predicting time period.

S (v, i, k) is cumulative score prior to predicting time period according to (5.19) given to the combination pair of prediction models k of Fuzzy profile i of predicting variable v.

The predictions of the stock return and volatility for a time period are:

$$r_s^{e3} = \sum_{v=1}^{9} \sum_{i=1}^{5} \sum_{k=1}^{81} \, (r_s^e(k) \, . \, CFS(v,i,k)) \, / \sum_{v=1}^{9} \sum_{i=1}^{5} \sum_{k=1}^{81} CFS(v,i,k) \tag{6.31}$$

$$\sigma_s^{e3} = \sum_{v=1}^{9}\sum_{i=1}^{5}\sum_{k=1}^{81} (\sigma_s^e(k)_s^e . CFS(v,i,k)) / \sum_{v=1}^{9}\sum_{i=1}^{5}\sum_{k=1}^{81} CFS(v,i,k) \quad (6.32)$$

6.11 The Experimental Results and Analysis

A number of experiments are conducted to find out whether the proposed Fuzzy model selection methods as described above will indeed outperform forecasting performance of each class of models. Since the Fuzzy selection is based on errors in forecasting Sharpe ratio of each stock (6.20), the performance measurement is based on the average portfolio Sharpe ratio errors of the Fuzzy selection algorithmic forecasting values (i.e. SBVBR or Basic Rule 1, MVBR or Basic Rule 2 and AVAR or Basic Rule 3) compared with those of the forecasting models. The outcomes of 72 periods from 2002 (January) to 2007 (December), 17 stocks for each period are run and evaluated.

The Fuzzy selection algorithm Basic 1, Basic 2 and Basic 3 are compared to other single forecasting classes of models (ARMA, OLS, BPN1, BPN2, ELM1, ELM2, GANN1, GANN2 and MEAN) to see whether the forecasts of any single class of model or a proposed Fuzzy selection algorithm is more reliable for the purpose of stock portfolio optimisation. The results of average portfolio Sharpe ratio errors of each proposed forecasting algorithm with the other single models for 17 stocks as the sample for all experiment for this thesis are presented in Table 6.2, 6.3 and 6.4 accordingly. Finally, we include all of Basic 1, Basic 2 and Basic 3 in comparison among themselves and the other single forecasting model in Table 6.5.

Stock ID	Basic 1	ARMA	OLS	BPN1	BPN2	ELM1	ELM2	GANN1	GANN2	MEAN	Min	Best
AA_PSR	0.2728	0.4009	0.3273	0.6493	0.5614	1.8507	0.8321	15.004	6.2323	-0.031	-0.031	MEAN
BA_PSR	8.3511	-0.151	-0.058	0.4196	0.1692	1.46	0.4162	-0.039	0.6891	-0.091	-0.151	ARMA
CAT_PSR	-0.365	-0.192	-0.1	-0.274	-0.304	0.4404	-0.184	11.738	4.3998	0.0508	-0.365	Basic 1
DD_PSR	-0.007	0.1892	0.1499	0.1737	0.1662	1.1921	0.3418	0.7865	-0.1471	-0.042	-0.147	GANN2
DIS_PSR	0.2917	0.0916	0.1041	0.2396	0.2905	1.918	0.5991	-0.1058	7.5523	-0.124	-0.124	MEAN
GE_PSR	-0.057	0.1573	0.2648	0.3133	0.1315	0.6005	0.2018	-0.0384	-0.0242	-0.073	-0.073	MEAN
GM_PSR	-0.047	0.0429	0.0429	1.5521	0.2862	1.5521	0.4628	-0.0069	-0.0479	-0.073	-0.073	MEAN
HON_PSR	0.017	0.1284	0.0558	-0.14	-0.126	0.199	-0.126	7.9471	0.0396	-0.06	-0.14	BPN1
HPQ_PSR	-0.147	0.0937	-0.036	-0.154	-0.207	0.0657	-0.158	2.6202	1.0441	-0.099	-0.207	BPN2
IBM_PSR	-0.124	0.1525	0.2055	0.0597	0.0174	0.1102	0.0349	4.6284	3.1852	-0.057	-0.124	Basic1
JNJ_PSR	-0.101	0.3875	0.3457	0.1254	0.1679	1.0712	0.3007	0.1678	-0.1021	0.012	-0.102	Basic1
KO_PSR	-0.078	0.1376	0.3078	0.5186	0.3724	2.1592	0.6293	-0.0467	0.0451	-0.068	-0.078	Basic1
MMM_PSR	0.4492	0.1683	0.1164	0.2317	0.0809	1.0516	0.2176	-0.1403	0.7291	0.0158	-0.14	GANN1
MRK_PSR	-0.189	0.2331	0.4133	0.5991	0.3754	0.5991	0.5071	-0.0565	-0.057	-0.122	-0.189	Basic1
PG_PSR	-0.001	0.1753	0.0457	0.1498	0.0524	1.0653	0.2522	0.9326	1.706	0.0288	-0.001	Basic1
UTX_PSR	-0.321	0.0175	0.0853	-0.007	-0.018	0.9775	0.1217	3.5991	-0.3228	-0.009	-0.323	Basic1
XOM_PSR	-0.32	-0.084	-0.013	-0.038	-0.13	0.4098	-0.054	1.412	0.6303	-0.064	-0.32	Basic1

Table 6.2 Average Shape Ratio Errors of Basic 1 (SBVBR) and Other Forecasting Models (Column) according to stock names (Row)

Table 6.2 shows the comparison among the average Sharp Ratio errors resulting from SBVBR or Basic 1 to those of the other forecasting models The comparison shows that Basic 1 algorithm yields minimum average portfolio Sharp ratio error from 8 out of 17 stocks (CAT, IBM, JNJ, KO, MRK, PG, UTX and XOM), which is considered the best among other forecasting models. The next best is MEAN which yields minimum portfolio Sharp ratio error from 4 out of 17 stocks (AA, DIS, GE and GM). Although Basic 1 can beat all single model forecasting for 8 out of17 stock instances, 8 is still less than half of instances.

Stock ID	Basic 2	ARMA	OLS	BPN1	BPN2	ELM1	ELM2	GANN1	GANN2	MEAN	Min	Best
AA_PSR	0.2717	0.4009	0.3273	0.6493	0.5614	1.8507	0.8321	15.004	6.2323	-0.031	-0.031	MEAN
BA_PSR	7.2684	-0.151	-0.058	0.4196	0.1692	1.46	0.4162	-0.039	0.6891	-0.091	-0.151	ARMA
CAT_PSR	-0.34	-0.192	-0.1	-0.274	-0.304	0.4404	-0.184	11.738	4.3998	0.0508	-0.34	Basic 2
DD_PSR	-0.024	0.1892	0.1499	0.1737	0.1662	1.1921	0.3418	0.7865	-0.147	-0.042	-0.147	GANN2
DIS_PSR	0.2941	0.0916	0.1041	0.2396	0.2905	1.918	0.5991	-0.106	7.5523	-0.124	-0.124	MEAN
GE_PSR	-0.054	0.1573	0.2648	0.3133	0.1315	0.6005	0.2018	-0.038	-0.024	-0.073	-0.073	MEAN
GM_PSR	-0.039	0.0429	0.0429	1.5521	0.2862	1.5521	0.4628	-0.007	-0.048	-0.073	-0.073	MEAN
HON_PSR	0.0218	0.1284	0.0558	-0.14	-0.126	0.199	-0.126	7.9471	0.0396	-0.06	-0.14	BPN1
HPQ_PSR	-0.217	0.0937	-0.036	-0.154	-0.207	0.0657	-0.158	2.6202	1.0441	-0.099	-0.217	Basic 2
IBM_PSR	-0.109	0.1525	0.2055	0.0597	0.0174	0.1102	0.0349	4.6284	3.1852	-0.057	-0.109	Basic 2
JNJ_PSR	-0.081	0.3875	0.3457	0.1254	0.1679	1.0712	0.3007	0.1678	-0.102	0.012	-0.102	GANN2
KO_PSR	-0.057	0.1376	0.3078	0.5186	0.3724	2.1592	0.6293	-0.047	0.0451	-0.068	-0.068	MEAN
MMM_PSR	0.4225	0.1683	0.1164	0.2317	0.0809	1.0516	0.2176	-0.14	0.7291	0.0158	-0.14	GANN1
MRK_PSR	-0.13	0.2331	0.4133	0.5991	0.3754	0.5991	0.5071	-0.057	-0.057	-0.122	-0.13	Basic 2
PG_PSR	0.0613	0.1753	0.0457	0.1498	0.0524	1.0653	0.2522	0.9326	1.706	0.0288	0.0288	MEAN
UTX_PSR	-0.304	0.0175	0.0853	-0.007	-0.018	0.9775	0.1217	3.5991	-0.323	-0.009	-0.323	GANN2
XOM_PSR	-0.318	-0.084	-0.013	-0.038	-0.13	0.4098	-0.054	1.412	0.6303	-0.064	-0.318	Basic 2

Table 6.3 Average Shape Ratio Errors of Basic 2 (MVBR) and Other Forecasting Models (Column) according to stock names (Row)

Table 6.3 shows the comparison among the average Sharp Ratio errors resulting from MVBR or Basic 2 to those of the other forecasting models. The comparison shows that Basic 2 algorithm yields minimum average portfolio Sharp ratio error from only 5 out of 17 stocks (CAT, HPQ, IBM, MRK and XOM) which is considered the second place while those of MEAN yield 6 out of 17 stock instances (AA, DIS, GE, GM, KO and PG) which is considered the best among other forecasting models.

Stock ID	Basic 3	ARMA	OLS	BPN1	BPN2	ELM1	ELM2	GANN1	GANN2	MEAN	Min	Best
AA_PSR	-0.084	0.4009	0.3273	0.6493	0.5614	1.8507	0.8321	15.004	6.2323	-0.031	-0.084	Basic 3
BA_PSR	-0.374	-0.151	-0.058	0.4196	0.1692	1.46	0.4162	-0.039	0.6891	-0.091	-0.374	Basic 3
CAT_PSR	-0.456	-0.192	-0.1	-0.274	-0.304	0.4404	-0.184	11.738	4.3998	0.0508	-0.456	Basic 3
DD_PSR	-0.145	0.1892	0.1499	0.1737	0.1662	1.1921	0.3418	0.7865	-0.147	-0.042	-0.147	Basic 3
DIS_PSR	-0.268	0.0916	0.1041	0.2396	0.2905	1.918	0.5991	-0.106	7.5523	-0.124	-0.268	Basic 3
GE_PSR	-0.073	0.1573	0.2648	0.3133	0.1315	0.6005	0.2018	-0.038	-0.024	-0.073	-0.073	Basic 3
GM_PSR	-0.044	0.0429	0.0429	1.5521	0.2862	1.5521	0.4628	-0.007	-0.048	-0.073	-0.073	MEAN
HON_PSR	-0.211	0.1284	0.0558	-0.14	-0.126	0.199	-0.126	7.9471	0.0396	-0.06	-0.211	Basic 3
HPQ_PSR	-0.38	0.0937	-0.036	-0.154	-0.207	0.0657	-0.158	2.6202	1.0441	-0.099	-0.38	Basic 3
IBM_PSR	-0.054	0.1525	0.2055	0.0597	0.0174	0.1102	0.0349	4.6284	3.1852	-0.057	-0.057	MEAN
JNJ_PSR	-0.11	0.3875	0.3457	0.1254	0.1679	1.0712	0.3007	0.1678	-0.102	0.012	-0.11	Basic 3
KO_PSR	-0.101	0.1376	0.3078	0.5186	0.3724	2.1592	0.6293	-0.047	0.0451	-0.068	-0.101	Basic 3
MMM_PSR	-0.053	0.1683	0.1164	0.2317	0.0809	1.0516	0.2176	-0.14	0.7291	0.0158	-0.14	GANN1
MRK_PSR	-0.182	0.2331	0.4133	0.5991	0.3754	0.5991	0.5071	-0.057	-0.057	-0.122	-0.182	Basic 3
PG_PSR	-0.226	0.1753	0.0457	0.1498	0.0524	1.0653	0.2522	0.9326	1.706	0.0288	-0.226	Basic 3
UTX_PSR	-0.313	0.0175	0.0853	-0.007	-0.018	0.9775	0.1217	3.5991	-0.323	-0.009	-0.323	GANN2
XOM_PSR	-0.326	-0.084	-0.013	-0.038	-0.13	0.4098	-0.054	1.412	0.6303	-0.064	-0.326	Basic 3

Table 6.4 Average Shape Ratio Errors of Basic 3 (AVAR) and Other Forecasting Models (Column) according to stock names (Row)

Table 6.4 shows the comparison among the average Sharpe Ratio errors resulting from AVAR or Basic 3 to those of the other forecasting models. The comparison shows that Basic 3 algorithm yields minimum average portfolio Sharp ratio error for 13 out of 17 stocks (AA, BA, CAT, DD, DIS, GE, HON, HPQ, JNJ, KO, MRK, PG and XOM), which is considered the best among other forecasting models. The next best is MEAN which yields minimum portfolio Sharp ratio error from 2 out of 17 stocks (GM and IBM).

Stock ID	Basic 1	Basic 2	Basic 3	ARMA	OLS	BPN1	BPN2	ELM1	ELM2	GANN1	GANN2	MEAN	Min	Best
AA_PSR	0.273	0.272	-0.08	0.401	0.327	0.649	0.561	1.851	0.832	15.004	6.2323	-0.03	-0.08	Basic 3
BA_PSR	8.351	7.268	-0.37	-0.15	-0.06	0.42	0.169	1.46	0.416	-0.039	0.6891	-0.09	-0.37	Basic 3
CAT_PSR	-0.37	-0.34	-0.46	-0.19	-0.1	-0.27	-0.3	0.44	-0.18	11.738	4.3998	0.051	-0.46	Basic 3
DD_PSR	-0.01	-0.02	-0.14	0.189	0.15	0.174	0.166	1.192	0.342	0.7865	-0.147	-0.04	-0.15	GANN2
DIS_PSR	0.292	0.294	-0.27	0.092	0.104	0.24	0.29	1.918	0.599	-0.1058	7.5523	-0.12	-0.27	Basic 3
GE_PSR	-0.06	-0.05	-0.07	0.157	0.265	0.313	0.132	0.601	0.202	-0.0384	-0.024	-0.07	-0.07	Basic 3
GM_PSR	-0.05	-0.04	-0.04	0.043	0.043	1.552	0.286	1.552	0.463	-0.0069	-0.048	-0.07	-0.07	MEAN
HON_PSR	0.017	0.022	-0.21	0.128	0.056	-0.14	-0.13	0.199	-0.13	7.9471	0.0396	-0.06	-0.21	Basic 3
HPQ_PSR	-0.15	-0.22	-0.38	0.094	-0.04	-0.15	-0.21	0.066	-0.16	2.6202	1.0441	-0.1	-0.38	Basic 3
IBM_PSR	-0.12	-0.11	-0.05	0.153	0.206	0.06	0.017	0.11	0.035	4.6284	3.1852	-0.06	-0.12	Basic 1
JNJ_PSR	-0.1	-0.08	-0.11	0.388	0.346	0.125	0.168	1.071	0.301	0.1678	-0.102	0.012	-0.11	Basic 3
KO_PSR	-0.08	-0.06	-0.1	0.138	0.308	0.519	0.372	2.159	0.629	-0.0467	0.0451	-0.07	-0.1	Basic 3
MMM_PSR	0.449	0.423	-0.05	0.168	0.116	0.232	0.081	1.052	0.218	-0.1403	0.7291	0.016	-0.14	GANN1
MRK_PSR	-0.19	-0.13	-0.18	0.233	0.413	0.599	0.375	0.599	0.507	-0.0565	-0.057	-0.12	-0.19	Basic1
PG_PSR	-0	0.061	-0.23	0.175	0.046	0.15	0.052	1.065	0.252	0.9326	1.706	0.029	-0.23	Basic 3
UTX_PSR	-0.32	-0.3	-0.31	0.017	0.085	-0.01	-0.02	0.977	0.122	3.5991	-0.323	-0.01	-0.32	GANN2
XOM_PSR	-0.32	-0.32	-0.33	-0.08	-0.01	-0.04	-0.13	0.41	-0.05	1.412	0.6303	-0.06	-0.33	Basic 3

Table 6.5 Average Shape Ratio Errors of Basic 1 (SBVBR), Basic 2 (MVBR), Basic 3 (AVAR) and Other Forecasting Models (Column) according to stock names (Row)

Comparing among the three proposed Fuzzy selection algorithms and with the other single forecasting classes of models, Basic 3 is the best forecasting algorithms so far. It yields minimum average portfolio Sharp ratio errors for 11 out of 17 stock instances (AA, BA, CAT, DIS, GE, HON, HPQ, JNJ, KO, PG and XOM). (Note that when comparing to other single forecasting models as reported in Table 6.4, Basic 3 is best for 13 out of 17, however, when we include Basic 1 and Basic 2, it is best for only 11 out of 17.) Basic 1 and GANN2 are considered the next best by beating 2 out of 17 stock instances equally (IBM and MRK for Basic 1; and DD and UTX for GANN2). The remaining stock instances are taken by MEAN (GM) and GANN1 (MMM) (see Table 6.5).

6.12 Summary

In this chapter, three model selection algorithms, namely Basic 1 (SBVBR), Basic 2 (MVBR), and Basic 3(AVAR) are introduced and compared with single classes model forecasting. Evaluation is performed by comparing average Sharp ratio errors of the three proposed algorithms of Fuzzy model selection with all single model forecasting. The results show that Basic 3 is the best selection algorithm both when compared with all single model forecasting and with the model selection algorithms combined. Basic 1 is the next best algorithm for out-of-sample forecasting of stock Sharp ratios. We intend to use the results in this chapter as inputs to our proposed optimisation module. The complete portfolio optimal system will be evaluated as a whole. Therefore, we have just roughly evaluated the result in this chapter.

Chapter 7 Portfolio Optimisation with Minimum Expected Errors Using Multi-Objective Genetic Algorithm

7.1 Chapter overview

Generally the inputs for the Markowitz models are expected or forecasted asset returns and asset risk measures. If all asset returns are assumed to follow a normal distribution with constant long term mean, the inputs for the optimisation problem are the constant long term means and standard deviations. However, if the normal distribution assumption is not true, especially for the short term, the means and standard deviations vary from time to time. In order to find a short term optimal portfolio, those expected returns and standard deviations are usually forecasted based on some mathematical or computational models which have previous values and some previous economic variables as independent variables that are always changing when the markets and the economies in question are moving. However, those forecasting models are far from perfect. In fact, even the best forecasting models in economics and finance are very inaccurate when compared to those of physical science and engineering. The inaccuracy of the forecasting models is a source of model risk. It also needs to cope and to take care of this risk in order to select a portfolio that is nearest to the optimal one. In this chapter as in the earlier Chapter 4, the research is based on a realistic case with all realistic constraints imposed, and we choose a genetic algorithm which is an approximate algorithm to solve the problem. Novel algorithms are proposed in such a way to optimise a portfolio while also taking care of model risk.

This chapter is organized as follows. After problems of Markowitz's portfolio optimisation model are described in Section 7.2, in Section 7.3, the concepts of model risk are briefly discussed. The Multi-Objective Genetic Algorithms, which is modified to use in this research, are described in Section 7.4. Section 7.5 provides the settings of the experimentation. The results of the experiment and their analyses are discussed in Section 7.6. Finally, a summary of finding is given in Section 7.7.

7.2 Portfolio Optimisation and Modern Portfolio Theory

Modern portfolio theory originated from Morkowitz's seminal paper (Markowitz 1952). The theory is based on economic theory that economic agents are, facing any economic decisions, rational and thus are trying to maximise their utility given budget constraints. The Markowitz Mean-Variance Model assumes that investors make their decisions in portfolio construction by choosing assets that maximise their portfolio return at the end of investment period (expected return). By assuming that investors are risk averse, the simplest model with a number of unrealistic constraints, namely perfect market without taxes, no transaction costs and infinitely divisible assets, the Markowitz portfolio optimisation can be stated mathematically as follows:

$$Min_{x_i}\sigma_p^2 \qquad (7.1)$$

Subject to

$$\begin{aligned} \sigma_p^2 &= \textstyle\sum_i \sum_j x_i\, x_j \sigma_{ij} \\ r_p &= r^* \\ r_p &= \textstyle\sum_i x_i\, r_i \\ \textstyle\sum_i x_i &= 1 \\ x_i &\geq 0 \end{aligned}$$

Where, σ_{ij} is covariance between asset i and j, if i = j, it is variance of asset i.

σ^2_P is variance of the portfolio of assets.

r_i is expected return of asset i.

r_P is the expected return of the portfolio

The Markowitz's model has many criticisms concerning its simplification for the sake of the ease of solving by simplifying representations of the problem. Besides imposing some unrealistic constraints and ignoring some realistic constraints, the objective function is considered unrealistic too. The portfolio optimisation objective is to find a combination of the least risky assets providing a level of portfolio return. To optimise a portfolio of assets, we find a combination of assets which minimise risk of a portfolio given an expected return. There are two problems of definitions here. First, how one can measure the "risk", since the original meaning of risk is subjective and depend on individual's risk appetite. Second, how can we estimate the expected return since all of returns of assets are to be realized in the future? This will lead to an important assumption of the model, whether expected return in the future is certainly known. By taking the estimated returns of assets for granted, one can optimise the model based on them. Alternatively, by assuming that the returns of assets in the future are not certain but follow some stochastic rule, the result will be a stochastic optimisation problem. Another aspect of modeling is whether the investor cares only for the next period or cares also for a number of consecutive outcomes in the future, or cares for any moment cumulatively from now to the end of investment period. The objective function will need to be adjusted for single period optimisation or for multi-period optimisation or dynamic optimisation accordingly. There are also some motivations to alter the objective function of portfolio optimisation problem for other purposes beside the aforementioned "theoretical" issues. The objective function may be modified for convenience of solving, such as to reduce complexity of computation, to make compatible with some known solving methods or algorithms, etc.

A crucial issue for portfolio optimisation is how one can find the model's inputs. There are two inputs, namely the measure of risk and the estimated asset return that must be estimated or forecasted, because according to the theory these inputs must represent the uncertain values and will be realised in the future. Optimising the portfolio selection model with error estimations of the inputs will lead to the wrong combination of assets and thus inefficient portfolio. This is model risk which this research attempts to handle.

7.3 Model Risk

There are many definitions of model risk around. As far as we are concerned, especially for this chapter, model risk is defined in a broad sense as follows: Model risk is the risk that one uses inaccurate models to make a decision or to assist in a process of making a decision, and by making such decision, leads to financial loss or misleads by miscalculation of the possibility of financial loss. For other definitions, Kato and Yoshiba (Kato 2000) defined model risk separately in the area of pricing model and of risk measurement model. In pricing models, model risk is defined as "the risk arising from the use of a model which cannot accurately evaluate market prices, or which is not a mainstream in the market." And in the risk measurement model, model risk is defined as "the risk of not accurately estimating the probability of future losses." In the same paper, the sources of model risk in pricing model are described as the use of wrong assumptions, errors in estimations of parameters, errors resulting from discretisation and error in market data. While the sources of model risk in risk measurement model are the difference between assumption and actual distribution and error in the logical framework of the model.

In another perspective, Derman (Derman 1996) classifies financial models into three categories, namely fundamental models, phenomenological models and statistical models. Different categories of models are prone to different sources of model risk. For the fundamental models which are mathematical models based on a set of postulates, the sources of model risk are wrong assumptions and wrong inputs. For the phenomenological models which are based on observations of the underlying behaviors, the main sources of risk are attempting to apply beyond their validity ranges and situations. Unlike both of aforementioned categories which embody some sort of causality, the statistical models rely on correlation rather than causation. The users of the statistical models hope that the correlations underpinning the models would eventually result from the cause and effect so that the correlations would be stable overtime. The main sources of model risk for this kind of models are misspecification, i.e. constructing models with the wrong variables or the wrong relationship functions, and instability of correlations which causes the model to be applicable for only a limited period of time.

In portfolio optimisation based on equation (7.1), it is crucial to accurately forecast portfolio return which can be calculated from returns of assets at the end of an investment period as well as portfolio standard deviation which can be calculated from standard deviations of the returns of assets during the period of investment. Note that the original Markowitz model uses means and standard deviations of assets returns as the best prediction of the future returns and volatilities of assets under the assumption that all assets' returns follow a certain statistical distribution, i.e. normal distribution. Most models used to forecast stock returns and standard deviations including that of past means and past standard deviations are a kind of the statistical models. Their predictability may suffer from the instability of correlations overtime. This kind of model risk in turn causes the outcomes of portfolio optimisation to be suboptimal, i.e. shortfalls in expected portfolio returns or larger volatilities than expected and thus lager probability of future losses.

There are a number of measures to manage or mitigate model risks. These measures are complimentary rather than substitutable. Firstly, models used in any decision making need to be reviewed by an independent model controller and reported to the management. Secondly, the persons who make uses of models need to be aware of the limitations of the models. Thirdly, models need to be thoroughly examined for their validity and limitations before being put into use. Fourthly, models in use are subjected to regular reviews (Kato 2000).

A measure is proposed to handle model risk by embedding stock selections based on predicted accuracy or validity of their forecasting models in the portfolio selection process. Forecasting models are mostly statistical ones which often suffer from their instability. To have a pool of models for the same subjects and dynamically evaluate and select the best among them is obviously a way to mitigate model risk. In this case, one has a number of somewhat substitutable assets (stocks) which have different expected returns, expected volatilities, and the capability to forecast them (i.e. model risk). As one puts the minimisation of model risk into the portfolio selection process, we could effectively handle model risk.

The objective is to find a portfolio that can handle model risk effectively. The proposed algorithm should also render a better portfolio, i.e. a portfolio with a larger Sharpe Ratio on average and for most of time. The aim is to construct a portfolio that optimizes two objectives simultaneously, i.e. minimisation of risk as measured by the portfolio's standard deviation and maximisation of the portfolio's return. This is actually to maximise Sharpe Ratio at a given level of returns or at a given level of risk.

7.4 Multi-objective Genetic Algorithms for Portfolio Optimisation

We deploy the Multi-Objective Genetic Algorithm for portfolio optimisation here, which is based on MOGA proposed by Fonseca and Flemming in 1993 (Fonseca 1993) as well as SPEA2 proposed by Zizler et al. (2001). MOGA relies on Pareto rankings to assign the smallest ranking value to all non-dominated individuals. On the other hand, for those dominated individuals, they are ranked by how many individuals in the population actually dominate them. Thus, the raw fitness of an individual is an inverse function of its Pareto rank. MOGA for two objectives portfolio optimisation is to rank individuals in the population by portfolio return (to maximise) and portfolio standard deviation (to minimise). To state by equations, the two objectives can be stated as follows

$$Max: \quad r_p = \sum_i^N x_i r_i \tag{7.2}$$

$$Min: \quad \sigma_p = \sum_i^N \sum_j^N x_i x_j \rho_{ij} \sigma_i \sigma_j \tag{7.3}$$

Where,

x_i is a proportion of the asset i in the portfolio of assets,

x_j is a proportion of the asset j in the portfolio of assets,

ρ_{ij} is the correlation coefficient between asset i and j,

σ_P is expected standard deviation of the portfolio of assets,

σ_i is expected or forecasted standard deviation of asset I,

σ_j is expected or forecasted standard deviation of asset j,

r_{pi} is the expected return of the portfolio of assets and

r_i is expected or forecasted return of asset i.

For the purpose of handling model risk from forecasting asset returns and standard deviations, the third objective is added into the original two-objective MOGA, and we referred to it as MOGA3O here. The third objective is based on the equation (6.14) and can be stated as follow:

$$Min: \quad SRE_p = \sum_i^N x_i (\frac{\partial ESR_p}{\partial x_i})^2 \tag{7.4}$$

Where, SRE_P is the approximated Sharpe ratio error of the portfolio of assets resulting from inclusion of all of the assets.

$\frac{\partial ESR_p}{\partial x_i}$ is the approximate impact to portfolio Sharpe ratio error of inclusion of asset i into the portfolio. The term is square to eliminate the sign (see equation 6.14).

In order to distribute the individual in the population evenly along the Pareto front, the overall fitness function is then adjusted by sum of sharing distance. The sharing distance between individuals i and j is given by:

$$SF_{ij} = 1 - \frac{d(x_i, x_j)}{\sigma_{share}}, \qquad if\ d(x_i, x_j) < \sigma_{share} \qquad (7.5)$$

$$SF_{ij} = 0, \qquad if\ d(x_i, x_j) \geq \sigma_{share}$$

Where, $d(x_i, x_j)$ is a metric distance between two individuals in objective domain,

σ_{share} is a predefined sharing distance. And, the overall fitness is defined by

$$F_i = \frac{Fit(i)}{\sum_j SF_{ij}} \qquad (7.6)$$

Where, fit (i) is the inverse of Pareto rank (i) (1/rank (i) in this test).

The overall fitness values of individuals are to be used in the probabilistic selection process by the comparative overall fitness to the individual that has maximum overall fitness. The comparative fitness values are used to compare with random number. If they exceed the random number, the individual will be selected (roulette selection method). MOGA usually has O (n^2) for a single round, because it needs to compute Pareto ranks and the sharing distance for all individuals. The pseudo code for MOGA is shown in Figure 4.4, Chapter 4.

```
// compute Pareto Rank for each port
for(Portfolio ith: pop) {     // for ith
int q = 1;
for(Portfolio jth: pop) {     // for jth
        if( (ith.getPortYield() <jth.getPortYield()) && (ith.getPortStd() >jth.getPortStd()
        &&(ith.getPortPSRError() >jth.getPortPSRError()))||
        (ith.getPortYield() <jth.getPortYield()) && (ith.getPortStd() >jth.getPortStd()
        && (ith.getPortPSRError() == jth.getPortPSRError()))||
        (ith.getPortYield() <jth.getPortYield()) && (ith.getPortStd() == jth.getPortStd()
        && (ith.getPortPSRError() >jth.getPortPSRError()))||
        (ith.getPortYield() == jth.getPortYield()) && (ith.getPortStd() >jth.getPortStd()
        && (ith.getPortPSRError() >jth.getPortPSRError()))||
        (ith.getPortYield() == jth.getPortYield()) && (ith.getPortStd() == jth.getPortStd()
        && (ith.getPortPSRError() >jth.getPortPSRError()))||
        (ith.getPortYield() == jth.getPortYield()) && (ith.getPortStd() >jth.getPortStd()
        && (ith.getPortPSRError() == jth.getPortPSRError()))||
        (ith.getPortYield() <jth.getPortYield()) && (ith.getPortStd() == jth.getPortStd() &&
        (ith.getPortPSRError() == jth.getPortPSRError())))
    q++;
    }     // for jth
ith.setParetoRank(q);
ith.setFitness((1/(double)q));
```

Figure 7.1: Calculation of Pareto Front Routine

While in SPEA2, the size of P' is fixed, the non-dominated individuals in a generation exceeding the size of P' will be removed. In contrast, if they are less than P' then some dominated individuals will be included in the archive P'. The exclusions and inclusions of dominated individuals are incorporated onto density information as a strategy to make the solutions distribute along the Pareto front. The density estimation of an individual i is defined as D (i) = $1/(d_i+2)$, where d_i is the distance of individual i from the nearest neighbor. SPEA2 deterministically selects all non-dominated individuals from the population P in the first round and then selects the combined population of P and archive of P' in the subsequent rounds. SPEA2 usually has O (N^2 log N) complexity in a single round, due to the density estimation calculation (Zitzler 2004.)

The problem is represented by hybrid encoding (Streichert 2004a, 2005). A pair of genetic strings stands for a particular portfolio (an individual of population) as discussed in Section 4.3 in Chapter 4. The binary value string represents which stocks (or assets) are included in portfolio (0 stands for not included and 1 stands for included). The real value string represents the weight of each stock in portfolio. So, the lengths of both strings are equal to the number of stocks in the market (or the stocks of interest.) The strings are generated with their real elements normalised in such a way that the summation of all elements of each combined string is always one. Before the repair algorithm begins, both strings combine by scalar product of the Binary string and the Real value string. Then after the repair process ends, the combined string is normalised to ensure that the summation of all elements is one. Finally, the combined string separates into new and normalized Binary string and Real value string (see Figure 4.1).

Crossover and mutation operations are performed independently for both strings. But before evaluation both strings need to be combined so that the objective values can be calculated. Crossover operation for all GAs is a three-point crossover by randomly selecting three points for the string independently. Mutation operation for all algorithms in this paper is a one-point mutation by randomly selecting the mutation point. For Binary strings, the mutation is a flip-flop mutation by changing from 1 to 0 and 0 to 1 respectively. For those of real value strings, mutation points are added by random numbers (between 0 and 1) multiplying by 0.1 (5% weight.)

The Markowitz model is a simplified model to focus only a theoretical point of view. In investment management practices, portfolio managers face a number of realistic constraints arising from normal business practices, practical matters, and industry regulations. The realistic constraints that are of practical importance include (not exhaustively) integer constraints, cardinality constraints, floor and ceiling constraints, turnover constraints, trading constraints, buy-in threshold, and transaction cost inclusions. Integer constraints or round lot constraints require the number of any asset included in the portfolio be an integer or be indivisible (i.e. cannot be in any fraction of normal trading lot). This may not suffer for GA optimisations because they are combinatorial but suffers for other optimisation methods that require continuity of the variables. The integer constraints (or round-lot constraints) can be expressed in equation (4.5) and (4.6). Cardinality constraints are the maximum number and minimum number of assets that a portfolio manager wishes to include in the portfolio due to monitoring reasons or diversification reasons or transaction cost control reasons (Stein 2005). The constraint is mathematically described in equation (4.7). Floor and ceiling constraints define lower and upper limits on the proportion of each asset which can be held in a portfolio. These constraints may result from institutional policy in order to diversify portfolio and to rule out negligible holding of assets for the ease of control (Crama 2003). They have been expressed mathematically as follows

$$f_i \leq x_i \leq c_i \forall i \qquad (7.7)$$

Where f_i and c_i are the lowest proportion and the highest proportion that asset i can be held in the portfolio respectively.

The repair algorithm first handles the cardinality constraints by setting smaller (S-K) values (from S values) of combined string to zero, where S is the number of selectable stocks (equal to the length of the strings) and K is the maximum number of stocks permitted in a portfolio (cardinality constraint.) Then, it handles floor constraint (buy-in threshold) by setting stocks whose weights are below the buy-in threshold to zero. Next, it normalizes those remaining non-zero weights to make all weight sum to 1 by setting $w_i^{/} = l_i + (w_i - l_i) / \Sigma(w_i - l_i)$, where w_i is non-zero weight of stock i and l_i is the buy-in threshold (the minimum weight amount that can be purchase) for stock i. Then, the round-lot constraints are handled by rounding the non-zero weights to the next round-lot level such that $w_i^{//} = w_i^{/} - (w_i^{/} \bmod c_i)$, where, c_i is the smallest volumes can be normally purchased from the stock market for stock i. The remainder of the rounding process ($\Sigma w_i^{/} \bmod c_i$) is allocated in quantity of c_i to $w_i^{//}$ which has the biggest value of $w_i^{/} \bmod c_i$ until all of the remainder is depleted.

All pairs of strings first are filled with random numbers, so, they need to be repaired by the repair algorithm. And since crossover and mutation operations cause the string to be deformed, the repair algorithm needs to be applied again to preserve the aforementioned constraints before the evaluations and selections.

7.5 The Experimental Design

An experiment has been conducted to find out whether the modification of MOGA and SPEA2 to include the third objective, portfolio sharp ratio error, is indeed for improving the outcome of actual portfolio in the term of actual portfolio sharp ratio. The experiment is to compare the actual portfolio among 5 cases for those of

MOGA group and 3 cases for those of SPEA2 group. The MOGA group consists of MOGA with two objectives (forecasted portfolio return and forecasted portfolio standard deviation) and Fuzzy MOGA (as MOGA_Fuz in Chapter 3) using stock forecasting by the Fuzzy selection algorithm Basic 3 in chapter 5, MOGA with two objectives using stock mean returns and standard deviations as stock forecasting values (MOGA_M), MOGAFuz with two objectives using stock mean returns and standard deviations as stock forecasting values (MOGAFuz_M) and MOGA with three objectives (forecasted portfolio return, forecasted portfolio standard deviation and estimated portfolio Sharpe ratio error-MOGA3O). The SPEA2 group consists of SPEA2 with two objectives, i.e. forecasted portfolio return and forecasted portfolio standard deviation from the BASIC 3 model selection algorithm from chapter 5 (SPEA2_MS), SPEA2 with two objectives using stock mean returns and standard deviations as stock forecasting values (SPEA2_MEAN), and SPEA2 with three objectives, i.e. forecasted portfolio return, forecasted portfolio standard deviation and estimated portfolio sharp ratio error (MOGA3O). The outcomes of 60 periods from 2002 (January) to 2006 (December) with 10 times for each period are evaluated and compared. All MOGAs and SPEA2s have 400 population and 1000 rounds. Note that we test the outcomes over a small set of data here, because we used most of the data of earlier periods to estimate parameters and train ANNs (see Chapter 5) as well as to build and parametise the selection algorithm (see Chapter 6). The estimate stock portfolio sharp ratio errors are calculated according to equation (6.14) and (7.4) to the next12 monthly observations of actual stock returns and standard deviations and 12 monthly out of sample forecasting of the corresponding forecasting models. For example, in the first year of 2002, the estimated models are used to forecast stock returns and deviations for the next 12 months of 2001. Then, the forecasted values with the actual values of the same periods are used to calculate the estimated stock portfolio sharp ratio errors by taking square of the numerical results of equation (7.4) and (7.8). The aforementioned estimated models and estimated portfolio sharp ratio errors are used for portfolio selections of 12 next monthly periods (2002). The forecasting models use actual month by month observations while the estimated stock portfolio sharp ratios error remains the same for all 12 month periods. For the next year, the window of observations will shift for one year to 2002 for the estimation of stock portfolio sharp ratio errors and to 2003 for portfolio selections and evaluations. The window of observations is rolling to the last year of 2006. For MOGA with two objectives using stock mean returns and standard deviations as stock forecasting values (MOGA_M), the previous 24 month data are used to calculate the stock mean returns and standard deviations.

For the reality constraints, the budget (i.e. the sum of money to invest) is set to USD 10,000,000.00. We need to set a budget because it is necessary to find how many stock the money can buy in order to handle the rounding constraint. The rounding lot for each stock is set to equal 10 stocks a lot, therefore the money amount of a rounding lot is 10 times stock closing price. The cardinality constraint, i.e. the maximum number of name of stocks in a portfolio, is assumed to be equal to 10. Finally, the floor weight (the minimum proportion of each stock allowed in a portfolio) is set to be equal to 0.01 (1%) and the ceiling weight (the maximum proportion of each stock allowed) equals to 0.25 (25%). Assuming that portfolio with sizable assets would be approximation of the market portfolio (under CAPM theory), we only estimate the values. For simplicity, the portfolio return is assumed to be equal to the average return of S&P 500 from 1980 to 1999 (as represent the market return) and set the portfolio standard deviation to that of S&P 500 returns from 1980 to 1999. The values are 0.011874 and 0.043027 accordingly and will be remain the same for all later periods. Again, for simplicity and estimation purposes, since the proportion of each stock in portfolio varies, the optimal portfolio is assumed to consist of equal proportion of 17 stocks, thus x_i is assumed to be 1/17 or 0.0588. These values are set constant for the entire experimental periods. One substitutes these values into equation (6.20) yielding the follow equation.

$$\frac{\partial ESR_p}{\partial x_i} = \frac{1}{0.043027}(R_i' - R_i) + \frac{0.011874}{0.00038402}(0.0588\sigma_i^2 - 0.0588\sigma_i'^2) \qquad (7.8)$$

7.6 Results and Analyses

In the experiments, we test MOGA and its modified versions as well as SPEA2 and its modified versions. We find the average of actual Sharpe Ratio between them to judge the algorithms' performance to obtain optimal choices of stock investments for one period (one month) ahead. There are 10 samples per one month period for 60 months so there are totally 600 samples in each panel outcome data set.

Period	MOGA3O	MOGA	MOGAFuz	MOGA_M	MOGAFuz_M	Period	MOGA3O	MOGA	MOGA_Fuz	MOGA_M	MOGAFuz_M
1	-10.7	10.936	-11.1903	15.558	16.2272	**31**	8.6468	7.3917	7.39173	12.037	5.578077
2	8.2047	2.5877	9.399086	14.721	12.07396	**32**	9.6496	8.2099	8.2099	-2.457	-2.25004
3	4.9353	-2.559	4.828431	2.1185	-8.72976	**33**	-0.354	-5.724	-5.724	-5.363	-2.14992
4	-4.069	-1.832	-4.93287	-4.689	-4.04255	**34**	19.833	19.523	19.5227	21.822	21.15358
5	-4.423	-6.995	-4.13416	-17.03	-18.0214	**35**	-0.849	2.0407	2.04071	11.001	7.177073
6	-5.947	-8.078	-5.21733	-4.794	-9.47081	**36**	-14.53	-16.95	-16.952	-1.082	-2.36134
7	0.1723	-1.37	-0.95915	-0.981	-0.20795	**37**	11.028	11.825	11.8248	39.85	28.85629
8	6.6373	-10.98	8.323575	-8.354	-7.2497	**38**	-5.726	-9.742	-9.7418	-2.903	-2.24428
9	12.629	9.32	11.09712	6.7079	4.75702	**39**	-11.87	-11.98	-11.978	-12.07	-5.16499
10	4.3628	4.7937	5.539727	0.1361	1.330263	**40**	4.0427	10.292	10.2925	2.1794	3.903933
11	3.2142	-5.132	3.346566	-5.504	-3.95819	**41**	-3.625	-3.13	-3.1303	-7.944	-1.01682
12	15.253	-4.399	9.059314	-9.246	-5.34831	**42**	5.9609	14	13.9996	1.9853	3.597757
13	-7.781	-5.455	-14.1861	-3.23	-2.32542	**43**	-7.735	-0.079	-0.0794	2.3364	5.164895
14	25.658	-0.912	29.45351	16.827	10.82595	**44**	16.373	10.793	10.7926	9.6009	7.715435
15	-2.709	7.4125	-1.58568	3.4126	4.652872	**45**	-12.29	-13.28	-13.284	-21.77	-19.4714
16	45.757	11.095	23.73846	3.2743	7.510827	**46**	7.8234	9.9439	9.94386	7.0949	6.978559
17	3.1531	4.8879	4.952013	1.7062	-0.88106	**47**	2.6063	-3.967	-3.9673	-9.605	0.179903
18	7.7796	3.3308	6.783529	7.7091	5.106529	**48**	15.502	21.29	21.2901	11.384	23.12005
19	-0.718	9.2552	-1.29663	1.0517	1.829317	**49**	8.4848	8.213	8.21299	10.601	6.173813
20	-5.723	-14.17	-9.64393	16.407	-6.39844	**50**	8.6246	7.1372	7.13717	0.67	3.172302
21	-0.122	29.745	2.569789	13.997	13.99057	**51**	23.064	17.052	17.052	16.219	16.27077
22	9.7297	-0.974	9.783416	1.7364	2.986315	**52**	-1.455	-7.243	-7.2434	-4.275	-2.95829
23	-6.16	27.604	-9.58369	44.56	39.3206	**53**	-1.2	-0.641	-0.6414	-1.563	-0.30061
24	8.6486	4.8403	7.342368	-12.63	-7.10473	**54**	-3.273	-5.136	-5.1357	0.723	-2.44524
25	9.6471	6.7889	8.14987	-0.977	2.310339	**55**	7.0328	6.3053	6.30527	7.239	1.561448
26	-0.32	-1.177	-5.98195	-5.622	-10.8787	**56**	9.6689	7.1659	7.16594	5.2409	4.174184
27	19.83	-8.786	19.56304	-0.485	-1.93078	**57**	13.367	9.9828	9.98281	32.554	16.87682
28	0.1832	2.9344	1.800028	0.4352	4.248483	**58**	4.5668	3.1056	3.10556	2.4822	11.06094
29	-11.51	9.8012	-17.208	13.978	14.59579	**59**	-2.23	1.0861	1.08607	10.109	2.758595
30	11.574	-9.604	11.91136	-20.8	-7.56033	**60**	28.52	25.028	25.0275	7.9946	3.895983

Table 7.1: Average portfolio sharp ratios of the outcomes of MOGA and its modified versions.

Table 7.1 shows the average of portfolio sharp ratios of all 10 samples of selected portfolio from using different algorithms, namely, MOGA with 2 objectives (portfolio returns and portfolio standard deviation) using the stock forecasting by Basic 3 (MOGA), Fuzzy MOGA with 2 objectives using the stock forecasting by Basic 3 (MOGAFuz), Multi-objective genetic algorithm with 2 objectives using means and standard deviations of stocks as forecasting values (MOGA_M), MOGA with 2 objectives using means and standard deviations of stocks as forecasting values

(MOGAFuz_M) and MOGA3O with 3 objectives in which the third objective, the estimated portfolio sharp ratio error, is incorporated into MOGA. Figure 7.2 shows graphical decription of the average portfolio sharp ratios from Table 7.1. The horizontal axis is the number of period (1 – 60) and the vertical axis is the value of portfilio sharp ratio.

Period	SPEA2_3O	SPEA2_MS	SPEA2_MEAN	Period	SPEA2_3O	SPEA2_MS	SPEA2_MEAN
1	11.41458	12.88161	2.655231	31	8.638711	7.794607	0.497722
2	3.453195	3.40676	-15.8484	32	9.70863	7.752666	-5.23613
3	-3.29575	-1.52629	8.350613	33	-3.54513	-4.25042	17.64804
4	0.12467	-0.35849	1.712594	34	19.79139	19.27034	3.351981
5	-7.07564	-6.70003	-11.8284	35	1.16427	1.818865	40.73078
6	-7.36537	-9.75236	6.724451	36	-12.5429	-19.2626	-13.1963
7	-0.95063	-1.46031	-5.8192	37	13.94309	11.94271	-3.05708
8	-10.7036	-11.1338	-7.08724	38	-3.61738	-9.99209	-4.77081
9	8.26454	9.035181	1.629333	39	-11.6848	-12.3176	-1.89881
10	4.956189	4.896407	8.274699	40	10.24371	9.291814	0.189116
11	-4.12682	-7.53972	6.919482	41	0.511999	-2.40561	14.81651
12	-4.4144	-4.12899	-6.26818	42	7.835436	15.11159	-18.4152
13	-6.60596	-5.53776	15.29978	43	-3.53328	-2.16439	12.12039
14	1.366036	0.314661	11.32056	44	13.27279	13.2585	-2.85351
15	7.41045	7.966957	3.164859	45	-14.047	-14.5559	-7.70022
16	12.71714	12.15893	-4.95533	46	8.447448	10.2491	21.60546
17	4.828903	5.739117	-15.922	47	0.554994	-6.48217	11.46286
18	2.884883	3.139503	-5.48336	48	15.65134	18.00374	-0.6437
19	2.975822	9.458178	-0.31595	49	10.15464	3.989328	37.58161
20	-10.8415	-12.2569	-8.64287	50	6.674794	6.345272	-4.2084
21	26.50963	30.52927	5.743597	51	19.08311	19.75499	-10.3674
22	-2.72743	-2.13708	0.896455	52	-1.91937	-3.58384	1.908551
23	40.17496	30.73534	-6.46243	53	-1.76723	-0.79414	-3.47836
24	6.099512	5.023068	-8.70226	54	-3.04296	-4.91734	2.853264
25	7.779155	4.229403	-3.0615	55	4.780385	6.830131	4.638343
26	0.153657	-1.19437	16.1271	56	9.551172	7.722105	9.770464
27	-2.75949	-7.85417	3.874901	57	8.256564	15.64671	-23.9768
28	0.1706	4.031284	2.785285	58	1.274766	2.486074	7.011145
29	16.41433	10.06229	1.767675	59	-2.15096	1.434787	-8.56114
30	-9.20117	-9.45243	9.953307	60	25.34843	24.15209	13.5576

Table 7.2: Average portfolio sharp ratios of the outcomes of SPEA2 and its modified versions.

Table 7.2 shows the average of portfolio sharp ratios of all 10 samples of selected portfolio from the SPEA2 and its modifications, namely SPEA2_MS with 2 objectives (portfolio returns and portfolio standard deviation) using the stock forecasting by Basic 3 (MOGA), SPEA2_MEAN with 2 objectives using means and standard deviations of stocks as forecasting values, SPEA2 with 3 objectives (SPEA2_3O) in which the third objective, the estimated portfolio Sharpe ratio error, is incorporated into SPEA2. Figure 7.3 shows graphical decription of the average portfolio Sharpe ratios from Table 7.2. The horizontal axis is the number of period (1 – 60) and the vertical axis is the value of portfilio Sharpe ratio.

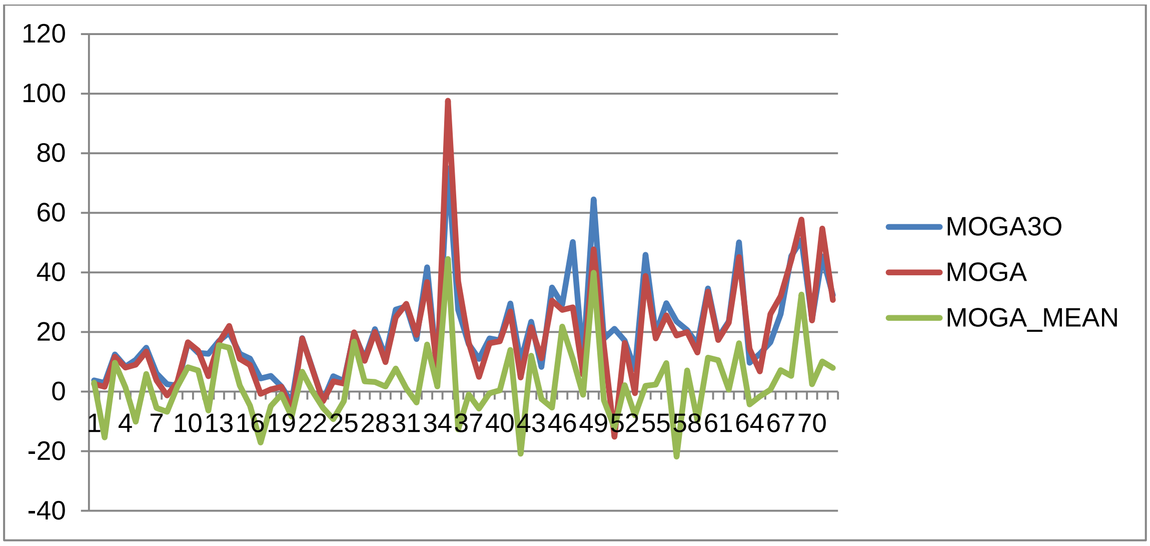


Figure 7.2Average portfolio sharp ratios of the outcomes of MOGA and its modified versions (Y axis shows sharp ratio values and X axis shows monthly period from 2002-2006)

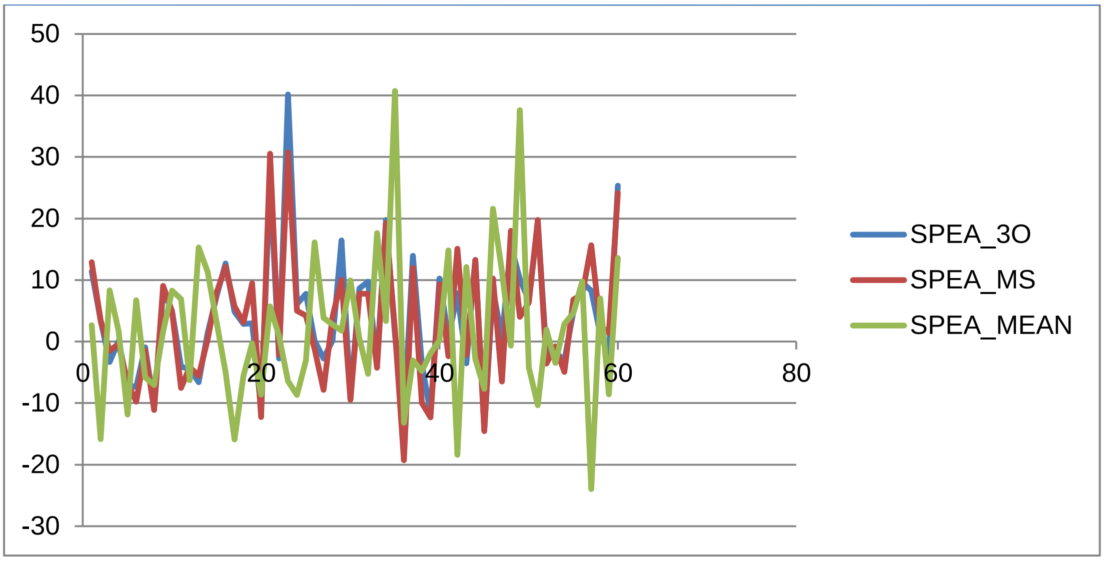


Figure 7.3: Average portfolio sharp ratios of the outcomes of SPEA2 and its modified versions (Y axis shows Sharpe ratio values and X axis shows monthly period from 2002-2006)

Next, it is to find from the results, whether handling model risk by incorporating the third objective, the approximation of portfolio sharp ratio error which is estimated from stock impacts on portfolio sharp ratio errors, will indeed improve the actual outcomes of portfolios at the end of each investment periods. There are the results from 60 periods, 10 samples for each period, thus 600 polled samples in totality. To see whether the algorithms with 3 objectives are better off than those of 2 objectives, we perform testing hypotheses about a proportion (Utts 2006). Since we consider two classes of Multi-objective Genetic Algorithms, MOGA and its modifications as well as SPEA2 and its modifications need to be evaluated.

For the testing hypotheses about a proportion, we need to know that, at a given level of confidence, whether the outcomes of MOGA3O and SPEA2_3O are better off those of MOGA, MOGAFuz, MOGA_MEAN or MOGAFuz_M and SPEA2_MS or SPEA2_MEAN respectively for the most of time. If the outcomes of MOGA3O/SPEA2_3O are not better off for most of the time the proportion of samples in which MOGA3O/SPEA2_3O beats those of MOGA, MOGAFuz, MOGA_MEAN or MOGAFuz_M and SPEA2_MS or SPEA2_MEAN would be ranged from less than to around 50:50 percent or 0.5, else the proportion would be significantly greater than 0.5. These tests are not concerning the means of portfolio sharp ratio values but only the number of samples where MOGA3O/SPEA3O's outcomes are better than those of MOGA or MOGA_MEAN and SPEA2_MS or SPEA2_MEAN. The appropriate significant test is as follows:

$$H_0:\ p \leq\ p_0$$

$$H_1:\ p > p_0$$

$$z = \frac{\bar{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}} \qquad (7.9)$$

Where, z is z score of the statistical test,

$\overline{p}$ is the sample estimate of population proportion (in this case, is that of MOGA3O or SPEA2_3O beating),

$p_0$is the hypothetical value to test against (in this case, is 0.5),

$n$is the number of the second population.

Algorithms	n	$\overline{x}$	> MOGA3O	> MOGA	>MOGAFuz	>MOGA_M	> MOGAFuz_M
MOGA3O	600	3.78571	0 (0.00%)	**354** **(59.00%)**	**346** **(57.67%)**	**333** **(55.50%)**	**316** **(52.67%)**
MOGA	600	3.10545	246 (41.00%)	0 (0.00%)	**314** **(52.33%)**	**315** **(52.50%)**	284 (47.33%)
MOGAFuz	600	2.93361	254 (42.33)	286 (47.67%)	0 (0.00%)	298 (49.67%)	283 (47.12%)
MOGA_M	600	3.23048	267 (44.50%)	285 (47.50%)	**302** **(50.33%)**	0 (0.00%)	279 (46.50%)
MOGAFuz_M	600	3.11109	284 (47.33)	**316** **(52.67%)**	**317** **(52.83%)**	**321** **(53.50%)**	0 (0.00%)

Table 7.3 Summary of important statistical values MOGA and its modifications

Table 7.3 summarizes important statistics for the test according to equations (7.9). Obviously, the outcomes of MOGA3O are comparatively better than those of MOGA, MOGAFuz, MOGA_M and MOGAFuz_M. MOGA3O's sharp ratios have an average of 3.78571 while those of MOGA, MOGAFuz, MOGA_M and MOGAFuz_M are 3.10545, 2.93361, 3.23048 and 3.11109 respectively.

MOGA3O > MOGA	**t/z**	**One-tail prob.**	**Reject Ho (at % level)**
$H_0:\ p \leq\ p_0$ $H_1:\ p > p_0$	z = 4.40908	5.1905E-06	Yes (0.1% level)

Table 7.4 Result of the proportional test whether MOGA3O's outcome is better than that of MOGA

Table 7.4 concludes the result of the proportional test of equation (7.9) comparing the outcomes of MOGA3O and of MOGA. The calculated z-score is

4.40908. With reference to the normal distribution table, the null hypothesis that the proportion of the outcomes of MOGA3O is worse than or equal to those of MOGA is rejected at the 0.1 % level of confidence (with the probability that one wrongly rejected the hypothesis less than 1 out of 1000 times). Thus, one can safely conclude that the outcomes of MOGA3O, indeed, are better than those of MOGA for most of the time.

MOGA3O > MOGAFuz	**t/z**	**One-tail prob.**	**Reject Ho (at % level)**
$H_0:\ p \leq\ p_0$ $H_1:\ p > p_0$	z =3.75588	8.6365E-05	Yes (0.1% level)

Table 7.5 Result of the proportional test whether MOGA3O's outcome is better than that of MOGAFuz

Table 7.5 shows the results of the proportional test of equation (7.9) comparing the outcomes of MOGA3O and of MOGAFuz. The calculated z-score is 3.75588. Referencing to the normal distribution table, the null hypothesis that the proportion of the outcomes of MOGA3O is worse than or equal to those of MOGAFuz is rejected at the 0.1 % level of confidence. Therefore, it safely concludes that the outcomes of MOGA3O, indeed, are better than those of MOGAFuz for most of the time.

MOGA3O> **MOGA_MEAN**	**t/z**	**One-tail prob.**	**Reject Ho (at % level)**
H_0: $p \leq p_0$ H_1: $p > p_0$	z = 2.69444	0.003525	Yes (0.5% level)

Table 7.6 Result of the proportional test whether MOGA3O's outcome is better than that of MOGA_M

Table 7.6 concludes the result of the proportional test comparing the outcomes of MOGA3O and of MOGA_M. The calculated z-score is 2.69444. With reference to the normal distribution table, the null hypothesis that the proportion of the outcomes of MOGA_M is worse than or equal to those of MOGA3O is rejected at the 0.1 % level of confidence. Thus, it can safely conclude that the outcomes of MOGA3O, indeed, are better than those of MOGA_MEAN for most of the time.

MOGA3O> **MOGAFuz_M**	**t/z**	**One-tail prob.**	**Reject Ho (at % level)**
H_0: $p \leq p_0$ H_1: $p > p_0$	z = 1.30639	0.095709	Yes (10% level)

Table 7.7 Result of the proportinal test whether MOGA3O's outcome is better than that of MOGAFuz_M

Table 7.7 concludes the result of the proportional test comparing the outcomes of MOGA3O and of MOGAFuz_M. The calculated z-score is 1.30639. With reference to the normal distribution table, the null hypothesis that the proportion of the outcomes of MOGAFuz_M is worse than or equal to those of MOGA3O is rejected at the 10 % level of confidence. Thus, one can conclude that the outcomes of MOGA3O are better than those of MOGAFuz_M for most of the time.

Algorithms	**n**	$\overline{x}$	**> SPEA2_3O**	**> SPEA2_MS**	**>SPEA2_MEAN**
SPEA2_3O	600	3.744453402	0 (0.00%)	**332 (55.33%)**	**333 (55.50%)**
SPEA2_MS	600	3.245078749	268 (44.67%)	0 (0.00%)	**320 (53.33%)**
SPEA2_MEAN	600	1.636378534	267 (44.50%)	280 (46.67%)	0 (0.00%)

Table 7.8 Summary of important statistical values of the SPEA2 and its modifications

Table 7.8 summarizes important statistics for the test according to equations (7.9) for SPEA2 and its modifications. As we can see, the outcomes of SPEA2_3O are comparatively better than those of SPEA2_MS and SPEA2_MEAN. SPEA2_3O's sharp ratios have an average of 3.74445 while those of SPEA2_MS and SPEA2_MEAN are 3.24508 and 1.63638 respectively.

SPEA2_3O > SPEA2_MS	**t/z**	**One-tail prob.**	**Reject Ho (at % level)**
$H_0: \ p \leq \ p_0$ $H_1: \ p > p_0$	z = 2.612789059	0.004490336	Yes (0.5% level)

Table 7.9 Result of the proportional test whether SPEA2_3O's outcome is better than that of SPEA2_MS

Table 7.9 concludes the result of the proportional test of equation (7.9) comparing the outcomes of SPEA2_3O and of SPEA2_MS. The calculated z-score is 2.612789. With reference to the normal distribution table, the null hypothesis that the proportion of the outcomes of SPEA2_3O is worse than or equal to those of SPEA2_MS is rejected at the 0.5 % level of confidence (with the probability that one wrongly rejected the hypothesis less than 5 out of 1000 times). Thus, one can safely conclude that the outcomes of SPEA2_3O, are indeed, better than those of SPEA2_MS for most of the time.

SPEA2_3O > SPEA3O_MEAN	**t/z**	**One-tail prob.**	**Reject Ho (at % level)**
$H_0\colon\ p \leq\ p_0$ $H_1\colon\ p > p_0$	z = 2.69444	0.003525	Yes (0.5% level)

Table 7.10 Result of the proportional test whether SPEA2_3O's outcome is better than that of SPEA2_MEAN

Table 7.10 shows the result of the proportional test of equation (7.9) comparing the outcomes of SPEA2_3O and of SPEA2_MEAN. For the test hypotheses about a proportion described in the first row, the calculated z-score is 2.69444. Referencing to the normal distribution table, the null hypothesis that the proportion of the outcomes of SPEA2_3O is worse or equal to those of SPEA2_MEAN is rejected at the 0.5 % level of confidence. Therefore, it safely concludes that the outcomes of SPEA2_3O, indeed, are better than those of SPEA2_MEAN for most of the time.

Algorithms	**n**	$\bar{x}$	**> SPEA2_3O**	**> MOGA3O**
SPEA2_3O	600	3.74445	0 (0.00%)	**304 (50.67%)**
MOGA3O	600	3.78571	296 (49.33%)	0 (0.00%)

Table 7.11 Summary of important statistical values comparing between SPEA2_3O and MOGA3O

Table 7.11 summarizes important statistics for the test according to equations (7.9) for comparing between SPEA2_3O and MOGA3O. As we can see, the outcomes of SPEA2_3O are comparatively on par with that of MOGA3O. SPEA2_3O's sharp ratios have an average of 3.74445 while that of MOGA3O is and 3.78571. While only 304 periods, only slightly more than half, in which SPEA2_3O's sharp ratios are better than those of MOGA3O.

SPEA2_3O> MOGA3O	t/z	One-tail prob.	Reject Ho (at % level)
$H_0: p \leq p_0$ $H_1: p > p_0$	z = 0.326598	0.371986	No

Table 7.12 Result of the proportional test whether SPEA2_3O's outcome is better than that of MOGA3O

Table 7.12 shows the result of the proportional test comparing the outcomes of SPEA2_3O and of MOGA3O. The calculated z-score is 0.326598. With reference to the normal distribution table, the null hypothesis that the proportion of the outcomes of SPEA2_3O is worse than or equal to those of MOGA3O cannot be rejected at any level of confidence. Thus, we can conclude that the performance of SPEA2_3O and MOGA3O is equal.

7.7 Summary

In this chapter, modified versions of two classes of Multi-Objective Genetic Algorithm namely MOGA and SPEA2 are proposed to handle model risk in portfolio optimisation problem with realistic constraints. An experiment has been conducted for the proposed algorithm with real data from US stock market over 60 short term investment periods. The results of the proposed algorithms (MOGA3O and SPEA2_3O) are compared with those of original two-objective optimisation (MOGA, MOGAFuz, MOGA_M, MOGAFuz_M and SPEA2_MS, SPEA2_MEAN respectively). The results show that the outcomes of MOGA3O are comparatively the best in its group for most of time periods. The similar results also can be concluded for SPEA2_3O in its group. Statistical tests of proportion confirm this conclusion. Incorporating a third objective, the approximation of portfolio sharp ratio error, helps to reduce the inherent model risk in the forecasting process. By comparing the performance between MOGA3O and SPEA2_3O, we found that their performances are on par to each other. MOGA3O and SPEA2_3O optimize portfolio selections by inclining to choose more proportions of stocks which are able to forecast their returns and volatilities more accurately given their returns and volatilities are equal and with more accurate forecasted inputs can render superior stock portfolio selections for a single period investment.

Chapter 8 Conclusions and Future Works

8.1 Conclusion

Portfolio optimisation is always in a dynamic domain since information and factors that affect optimisation are changing over time. An effective portfolio optimisation system needs to be adaptive. An adaptive portfolio optimisation system consists of three parts working cooperatively, namely the optimisation part, the predictive part, and the adaptability part.

Portfolio managers face a number of realistic constraints resulting from normal business practices, practical matters, and industry regulations. The realistic constraints of practical importance are integer constraints, cardinality constraints, floor and ceiling constraints, turnover constraints, trading constraints, buy-in threshold, and transaction cost inclusions. The closed-form mathematical solution of portfolio optimisation problem usually cannot include these realistic constraints. The problem needs to be solved by numerical methods. However, solving them by exhaustive searches normally takes too much, if not prohibitive, computational time. Thus, heuristic searches are more appropriate.

In this thesis, multi-objective genetic algorithms are proposed to serve in the optimisation module. A number of Multi-Objective Genetic Algorithms are explored to solve the problem with three common realistic constraints (namely cardinality constraints, floor constraints, and round-lot constraints). Fuzzy logic is incorporated to see whether it can improve the performances of the Vector Evaluated Genetic

Algorithm (VEGA). The results show that using fuzzy logic to combine optimisation objectives of VEGA (in VEGA_Fuz1) for this problem does improve performances especially in Generation Distance from the true Pareto front, but its solutions tend to cluster around a few points. With additional fuzzy logic to make VEGA solution more distributed, the performance is worsened. MOGA and SPEA2 are more complex algorithms but they perform better. SPEA2 is rendered more evenly distributed portfolio along the effective front. However, MOGA is more time efficient for solving portfolio optimisation with realistic constraints.

For the predictive part, two classes of traditional econometric models, i.e. Auto-Regressive-Moving-Average models (ARMA) and multivariate Ordinary Least Square (OLS) are used as benchmark in comparison with Back-Propagation ANN, Elman Recurrent ANN and the novel proposed hybrid ANNs. ANNs have potentials to make a better forecasting of financial and economics time series. The novel proposed hybrid evolutionary ANNs (EANNs) go a step further to automatically evolve both initial and structures (number of hidden nodes and number of hidden layers). Compared with the traditional OLS, the ANNs show promising results and most of the proposed EANNs can improve the performances of the ANNs as expected even if the improvements are slights. Although ANNs have the potential to make a better forecasting of financial and economics time-series, selecting the appropriate structure and relevant input variables by using Evolutionary Algorithm techniques that are quite time consuming.

For the adaptive part, three Fuzzy adaptive model selection algorithms are explore namely Basic 1 (SBVBR), Basic 2 (MVBR) and Basic 3(AVAR). Their predictive performances are compared with those of single classes model forecasting (ARMA, OLS, BPN1, BPN2, ELM1, ELM2, GANN1, GANN2 and MEAN).

Their performances are evaluated by comparing average Sharp ratio errors of the three proposed algorithms of Fuzzy model selection with all single model forecasting. The results show that Basic 3 is the best selection algorithm comparing with all single model forecasting and the model selection algorithms combined. Basic 1 is the second best algorithm for out-of-sample forecasting of stock Sharp ratios.

As a complimentary adaptive mechanism, a modified version of Multi-Objective Genetic Algorithm (MOGA3O) is proposed to handle model risk in portfolio optimisation problem with realistic constraints. An experiment is conducted with real data from US stock market over 60 short term investment periods. The results with those of MOGA and MOGA with Fuzzy optimisation and also with MOGA with using medium terms stock mean returns and standard deviations (MOGA_M) as well as MOGA_M with Fuzzy optimisation. The outcomes of MOGA3O are comparatively the best of all five algorithms, and statistical tests confirm this conclusion. By incorporating a third objective, the approximation of portfolio sharp ratio error is reduced as well as the inherent model risk in the forecasting process. MOGA3O adaptively optimise portfolio selections by choosing more proportions of stocks whose returns and volatilities can be predicted more accurately given the same returns and volatilities. Then, SPEA2 which is considered the best algorithm regarding to its closeness to the real Pareto front and its distribution of solutions along the Pareto front. SPEA2 is modified such that it incorporates the third objective as MOGA3O does to handle input model/estimation risk which is denoted SPEA2_3O. The modified algorithm is compared with the original version of SPEA2 with using forecasts from the fuzzy model selection algorithm and with using historical means and standard deviations as the inputs respectively (denoted SPEA2_MS and SPEA2_MEAN respectively). The result for this group is quite similar for those of MOGA. The performance order is SPEA2_3O, SPEA2_MS and SPEA_MEAN. However, when the performance of SPEA2_3O is compared with that of MOGA_3O, they are quite on par and with statistical tests, it can be concluded that no one performs better than the other.

8.2 Future Works

Some directions for future works that can be built from the research outcomes presented in this thesis are discussed in the following paragraphs.

An aspect that is worth further research is forecasting stock correlations. In this thesis we assume stock correlations are constant overtime and for entire periods of studies. If the forecasting of correlations is included then the portfolio selection would more realistic and may reduce errors.

There is also ample room for further researches in more accurate prediction models. Large amount of research has been done through many decades in this area. But it is still inadequate. Also, the running time is quite long for some of the proposed predicting models. Due to limited computer power, experiments with many populations and many generations have not been possible. Running the experiments in parallel high performance computer may provide some more improvements. The EANN algorithm also can be modified for further improvements such as introducing more variations of ANNs or selection of inputs. The application of the EANN to other related forecasting problems such as to predict stock volatilities, exchange rates, etc., is quite a natural extension of this research.

The proposed Fuzzy model selection algorithms are open module in such a way that they can incorporate more forecasting models into them. If a number of better forecasting models are included, they can yield more accurate predictions, hence more efficient selected portfolios. Experiment with more complex Fuzzy rules is also worth trying.

For the proposed three objective portfolio optimisations, MOGA3O and SPEA2_3O, one could also modify other multi-objective algorithms such as VEGA, VEGA_Fuzzy or MOGA_Fuz to incorporate the third objective. The results should be compared among them to see which one is yielding a better result.

It would also be interesting to use the Multi-objective Genetic Algorithm platform to explore other optimisation objectives such as the third moment of distribution (skewness), the fourth moments (kurtosis) or even to explore with the other risk measures such as VaR, CVaR, expected shortfall, etc., to see whether the ex-post or out-of-sample performances could be improved compared with the proposed algorithms in this thesis.

References

(Abdelazim 2006) Abdelazim, H.Y., and Wahba, K. (2006) ‘An artificial intelligence approach to portfolio selection and management.’ *International Journal of Financial Services Management, 1(2),* pp. 243-254.

(Alba 2005) Alba, E., Talbi, E., Luque, G., and Melab, N. (2005) ‘Metaheuristic and Parallelism’ in Alba, E. (ed.) *Parallel Metaheuristic: A New Class of Algorithms*. Hoboken: John Wiley and Sons.

(Armananzas 2005) Armananzas, R., and Lozano, J.A. (2005) ‘A multiobjective approach to the portfolio optimisation problem Evolutionary Computation.’ *Proceedings of IEEE Congress on Evolutionary Computation,* vol. 2, IEEE Publications, pp. 1388-1395.

(Anagnostopoulos 2010) Anagnostopoulos, K.P., and Mamanis, G. (2010) ‘A Portfolio Optimisation Models with Three Objectives and Discrete Variables.’ *Computer and Operation Research,* 37(7), pp. 1388-1395.

(Anenc 2003) Amenc, N., and Le Sourd, V. (2003) *Portfolio Theory and Performance Analysis.* Chichester: JohnWiley and Sons.

(Armano 2005) Armano, G., M. Marchesi, M., and Murru, A. (2005) ‘A hybrid genetic-neural architecture for stock indexes forecasting.’ *Information Sciences, 170,* pp. 3-33.

(Arnone 1993) Arnone, S., Loraschi, A., and Tettamanzi, A. (1993) ‘A genetic approach to portfolio selection.’ *Journal on Neural and Mass-Parallel Computing and Information Systems, 3*, pp. 597-604.

(Asteriou 2007) Asteriou, D., and Hall, S.G. (2007) *Applied Econometrics: A Modern Approach*. revised edn. New York: Palgrave Macmillan.

(Avouyi-Dovi 2004) Avouyi-Dovi, S., Morin, S., and David, N. (2004) 'Optimal Asset Allocation with Omega Function.' Available at: http://crem.univ-rennes1.fr/site_francais/seminaires/textesseminaires/textes_seminaires_2004/sadovi_6_04.pdf (Assessed: 6 December 2011).

(Bienstock1996) Bienstock, D. (1996) 'Computational study of a family of mixed-integer quadratic programming problems.' *Mathematical Programming,* 74, pp. 121-140.

(Black 1972) Black, F., Jensen, M.C., and Scholes, M. (1972) 'The Capital Asset Pricing Model: Some Empirical Tests.' in Jensen, M.C. (ed.) *Studies of the Theory of Capital Markets.* New York: Praeger Publishers.

(Blum 2003) Blum C., and Roli, A. (2003) 'Metaheuristic in Combinatorial Optimisation: Overview and Conceptual Comparison.' *Computing Surveys,* 35, pp. 268-308.

(Blume 1975) Blume, M.E., and Friend, I. (1975) 'The Asset Structure of Individual Portfolio and Some Implications for Utility Functions.' *The Journal of Finance,* 30(2), pp. 585-603.

(Box 1976) Box, G.E.P., and Jenkins, G.M. (1976*) Time Series Analysis: Forecasting and Control*. revised edn. San Francisco: Holden-Day.

(Briec 2007) Briec, W.K., Kerstens, K. and Jokung, O. (2007) 'Mean-Variance-Skewness Portfolio Performance Gauging: A General Shortage Function and Dual Approach.' *Management Science, 53(1),* pp. 135-149.

(Busetti 2000) Busetti, F.R. (2000) *Metaheuristic Approaches to Realistic Portfolio Optimisation.* MSc thesis, University of South Africa.

(Chang 2000) T.J. Chang, T.J., Meade, N., Beasley, J.E., and Sharaiha, Y.M. (2000) 'Heuristics for Cardinality Constrained Portfolio Optimisation.' *Computers and Operations Research,* 27, pp. 1271-1302.

(Chekhlov 2003a) Chekhlov, A., Uryasev, S., and Zabarankin, M. (2003) 'Portfolio Optimisation with Drawndown Constraints.' Available at:

http://plaza.ufl.edu/zabarank/Drawdown.pdf (Accessed: 4 July 2006).

(Chekhlov 2003b) Chekhlov, A., Uryasev, S., and Zabarankin, M. (2003) *Research Report on Drawndown Measure in Portfolio Optimisation.* Florida: Center for Applied Optimisation, Department of Industrial and Systems Engineering, University of Florida.

(Cheng 1980) Cheng, P.L., and Grauer, R.R. (1980) 'An Alternative Test of the Capital Asset Pricing Model.' *American Economic Review,* 70(4), pp. 660-671.

(Crama 2003) Y. Crama, Y., and Schyns, M. (2003) 'Simulated Annealing for Complex Portfolio Selection Problems.' *European Journal of Operational Research, 150,* pp. 546-571.

(Davies 2009) Davies, R.J., Kat, H.M., and Lu, S. (2009) 'Fund of Hedge Funds Portfolio Selection: A Multiple-Objective Approach.' *Journal of Derivatives and Hedge Funds,* 15, pp. 91-115.

(Deb 1998) Deb, K. (1998) *Technical Report on Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Functions*. Dortmund: Department of Computer Science, University of Dortmund.

(Deb 2001) Deb K. (2001) *Multi-objective Optimization using Evolutionary Algorithms.* New York: John Wiley and Sons.

(Dembo 1989) Dembo R.S., Mulvey J.M., and Zenios S.A. (1989) 'Large-scale nonlinear network models and their application.' *Operations Research,* 37, pp. 353-372.

(DeMiguel 2009) DeMiguel, V., Garlappi, L., Nogales, F.J., and Uppal, R. (2009) 'A Generalized Approach to Portfolio Optimisation: Improving Performance by Constraining Portfolio Norms.' *Management Science,* 55(5), pp. 798-812.

(DeMiguel 2012) DeMiguel, V., Plyakha, Y., Uppal, R., and Vilkov, G. (2012) 'Improving portfolio selection using option-implied volatility and skewness.' *Journal of Financial and Quantitative Analysis. Available at:* http://www.edhec-risk.com/edhec_publications/all_publications/RISKReview.2011-09-09.2026/attachments/EDHEC_Working_Paper_Improving_Portfolio_Selection_F.pdf (Accessed 15 June 2013)

(Derigs 2004) Derigs, U., and Nickel, N.H. (2004) 'On a Local-Search Heuristic for a Class of Tracking Error Minimisation Problems in Portfolio Management.' *Annals of Operations Research,* 131 (1-4), pp. 45-77.

(Derman 1996) Derman, E. (1996) *Goldman Sachs Quantitative Strategies Research Notes on Model Risk.* New York: Goldman Sachs & Co.

(Doerner 2001) KL Doerner, K.L., Gutjahr, W.J.L., Harti, R.F.L., Strauss, C.L., and C Stummer C. (2001) 'Ant Colony Optimisation in Multiobjective Portfolio Selection.' *MIC'2001: proceedings of the 4th Metaheuristics International Conference,* Porto, Portugal, 16-20 July*. Available at:* http://webhost.ua.ac.be/eume/MIC2001/MIC2001_243_248.pdf (Accessed: 4 July 2006).

(Doerner 2004) Doerner, K.L., Gutjahr, W.J.L., Harti, R.F.L., Strauss, C.L., and Stummer, C. (2004) ‘Pareto Ant Colony Optimisation: A Metaheuristic Approach to Multiobjective Portfolio Selection.’ *Annals of Operations Research,* 131(1-4), pp. 79-99.

(Dorigo 1997) Dorigo, M. (1997) ‘Ant Colony System: A Cooperative Learning Approach to the Travelling Salesman Problem.’ *IEEE Transactions on Evolutionary Computation, 1(1)*, pp.53-66.

(Dupaèová 1999) Dupaèová, J. (1999) ‘Portfolio optimisation via stochastic programming: Methods of output analysis.’ *Mathematical Methods of Operations Research (ZOR), 50(2)*, pp. 245 – 270.

(Ehrgott 2004) Ehrgott, M., Klamroth, K., and Schwehm, C. (2004) ‘An MCDM approach to portfolio optimization.’ *European Journal of Operational Research,* 155, pp. 752-770.

(Elton 1974) Elton E.J., and Gruber, M.J. (1974) ‘Portfolio Theory when Investment Relatives are Lognormally distributed.’ *Journal of Finance, 29,* pp. 1265-1273.

(Elton 1997) Elton E.J., and Gruber, M.J. (1997) ‘Modern Portfolio Theory 1950 to date.’ *Journal of Banking & Finance, 21*, pp. 1743-1759.

(Fama 1965) Fama, E.F. (1965) ‘Portfolio Analysis in a Stable Paretian Market.’ *Management Science,* 11, pp. 409-419.

(Fama 1973) Fama, E.F., and MacBeth, J.D. (1973) ‘Risk, Return and Equilibrium: Empirical Tests.’ *Journal of Political Economy,* 81(3), pp. 607-636.

(Feo 1995) Feo, T.A. and Resende, M.G.C. (1995) ‘Greedy Ramdomized Adaptive Search Procedure.’ *Journal of Global Optimization,* 9, pp. 109-134.

(Fernholz 1982) Fernholz, R., and Shay, B. (1982) ‘Stochastic Portfolio Theory and Stock Market Equilibrium.’ *Journal of Finance,* 37, pp. 615-622.

(Fernholz 2002) Fernholz, R. (2002) *Stochastic Portfolio Theory*. Springer

(Fernholz 2003) R. Fernholz, R. (2003) 'The Application of Stochastic Portfolio Theory to Equity Management.' Available at: http://www.intechjanus.com/pub/research/The_Application_of_Stochastic_Portfolio_Theory_to_Equity_Management.pdf. (Accessed: 7 July 2010).

(Filedsand 2004) Filedsand, J.E., Matatko, J., and Peng, M. (2004) 'Cardinality Constrained Portfolio Optimisation.' *IDEAL 2004: proceedings of Conference on Intelligent Data Engineering and Automated Learning.* Berlin: Springer, pp. 788-793.

(Fonseca 1993) Fonseca, C.M., and Flemming, P.J. (1993) 'Genetic Algorithm for Multiobjective Optimisation: Formulation, Discussion and Generalization.' in Forrest S. (ed.) *Proceeding of the Fifth International Conference on Genetic Algorithms 1993.* San Mateo: Morgan Kaufmann, pp. 416-423.

(Gaivoronski 2004) Gaivoronski, A.A., and Pflug, G. (2004) 'Value-at-Risk in Portfolio Optimisation: Properties and Computational Approach.' *Journal of Risk, 7(2),* pp. 1-31.

(Gasparo 2007) Gasparo, L.D., Tollo, G., Roli, A. and Schaerf, A. (2007) 'Hybrid Local Search for Constrained Financial Portfolio Selection Problems.' In Van Hentenryck, P., and Wolsey, L. (eds.) *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimisation Problems*. Berlin: Springer, pp. 44-58.

(Ghaoui 2003) Laurent, E.l.G., Oks, M., and Oustry, F. (2003) 'Worst-Case Value-At-Risk and Robust Portfolio Optimisation: A Conic Programming Approach.' *Operations Research, 51(4),* pp. 543-556.

(Glover 1990) Glover, F. (1990) 'Tabu Search: A Tutorial.' *Interfaces,* 20 (4), pp. 74-94.

(Glover 2000) Glover, F., and Laguna, M. (2000) ‘Fundamentals of Scatter Search and Path Relinking.’ *Control and Cybernetics*, 29 (3), pp. 653-684.

(Gibbson 1982) Gibbsons, M.R. (1982) ‘Multivariate Tests for Financial Models: A new Approach.’ *Journal of Financial Economics,* 10(1), pp. 3-27.

(Gilli 2002) Gilli, M., and Kellezi, E. (2002) ‘A Global Optimisation Heuristic for Portfolio Choice with VaR and Expected Shortfall.’ in Kontoghiorghes, E.J., Rustem, B. and Siokos, S. (eds.) *Computational Methods in Decision-Makings, Economics and Finance*, Berlin: Springer.

(Gilli 2006) Gilli, M., Kellezi, E., and Hysi, H. (2006) ‘A Data-Driven Optimisation Heuristic for Downside Risk Minimisation.’*Journal of Risk.*

(Goldberg 1989) Goldberg, D.E. (1989) *Genetic Algorithms for Search, Optimization and Machine Learning,* Addison-Wesley.

(Gomez 2006) Gomez, M.A., Flores, C.X. and Osorio, M.A. (2006) ‘Hybrid Search Cardinality Constrained Portfolio Optimisation’ *Proceeding of GECCO 2006.*

(Greene 2008) Greene, W.H. (2008), *Econometric Analysis,* 6th edn. Upper Saddle River: Pearson International.

(Gulpinar 2003) Gulpýnar, N., Rustem, B., and Settergren, R. (2003) ‘Multistage Stochastic Mean-Variance Portfolio Analysis with Transaction Costs Innovations.’ *Financial and Economic Networks, 3*, p.46-63.

(Hassan 2010) Hassan, G.N.A. (2010) *Multiobjective Genetic Programming for Financial Portfolio Management in Dynamic Environments. PhD Thesis*, University College of London.

(Hochreiter 2007) Hochreiter, R. (2007) ‘An Evolution Computation Approach to Scenario-Based Risk-Return Portfolio Optimisation for a General Risk Measures.’ in

Giacobini, M. (ed.) *Application of Evolutionary Computing*. Berlin: Springer, pp. 199-207.

(Horn 1994) Horn, J., Nafpliotis, N., and Goldberg, D.E. (1994) 'A Niched Pareto Genetic Algorithm for Multiobjective Optimisation' *IEEE World Congress on Computational Intelligence: proceedings of the First IEEE Conference on Evolutionary Computation.*

(Horniman 2001) Horniman, M.D., Jobst, N.J., Lucas, C.A., and Mitra, G. (2001) 'Computational Aspects of Alternative Portfolio Selection Models in the Presence of Discrete Assets Choice Constraints.' *Quantitative Finance, 1*, pp. 489-501.

(Jagannathan 2003) Jagannathan, R., and Ma, T. (2003) 'Risk Reduction in Large Portfolios: Why Imposing the Wrong Constraints Helps.' *Journal of Finance, 58(4),* pp. 1651-1638.

(Jobst 2001) N.J. Jobst , N.J., Horniman, M.D., Lucas, C.A., and Mitra, G. (2001) 'Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints.' *Quantitative Finance, 1(* 5), pp. 489-501.

(Jorion 1992) Jorion, P. (1992) 'Portfolio Optimisation in Practice.' *Financial Analysis Journal*.

(Jurczenko 2006) Jurczenko, E., Maillet B., and Merlin, P. (2006) 'Hedge Fund Portfolio Selection with Higher-order Moments: A Nonparametric Mean-Variance-Skewness-Kurtosis Efficient Frontier.' in Jurczenko, E. and Maillet, B. (eds.) *Multi-moment Asset Allocation and Pricing Models,* Chichester: John Wiley and Sons, pp. 51-66.

(Kabanov 2004) Kabanov, Y., and Klüppelberg, C. (2004) ‘A geometric approach to portfolio optimisation in models with transaction costs.’ *Finance and Stochastic, 8 (2),* pp. 207-227.

(Kato 2000) Kato, T., and Yoshiba, T. (2000) ‘Model Risk and Its Control.’ *Monetary and Economic Studies,* 18(2), Tokyo: Institute for Monetary and Economic Studies, the Bank of Japan.

(Kirkpatrick 1983) Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. (1983) ‘Optimization by Simulated Annealing.’ *Science,* 220(4598).

(Konno 1991) Konno, H., and Yamazaki H. (1991) ‘Mean-absolute deviation portfolio optimisation model and its applications to Tokyo stock market.’ *Management Science,* 37, pp. 519-531.

(Knowles 1999) Knowles, J.D., Corne D.W. and Oates, M.J. (1999) ‘The Pareto Archived Evolutionary Strategy: A New Baseline Algorithm for Multiobjective Optimisation.’ *Proceeding IEEE Congress on Evolutionary Computation,* pp. 98-105.

(Kraus 1976) Kraus, A., and Litzenberger, R. (1976) ‘Skewness Preference and the Valuation of Risky Assets.’ *Journal of Finance, 21 (4),* pp. 1085-1100.

(Kwon 2005) Kwon, Y., Choi, S. and Moon, B. (2005) ‘Stock Prediction Based on Financial Correlation.’ *Proceeding of GECCO 2005.*

(Kwon 2007) Kwon, Y. and Moon, B. (2007) ‘A hybrid neuro-genetic approach for stock forecasting.’ *IEEE Transactions on Neural Networks,* 18(3), pp. 851-864.

(Larragara 2002) Larragra, P. and Lozano, J.A. (2002) *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic.

(Lee 1977) Lee, C.F. (1977) ‘Function Form Skewness Effect and Risk Return Relationship.’ *Journal of Finance and Quantitative Analysis*, 12, p. 55.

(Lintner 1965) Lintner J. (1965) ‘The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portfolio and Capital Budgets.’, *Review of Economics and Statistics, 47(1),* pp. 13-37.

(Loraschi 1995a) Loraschi, A., and Tettamanzi, A. (1995) ‘An evolutionary algorithm for portfolio selection in a downside risk framework.’ *Working Papers in Financial Economics, 6,* pp. 8-12.

(Loraschi 1995b) Loraschi, A., Tettamanzi, A., Tomassini, M., and Verda, P. (1995) ‘Distributed genetic algorithms with an application to portfolio selection problems.’ in Pearson, D.W., Steele, N.C. and Albrecht, R.F. (eds.) *Artificial Neural Networks and Genetic Algorithms,* Wein: Springer, pp. 384-387.

(Lutgens 2010) Lutgens, F. and Schotman, P. (2010) ‘Robust Portfolio Optimisation with Multiple Experts.’ *Review of Finance*, 14 (2), pp. 343-383.

(Lyuu 2002) Lyuu, Y. (2002) *Financial Engineering and Computation: Priciples, Mathematics, Algorithms*. Cambridge: Cambridge University Press.

(Mansiri 2003) Mansini, R., Ogryczak, W. and Speranza, M.G. (2003) ‘LP solvable models for portfolio optimisation: a classification.’ *IMA Journal of Management Mathematics, 14(3)*, pp.187-220.

(Markowitz 1952) Markowitz, H.M. (1952) ‘Portfolio Selection.’ *The Journal of Finance, 7(1),* pp. 77-91.

(Maringer 2003) Maringer, D., and Kellerer, H. (2003) ‘Optimisation of cardinality constrained portfolios with a hybrid local search algorithm.’ *OR Spectrum*, 25(4), pp. 481-495.

(Maringer 2005) Dietmar, M. (2005) *Portfolio Management with Heuristic Optimisation*, Dordrecht: Springer.

(Markus 1994) Markus, R. (1994) *Algorithms for Portfolio Optimisation and Portfolio Insurance.* Doctoral Dissertation, University of St. Gallen, Germany.

(McNelis 2005) McNelis, P.D. (2005) *Neural Networks in Finance: Gaining Predictive Edge in the Market*, Burlington: Elsevier.

(Mendel 2001) Mendel, J.M. (2001) *Uncertainty Rule-Based Fuzzy Logic Systems,* Upper Saddle River: Prentice Hall.

(Mladenovic 1997) Mladenovic M. and Hansen, P. (1997) 'Variable Neighborhood Search.' *Computer and Operations Research, 24(11),* pp. 1097-1100.

(Michalewicz 2004) Michalewicz Z., and Fogel, D.B. (2004) *How to Solve It: Modern Heuristics.* 2nd edn. *Berlin:* Springer.

(Mitchell 2002) Mitchell, J.E., and Braun, S. (2002) 'Rebalancing an Investment Portfolio in the Presence of Transaction Costs.' Available at: http://www.optimization-online.org/DB_FILE/2002/11/573.pdf (Assessed 6 Mar 2011).

(Mitra 2006) Mitra, G., and Ellison, E.F.D. (2006) *TECHNICAL REPORT on A Scowcroft Quadratic Programming for Portfolio Planning: Insights into Algorithmic and Computational Issues, CTR/45/06.* The Centre for the Analysis of Risk and Optimisation Modelling Applications, Brunel University. Available at: http://carisma.brunel.ac.uk/papers/2006/CTR%2045.pdf (Aceessed: 7 July 2010).

(Moody 1998) Moody, J. (1998) 'Forecasting the economy with neural nets : A survey of challenges and solutions.' In Orr G.B. and Muller, K. (eds.) *Neural Networks : Tricks of the Trade,* Berlin, Germany. Springer-Verlag, pp. 347-371.

(Mossin 1966) Mossin, J. 'Equilibrium in a Capital Asset Market.' *Econometrica 34(4),* pp. 768-783.

(Mukherjee 2002) Mukherjee, A., Biswas, R., Deb, K., and Mathur, A.P. (2002) 'Multi-objective Evolutionary Algorithms for the Risk-return trade-off in Bank Loan Management.' *International Transaction in Operations Research,* 9, pp. 583-597.

(Nagai 2000) Nagai, H. (2000) 'Risky fraction processes and portfolio optimisation with transaction costs 2000.' Available at: http://www.econ.kyoto-u.ac.jp/daiwa/Daiwa-Nagai.pdf (Accessed: 7July 2010).

(Negnevitsky 2005) Negnevitsky, M. (2005) *Artificial Intelligence: A Guide to Intelligent System,* 2nd edn. Harlow: Addison Wesley.

(Oberuc 2004) Oberuc, R.E. (2004) *Dynamic Portfolio Theory and Management,* New York: McGraw-Hill.

(Ong 2005) Ong, C., Huang J., and Tzeng, G. (2005) 'A Novel Hybrid Model for Portolio Selection.' *Applied Mathematics and Computation,* 169, pp. 1195-1210.

(Pafka 2003) Pafka, S., and Kondor, I. (2003) 'Estimated Correlation Matrices and Portfolio Optimisation.' Available at: http://arxiv.org/PS_cache/cond-mat/pdf/0305/0305475.pdf (Assessed: 5 November 2011).

(Perold 1984) Perold, A.F. (1984) 'Large-scale portfolio optimization.' *Management Science,* 30, pp. 1143-1160.

(Rachav 2007) Rachev, S.T., Mittnik, S., Fabozzi, F.J., Forcardi, S.M., and Jasic, T. (2007) *Financial Econometrics.* Hoboken: John Wiley and Sons.

(Ramaswamy 1998) Ramaswamy, S. (1998) 'Portfolio selection using Fuzzy decision theory.' *BIS Working paper, 59.*

(Rechenberg 1989) Rechenberg, A. (1989) *Evolutionary Strategy: Nature's Way of Optimization.* Berlin: Springer.

(Roll 1980) Roll R.R., and Ross, S.A. (1980) 'In Empirical Investigation of the Arbitrage Pricing Theory.' *The Journal of Finance*, 39(5), pp. 1073-1104.

(Ross 1967) Ross, S.A. (1967) 'The Arbitrage Theory of Capital asset Pricing.' *The Journal of Economic Theory,* 13(3), pp. 341-360.

(Sankaran 1999) Sankaran, J., and Krishnamurti, C. (1999) 'On the Optimal Selection of Portfolios under Limited Diversification.' *Journal of Banking and Finance, 23(11),* pp. 1655-1668.

(Scherer 2005) Scherer, B., and Martin, D. (2005) *Introduction to Modern Portfolio Optimization,* New York: Springer.

(Schyns 2001) Schyns, M. (2001) *Modellng Financial Data and Portfolio Optimisation Problems.* Doctoral Dissertation, Ecole d' Administration des Affaires.

(Smithson 2006) Smithson, M., and Verkuilen, J. (2006) *Fuzzy Set Theory: Applications in the Social Sciences.* Thousand Oaks: SAGE Publication.

(Sharp 1964) Shape W.E. (1964) '*Capital Assets Prices: A Theory of Market Equilibrium and Conditions of Risk*.' *The Journal of Finance, 19(3),* pp. 425-442.

(Skolpadungket 2007) Skolpadungket, P., Dahal, K., and Harnpornchai, N. (2007) 'Portfolio optimisation using multi-objective genetic algorithms.' *CEC 2007: proceeding* in *IEEE Congress on Evolutionary Computation 2007,* Singapore.

(Srinivas 1994) Srinivas, N., and Deb, K. (1994) 'Multiobjective Optimisation using Non-dominated Sorting in Genetic Algorithms.' *Evolutionary Computation, 2*(3), pp. 221-248.

(Stein 2005) Stein, M., Branke, J. and Schmeck, H. (2005) 'Portfolio Selection: How to Integrate Complex Constraints.' Available at: http://symposium.fbv.uni-karlsruhe.de/10th/papers/Stein_Branke_Schmeck%20-%20Portfolio%20Selection%20How%20to%20Integrate%20Complex%20Constraints.pdf (Assessed: 6 December 2011).

(Steuer 2003) Steuer, R.E., and Qi, Y. (2003) 'Computational Investigations Evidencing Multiple Objectives in Portfolio Optimisation' in Tanino, T., Tanaka, T., and Inuiguchi, M. (eds.) *Multi-Objective Programming and Goal Programming: Theory and Applications*. Springer Netherlands, pp. 35-43.

(Streichert 2004 a) Streichert, F., Ulmer, H., and Zell, A. (2004) 'Comparing Discrete and Continuous Genotypes on the Constrainted Portfolio Selection Problem.' Available at: http:///www-ra.informatik.uni-tuebingen.de/ (Assessed: 7 September 2011).

(Streichert 2004 b) Streichert, F., Ulmer, H., and Zell, A. (2004) 'Evaluating a Hybrid Encoding and Three Crossover Operators on the Constrained Portfolio Selection Problem.' *CEC2004: proceedings of Congress on Evolutionary Computation 2004, 1,* IEEE*, pp.* 932-939.

(Streichert 2005) Streichert, F., Ulmer, H., and Zell, A. (2005) 'Hybrid Representation for Compositional Optimisation and Parallelizing MOEAs.' Available at: http://drops.dagstuhl.de/opus/volltexte/2005/251/pdf/04461 (Assessed: 19 September 2011).

(Subbu 2005) Subbu, R., Bonissone, P.P., Eklund, N., Bollapragada, S., and Chalermkraivuth, K. (2005) 'Multiobjective financial portfolio design: a hybrid evolutionary approach.' CEC 2005: proceeding of The IEEE Congress, 2, IEEE, pp. 1722- 1729.

(Takehara 1993) Takehara, H. (1993) 'An interior point algorithm for large scale portfolio optimisation.' *Annals of Operations Research,* 45, pp. 373-386.

(Tanaka 2001) Tanaka, H., and Guo, P. (2001) *Possibilistic Data Analysis for Operations Research.* New York: Physica-Verlag.

(Triana 2000) Triana, P. (2000) *Lecturing Birds on Flying,* Hoboken: John Wiley and Sons.

(Tobin 1958) Tobin, J. (1958) 'Liquidity Preference as Behaviour toward Risk.' *Review of Economic Studies*, 25, pp. 65-86.

(Tobin 1965) Tobin, J. (1965) 'The Theory of Portfolio Selection.' in Hahn F., and Brechling, F. (eds.) *The Theory of Interest Rates,* London: MacMilan.

(Varghese 2004) Varghese, B., and Poojari, C.A. (2004) *Technical Report on Genetic algorithm based technique for solving chance constrained problems arising in risk management* . CARISMA, Brunel University. Available at: http://www.carisma.brunel.ac.uk/papers/Varghese.pdf (Assessed 7 June 2011).

(Yao 1999) Yao, X. (1999) 'Evolutionary artificial neural networks.' *Proceeding of the IEEE, 87(9).*

(Yang 1993) Yang, M.S. (1993) 'A Survey of Fuzzy Clustering.' *Mathematical and Computer Modelling, 18(11),* pp. 1-16.

(Yilmaz 2011) Yilmaz, A.E., and Weber, G. (2011) 'Why You Should Consider Nature-Inspired Optimisation Methods in Financial Mathematics.' in Machado, J.A.T., Baleanu, D., and Luo, A.C. (eds.) *Non-Linear and Complex Dynamics,* Berlin: Springer, Berlin.

(Wang 2002) Wang, Y.F. (2002) 'Predicting Stock Price using Fuzzy Grey Predicting System.' *Expert Systems with Applications, 22(1),* pp. 33-39.

(Wolfe 1959) Wolfe, P. (1959) 'The Simplex Method for Quadratic Programming.' *Econometrica*, 27, pp. 382-398.

(Wyatt 1997) Wyatt, K. (1997) 'Decomposition and Search Techniques in Disjunctive Programs for Portfolio Selection.' Available at http://www.sci.brooklyn.cuny.edu/~wyatt/dpport.ps. (Assessed on 7July 2009).

(Zadeh 1988) Zadeh, L.A. (1988) ‘Fuzzy Logic.’ *Computer, 21(4*), pp. 83-93.

(Zitzler 1999) Zitzler, E., and Thiele, L. (1999) ‘Multiobjective Optimisation using Evolutionary Algorithms – A Comparative Case Study and the Strength Pareto Approach.’ *IEEE Transactions on Evolutionary Computation,* 3(4), pp. 257-271.

(Zitzler 2002) Zitzler, E., Laumanns, M., and Thiele, L. (2002) ‘SPEAII: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimisation.’ *CIMNE 2002: proceedings of the Conference on Evolutionary Methods for Design, Optimisation, and Control, pp. 95-100.*

(Zhou 1996) Zhou, C. (1996) ‘Forecasting long- and short- horizon stock returns in a unified framework.’ *Board of Governors of the Federal Reserve System Finance and Economics Discussion Series: The Federal Reserve System,* 96-4.