Centre for Quantitative Finance

Imperial College London

An Evolutionary Approach to Multistage Portfolio Optimization

Jonathan Rosmarin

Thesis Submitted for the Degree of

Doctor of Philosophy

September 2007

BIBL. LONDON UNIV.

Abstract

Portfolio optimization is an important problem in quantitative finance due to its application in asset management and corporate financial decision making. This involves quantitatively selecting the optimal portfolio for an investor given their asset return distribution assumptions, investment objectives and constraints. Analytical portfolio optimization methods suffer from limitations in terms of the problem specification and modelling assumptions that can be used. Therefore, a heuristic approach is taken where Monte Carlo simulations generate the investment scenarios and a problem specific evolutionary algorithm is used to find the optimal portfolio asset allocations.

Asset allocation is known to be the most important determinant of a portfolio's investment performance and also affects its risk/return characteristics. The inclusion of equity options in an equity portfolio should enable an investor to improve their efficient frontier due to options having a non-linear payoff. Therefore, a research area of significant importance to equity investors, in which little research has been carried out, is the optimal asset allocation in equity options for an equity investor. A purpose of my thesis is to carry out an original analysis of the impact of allowing the purchase of put options and/or sale of call options for an equity investor. An investigation is also carried out into the effect of changing the investor's risk measure on the optimal asset allocation.

A dynamic investment strategy obtained through multistage portfolio optimization has the potential to result in a superior investment strategy to that obtained from a single period portfolio optimization. Therefore, a novel analysis of the degree of the benefits of a dynamic investment strategy for an equity portfolio is performed. In particular, the ability of a dynamic investment strategy to mimic the effects of the inclusion of equity options in an equity portfolio is investigated.

The portfolio optimization problem is solved using evolutionary algorithms, due to their ability to incorporate methods from a wide range of heuristic algorithms. Initially, it is shown how the problem specific parts of my evolutionary algorithm have been designed to solve my original portfolio optimization problem. Due to developments in evolutionary algorithms and the variety of design structures possible, a purpose of my thesis is to investigate the suitability of alternative algorithm design structures. A comparison is made of the performance of two existing algorithms, firstly the single objective stepping stone island model, where each island represents a different risk aversion parameter, and secondly the multi-objective Non-Dominated Sorting Genetic Algorithm2. Innovative hybrids of these algorithms which also incorporate features from multi-objective evolutionary algorithms, multiple population models and local search heuristics are then proposed.

A novel way is developed for solving the portfolio optimization by dividing my problem solution into two parts and then applying a multi-objective cooperative coevolution evolutionary algorithm. The first solution part consists of the asset allocation weights within the equity portfolio while the second solution part consists of the asset allocation weights within the equity options and the asset allocation weights between the different asset classes. An original portfolio optimization multi-objective evolutionary algorithm that uses an island model to represent different risk measures is also proposed.

Table of Contents

List of Figures..... 7

List of Tables..... 9

1. Introduction 10

2. Portfolio Optimization..... 14

2.1. Mean-Variance Analysis..... 15

2.1.1. Limitations of Mean-Variance Analysis..... 15

2.1.2. Addressing the Problem of Estimation Error in Mean-Variance Analysis..... 18

2.2. Scenario Generation and Optimization..... 20

2.2.1. Scenario Generation..... 20

2.2.2. Scenario Optimization..... 22

2.3. Methods for Multistage Portfolio Optimization..... 25

2.4. Literature Review on Heuristic Approaches to Portfolio Optimization.....27

2.5. Conclusion..... 32

3. Heuristic Optimization Methods and Evolutionary Algorithms 34

3.1. Heuristic Optimization Methods..... 35

3.1.1. Single-Solution Heuristics..... 36

3.1.2. Population Heuristics 39

3.2. Evolutionary Algorithms 42

3.2.1. Introduction to Evolutionary Algorithms..... 42

3.2.2. Selection Methods..... 48

3.2.3. Variation Operators..... 56

3.2.4. Methods for Ensuring Population Diversity 58

3.2.5. Multiple Population Models..... 60

3.2.6. Parameter Tuning and Parameter Control..... 68

3.2.7. Advantages of Evolutionary Algorithms..... 70
3.3. Incorporating Constraints..... 71
3.4. Designing and Reporting on Heuristic Methods..... 76
3.5. Conclusion..... 77

4. Implementation of the Multistage Portfolio Optimization Model..... 79

4.1. The Objective Function..... 80
4.1.1. Alternative Risk Measures..... 81
4.1.2. Coherent Risk Measures..... 87
4.2. Multiobjective Optimization..... 88
4.2.1. Multiple Single-Objective Searches..... 92
4.2.2. Single-Objective Simultaneous Parallel Searches..... 93
4.2.3. Pareto Dominance Approach..... 94
4.3. Multistage Portfolio Optimization Model 98
4.4. Scenario Generation 100
4.4.1. Equity Factor Models and Derivative Pricing..... 105
4.4.2. Data Summary..... 108
4.5. Common Features of the Portfolio Optimization Problem 110
4.6. Common Features of my Evolutionary Algorithms..... 110
4.7. Conclusion..... 113

5. Incorporating Derivatives in Portfolio Optimization 114

5.1. Benefits of using Derivatives in Portfolio Optimization 114
5.2. Specific Parts of the Problem Specification and Evolutionary Algorithm..... 116
5.3. Computational Results 119
5.3.1. Asset Allocation Weights for Different Risk Measures..... 120
5.3.2. Efficient Frontiers for Different Derivatives Strategies 126
5.4. Conclusion..... 130

6. Multistage Portfolio Optimization 131

6.1. Dynamic Asset Allocation Strategies..... 131

6.1.1. Multi-Period Models..... 131
6.1.2. Portfolio Insurance..... 135
6.2. Methods for Generating Scenarios for Multistage Optimization..... 139
6.3. Specific Parts of the Problem Specification and Evolutionary Algorithm 144
6.4. Computational Results..... 146
6.4.1. Constant Mix Strategy..... 146
6.4.2. Constant Proportion Portfolio Insurance Strategy..... 147
6.4.3. Fixed Equity Cluster Strategy..... 155
6.5. Conclusion..... 159
7. Comparison of various Evolutionary Algorithms 160
7.1. Memetic Algorithms..... 160
7.2. Coevolution..... 163
7.3. Multiobjective Optimization..... 166
7.3.1. Cooperative Coevolution Approach..... 166
7.3.2. Island Model Approach..... 168
7.3.3. Measuring Convergence..... 170
7.3.4. Performance Metrics..... 171
7.4. Alternative Evolutionary Algorithm Designs..... 172
7.5. Computational Results..... 177
7.5.1. Type SSRAM type algorithms..... 178
7.5.2. Type NSGA2 algorithms 181
7.5.3. Hybrid SSRAM and NSGA2 algorithm 184
7.6. Conclusion..... 186
8. Conclusions..... 187
8.1. Areas for Further Research 189
Abbreviations..... 190
References 191

List of Figures

Figure 3.1: Flow diagram of a basic evolutionary algorithm..... 45

Figure 3.2: Constraint handling by separating individuals and constraints 73

Figure 4.1: Example efficient frontier..... 88

Figure 4.2: Stepping stone risk aversion island model diagram..... 93

Figure 4.3: Simulated one year equity market returns 108

Figure 5.1: Equity portfolio split for the strategy where derivatives are allowed when variance is the risk measure..... 123

Figure 5.2: Efficient frontiers for the different derivative strategies when variance is the risk measure and the maximum allowed cash allocation is 30%..... 127

Figure 5.3: Efficient frontiers for the different derivative strategies when variance is the risk measure and no cash allocation is allowed 128

Figure 5.4: Efficient frontiers for the different derivative strategies when lower semi-variance is the risk measure 129

Figure 5.5: Efficient frontiers for the different derivative strategies when lower partial moment is the risk measure..... 130

Figure 6.1: Parallel simulation and sequential simulation example..... 142

Figure 6.2: 2-stage scenario tree of the equity market return..... 144

Figure 6.3: Efficient frontier for a constant mix strategy when variance is the risk measure 147

Figure 6.4: Efficient frontier for a CPPI strategy when variance is the risk measure..... 148

Figure 6.5: Efficient frontier for a CPPI strategy when lower semi-variance variance is the risk measure..... 149

Figure 6.6: Efficient frontier for a CPPI strategy when lower partial moment is the risk measure..... 150

Figure 6.7: A comparison of the efficient frontiers of a CPPI strategy with a derivative strategy when lower semi-variance is the risk measure..... 151

Figure 6.8: A comparison of the efficient frontiers of a CPPI strategy with a derivative strategy when lower partial moment is the risk measure 152
Figure 6.9: A comparison of the efficient frontiers of a CPPI strategy with alternative equity portfolios when variance is the risk measure..... 154
Figure 6.10: Equity portfolio split for the CPPI strategy when variance is the risk measure..... 155
Figure 6.11: Efficient frontier for the fixed equity cluster strategy when variance is the risk measure 157
Figure 6.12: Efficient frontier for the fixed equity cluster strategy when lower semi-variance is the risk measure 157
Figure 6.13: Efficient frontier for the fixed equity cluster strategy when lower partial moment is the risk measure..... 158
Figure 6.14: A comparison of the efficient frontiers of a fixed equity cluster strategy and CPPI strategy for quarterly rebalancing when lower partial moment is the risk measure..... 159
Figure 7.1: Specialized island model diagram 169
Figure 7.2: NSGA2-MP algorithm diagram 175
Figure 7.3: SS NSGA2 algorithm diagram..... 176
Figure 7.4: SSRAM-CC—NSGA2 algorithm diagram 177
Figure 7.5: Efficient frontier for the call/put derivative strategy where lower partial variance is the risk measure..... 181

List of Tables

Table 4.1: Equity return statistics..... 106
Table 4.2: Principal component eigenvalues 107
Table 4.3: Principal components for the FTSE All Share industrial sectors..... 108
Table 4.4: Example of two methods used for bounding asset allocations..... 112
Table 4.5: Common features and parameters of my evolutionary algorithms...113
Table 5.1: Asset allocations when variance is the risk measure 121
Table 5.2: Asset allocations when lower semi-variance is the risk measure..... 124
Table 5.3: Asset allocations when lower partial moment is the risk measure... 125
Table 6.1: Scenario tree structure 143
Table 6.2: Multiplier and floor values for the CPPI strategy..... 151
Table 6.3: Equity volume turnover for different CPPI parameters..... 153
Table 6.4: Fixed equity cluster strategy allocations for the semi-annual rebalancing case..... 156
Table 7.1: Features and parameters of type SSRAM algorithms 173
Table 7.2: NSGA2-MP model features and parameters..... 175
Table 7.3: SSRAM-CC—NSGA2 model features and parameters..... 177
Table 7.4: Performance comparison of the different type SSRAM algorithms. 179
Table 7.5: Performance metrics of the different type NSGA2 algorithms..... 182
Table 7.6: C metric of the different type NSGA2 algorithms.....183
Table 7.7: Performance metrics of the SSNSGA2 and NSGA2 algorithms..... 183
Table 7.8: C metric of the SSNSGA2 and NSGA2 algorithms..... 184
Table 7.9: Performance comparison of the type SSRAM algorithms and hybrid SSRAM-CC— NSGA2 algorithm..... 184
Table 7.10: Performance metrics of the type NSGA2 algorithms and hybrid SSRAM-CC— NSGA2 algorithm..... 185
Table 7.11: C metric of the type NSGA2 algorithms and hybrid SSRAM-CC—NSGA2 algorithm..... 186

Chapter 1

Introduction

Fund management involves the selection of assets for institutional investors such as pension schemes and insurance companies as well as for private individuals. A key aspect of this is asset allocation in which the proportion of the portfolio invested in each permissible asset is selected. Equity exposure to the UK stock market has traditional been gained through investment in either an active or passive equity fund. In active equity management the aim of the fund is to outperform its benchmark through a combination of sector allocation, stock selection and market timing. The aim of a passive equity fund is to perform in line with its benchmark through the process of tracking. A mixture of these two approaches is active indexing where an indexed portfolio can be actively tilted in order to increase its exposure to the types of stock expected to outperform.

In recent years, the growth of hedge funds has been fuelled by the demand from investors for equity exposure not limited to pure equity long only strategies. Hedge funds are able to leverage their position in order to gain exposure that is greater than the net value of its fund through either borrowing or the use of derivatives. Short selling can also be used in order to benefit from equity price falls and result in the possibility of market neutral equity funds which have no net equity exposure and a cash benchmark. Hedge funds are more likely to feature as part of a diversified portfolio rather than comprising the majority of an investor's portfolio.

Institutional investors have the ability to alter the risk/return profile of their equity portfolio through taking positions in equity derivatives. Retail investors have more limited choices, with guaranteed equity products being a

popular method for gaining equity exposure while protecting against equity falls through the provision of a minimum guarantee. The product issuer can hedge the promised payoff by investing an amount in a bond which at maturity meets the guaranteed element of the product, while using the remainder of the investor's payment to purchase call options. Alternatively, the issuer could combine a long position in the equity market with the purchase of put options.

A major objective of my research is to investigate the effects of being able to combine equity derivatives within an equity portfolio and of allowing a dynamic investment strategy. A further objective is to investigate how the chosen risk measure effects the optimal asset allocation. In order to allow for flexibility in the specification of an investor's market views and his chosen risk measure I use a scenario generation approach to simulate my scenarios and a heuristic optimization approach to find the efficient frontier (i.e. optimal tradeoff between risk and return). The third objective is to compare the performance of a variety of innovative evolutionary algorithms to the portfolio optimization problem.

This thesis is divided as follows:

Chapter 2 begins by discussing the limitations of using a traditional mean variance approach when solving portfolio optimization problems. I explain how these shortcomings can be addressed by using the alternative approach of scenario generation and optimization. This technique uses Monte Carlo simulation to produce realistic future scenarios and then applies a heuristic algorithm to solve the multistage portfolio optimization problem. The chapter concludes with a literature review of heuristic approaches that have been used to solve portfolio optimization problems.

Chapter 3 begins by discussing the main types of single-solution heuristics and population heuristics that can be used to solve the portfolio optimization problem defined as finding the optimal asset allocations for an equity investor who is allowed to incorporate equity options into their portfolio or

use a number of dynamic investment strategies. Evolutionary algorithms provide a flexible, robust and efficient method for solving my optimization problem. I detail how evolutionary algorithms work and key elements of their design such as specification of the selection and variation parameters, ways for ensuring population diversity, use of multiple populations and parameter settings. I describe how constraints can be handled by heuristic methods and issues that are important when designing heuristic algorithms.

In Chapter 4 I describe details of the method used to implement the multistage portfolio optimization problem. The simple form of the objective function is to maximize the linear combination of wealth less a risk measure, where the relative weighting is given by the risk aversion parameter. By optimizing for a full range of risk aversion parameters one can draw an entire efficient frontier. Alternative specifications of the risk measure are discussed along with my reasons for concentrating on downside risk measures in my investigations. The portfolio optimization problem can also be solved using a multiobjective evolutionary algorithm due to the two objectives of maximizing expected return and minimizing the specified risk measure. A popular approach is that of a Pareto dominance approach in which a solution that has Pareto dominance (i.e. has both higher expected return and lower risk) will be favoured in the evolutionary process. I then describe the general framework for the multistage portfolio optimization model, along with further detail of the specific methods used for scenario generation. Finally, design features that are common to all the different evolutionary algorithms and problem specifications are described.

Chapter 5 begins by discussing the reasons why it should be beneficial to allow equity derivatives in the portfolio optimization problem defined. The method used by my algorithm to allow derivatives into my problem is detailed. I give computational results that show that the efficient frontier improves when one

allows an equity investor to either sell call options or buy put options. Differences in the optimal asset allocation for various risk measures are also detailed.

In chapter 6 I explain the reasons why multi-period dynamic asset allocation strategies should be preferred to single period static strategies. Alternative dynamic asset allocation strategies are discussed, with particular emphasis on constant proportion strategies. I explain my choice of using a parallel simulation approach when generating a non-anticipatory scenario tree, which is needed due to the multistage nature of my problem. I give computational results that show the improvement in the efficient frontier possible when one allows an investor to rebalance the portfolio more frequently.

In chapter 7 I compare the performance of various evolutionary algorithms that can be used to solve the portfolio optimization problem. I begin by discussing how evolutionary algorithms can be designed more efficiently by incorporating a local search process into the evolutionary algorithm and using coevolution methods. I then explain how these techniques, along with multiple population models, can be used when designing multiobjective evolutionary algorithms. I then detail performance metrics that can be used to compare the performance of different multiobjective evolutionary algorithms, and use them to show how the novel algorithms can result in an improved performance.

Finally, Chapter 8 gives the conclusion of my research and discusses areas for possible future research.

Chapter 2

Portfolio Optimization

Markowitz (1952) introduced mean-variance optimization as the framework for determining the optimal portfolio allocation over a single time horizon. The only inputs needed for this model are the expected returns and variances of each asset class, along with the correlations between the returns of different asset classes, implicitly assuming returns are multivariate normal. Standard quadratic programming algorithms can then be used to find the portfolio allocation weights that minimize the risk, described by variance, for each given level of expected return. This has been a popular approach which provides insights into portfolio allocation, although more recently the limitations of this approach have resulted in the development of the alternative approach of scenario optimization.

Scenario optimization is being used as a flexible and intuitive modelling framework for solving a variety of financial decision making problems in the presence of uncertainty. Scenario optimization uses Monte Carlo techniques to simulate many future scenarios and stochastic optimization to determine the optimal financial decisions given the simulated scenarios. Scenario optimization is well suited to the problem of portfolio optimization through the combination of modern portfolio theory, stochastic optimization, Monte Carlo simulation and statistical techniques.

Traditional mean-variance optimization determines the initial optimal portfolio allocation over a single time horizon. In contrast, multistage optimization divides the time horizon into a number of intervals and determines the optimal investment strategy over each interval. The ability for financial

decisions to be made at periodic intervals over the investment horizon, rather than only once at the beginning, should result in a more realistic modelling framework with superior performance. Therefore, my modelling framework will be carried out in a multistage setting.

2.1. Mean variance analysis

2.1.1. Limitations of mean variance analysis

Assumptions behind mean variance analysis

Mean-variance optimization assumes that either asset returns follow a multivariate normal distribution or that investors have a quadratic utility function. The advantage of assuming normality of asset returns is the relatively simple numerical calculations needed due to being able to aggregate risks both across assets and over time. However, both of these assumptions which mean-variance analysis is based on can be strongly criticized.

Firstly, empirical evidence shows that asset returns are not normally distributed and may exhibit excess kurtosis (leptokurtosis) and skewness, see Mandelbrot (1963) and Fama (1965). In addition, the dependence structure between the returns of different asset classes cannot be completely described by a multivariate normal distribution. It is also unrealistic to assume that the covariance matrix is constant in time. The incorporation of options in a mean-variance analysis would not be appropriate due to the skewness of the return distribution. Secondly, one cannot assume that all investors have a quadratic utility function. A typical investor does not have an increasing absolute risk aversion as the level of wealth rises, which is implied by a quadratic utility function. The symmetric definition of risk given by variance is counter-intuitive as it has the feature of equal penalization of gains and losses, even though investors are concerned about the asymmetric behaviour of returns, see Sortino & Forsey (1996).

The risk of holding illiquid assets such as property or venture capital cannot be adequately described by variance. Therefore, in a mean-variance framework one will need to make an allowance for the over allocation in the portfolio weights that is given to illiquid asset classes due to the artificially low volatility. Mean-variance analysis is also unable to include features such as mean reversion in interest rates.

Constraints which need to be included in a portfolio optimization due to limits practitioners are faced with can be hard to incorporate in traditional portfolio optimization approaches. Common constraints include asset allocation weight constraints, portfolio cardinality constraints (i.e. the minimum number of assets allowed in the portfolio), minimum transaction lots (i.e. the minimum volume of a security that can be purchased at any one time) and limits on the maximum volume of assets that can be traded. These complexities turn the standard Markowitz quadratic programming problem into a harder mixed-integer mathematical programming problem with nonlinear constraints. Yoshimoto (1996) showed that a quadratic programming method cannot be used to solve a portfolio optimization problem with fixed transaction costs. However, Adcock & Meade (1994) showed that variable costs can be incorporated in quadratic programming.

Asset allocation sensitivity to input assumptions

In order to find the optimal portfolio weights in the mean-variance framework it is necessary to calculate point estimates of asset means, variances and correlations. Even if we assume that asset returns are stationary having constant means and non-time dependent covariances, unless the sample size is very large differences in the estimated and true distribution parameters, i.e. estimation error, will occur.

Best & Grauer (1991) and Chopra & Ziemba (1993) show that small changes in the estimated input parameters can result in large changes in the composition of the optimal portfolios given by mean-variance optimization. They

find that errors in estimated means are more important than errors in estimated variances. Provided the assets are in the basis mean-variance optimization assumes that the input parameters of expected returns, variances and correlations are known with certainty, which does not reflect the true uncertainty in the model inputs. Small changes in estimated inputs result in large changes in the mean-variance optimal asset allocation which would be impractical and non-intuitive to implement.

Michaud (1989) discusses the benefits and limitations of mean-variance optimization. He states that "the unintuitive character of many optimized portfolios can be traced to the fact that mean-variance optimizers are, in a fundamental sense, 'estimation error maximizers'. Mean-variance optimization significantly overweights (underweights) those securities that have large (small) estimated returns, negative (positive) correlations and small (large) variances. These securities are, of course, the ones most likely to have large estimation errors." Therefore, the optimized portfolio is likely to contain allocation weights which seem to high or low that what an investment manager might intuitively expect. Meade (1993) and Meade & Salkin (2000) show that the use of forecasts can improve the estimates used.

Michaud (1989) also says that the limitation of the non-uniqueness of the optimal solution arises because the uniqueness of the solution is dependent upon the erroneous assumption that the mean-variance inputs are without statistical estimation error.

Asset allocation stability over the efficient frontier

Adler (1987) states that output weights of a mean-variance optimizer "are excessively unstable as portfolios are reoptimized and rebalanced over time". Rasmussen (2003) provides an example of how mean-variance optimization can result in optimal asset allocations that change quickly and abruptly as the level of risk is increased from the minimum-variance portfolio to the maximum-variance portfolio. The speed at which asset classes move in and out of the optimal

portfolio as the portfolio risk is increased, and the total exclusion of certain asset classes for portfolios with either low or high risk levels often lacks good investment intuition. In addition, such radical alterations in the portfolio allocation would result in high transaction costs if the target volatility was changed. The asset allocation instability is due to the optimization algorithm searching for extreme asset allocations that minimizes risk for a given level of expected return resulting in highly unstable and often non intuitive asset allocations. New estimates of returns and variances will also result in excessive instability to the optimal portfolio weights. A common approach used to improve asset allocation stability is by the inclusion of asset allocation bounds, which is discussed further below.

2.1.2. Addressing the problem of estimation errors in mean-variance analysis

Several approaches that can be used to reduce the impact of estimation errors when carrying out mean-variance portfolio optimization are Bayesian techniques, portfolio resampling and the incorporation of constraints. These methods are detailed below.

A Bayesian approach can be used to generate a posterior probability density for the underlying asset distribution parameters by combining historically observed data with prior beliefs. As more data becomes available, prior beliefs becomes less influential in determining the posterior distribution. Michaud (1989) suggests Stein estimation as an approach of reducing estimation error in the inputs by methodically massaging the estimated inputs. Stein estimation can be used to prevent the means and covariances from varying substantially both cross-sectionally and intertemporally. The benefits of this approach are also shown in Jobson & Korkie (1981). The problem with this approach arises when investors do not have informative priors to use. In this situation, one can use the Black-Litterman (1992) model that allows investors to calculate their optimal portfolio

based on anchoring one's predictive views onto the market portfolio. This is a popular model that allows investors to incorporate both their market forecasts and level of conviction in those forecasts. However, it has the disadvantages of not using all the information available in historic data and does not directly address parameter uncertainty.

Jorion (1992) describes how portfolio resampling can be used when the true underlying asset distributions are stationary and the estimation errors are due to the limited data available for precise estimation to occur. This method uses the estimated asset returns and covariances to generate random portfolio returns which are used to calculate an efficient frontier. These generated portfolio returns are then used to calculate an updated estimate of the asset returns, covariances and efficient frontier. This process is repeated a number of times, and results in the input estimation error being transferred into uncertainty about the optimal portfolio weights. Michaud (1998) shows that the more diversified resampled portfolio outperforms the corresponding mean-variance portfolio out-of-sample. However, portfolio resampling has the unfavourable feature of deteriorating Sharpe ratios caused by the increased allocation given to assets with high volatility. This is because favourable returns scenarios lead to large portfolio allocations, while unfavourable return scenarios lead at most to zero allocations due to the long-only constraint.

An alternative approach is to place constraints on the permitted asset allocations, e.g. one can add minimum and maximum allocation constraints for each asset and asset class. Frost & Savarino (1998) and Chopra (1993) found that constraining weights resulted in a more diversified and stable portfolio with improved performance as the detrimental effects of estimation errors on portfolio weights were reduced. However, this approach requires the relative subjective setting of constraints as a way of pushing the optimized asset allocations towards a portfolio which the asset managers believe should be most attractive. When using constraints, the benefits of the reasonable portfolios achieved must be offset against the resulting deterioration of the efficient frontier.

2.2. Scenario generation and optimization

An alternative to a traditional approach to portfolio optimization is given by scenario generation and optimization. A scenario is a single deterministic realization of all the relevant economic uncertainties over the future specified time horizon. The aim of scenario generation is to produce a set of simulated scenarios that result in a realistic and representative set of possible future economic outcomes. In order to ensure that a consistent economic outcome is produced by each scenario the model needs to take into account the historic dependencies between all of the economic uncertainties. Therefore, the accuracy of the scenario optimization will be heavily dependent on the quality of the estimation of the asset return distributions and the relationship of returns between asset classes.

Once the scenarios have been generated, uncertainty has been removed. Scenario optimization can then be used to solve the deterministic problem of finding the optimal asset allocation weights given by the maximization of the chosen objective function.

2.2.1. Scenario generation

The quality of the optimal solution determined by the scenario optimization will depend on the quality of the scenarios generated. Scenarios need to be representative in order to provide a realistic description of the problem and should be carried out in a way that minimizes estimation error. In order to minimize the implementation time, this representativeness of the scenarios should be achieved with the fewest number of scenarios possible. Scenarios should also be arbitrage-free, which is discussed later in this chapter, so that the optimal solution found does not include highly attractive solutions that do not have sound investment intuition.

Methods for scenario generation

- *Bayesian techniques.* This method uses a predictive distribution combined with historic data to model asset distributions and benefits from allowing for data estimation uncertainty. The disadvantages of this approach arise from difficulty in choosing the form and level of belief in the prior distribution. In addition, the prior distribution chosen may be unable to adequately model the non-normality in the data.
- *Bootstrapping*. This technique involves repeated random independent draws with replacement of single period returns across different assets. In its simplest form it has the disadvantages of failing to capture any dependencies between two asset return realizations such as autocorrelation in returns and GARCH effects.
- *Monte Carlo simulation.* In this method one needs to choose a suitable parametric distribution to model the multivariate asset distributions and then estimate the model parameters using historical data. This is the method I am going to use due to its high level of flexibility and the large range of financial time series models that can be used within this approach.

Ensuring scenarios are arbitrage free

A desired feature of the scenarios generated is the elimination of possible arbitrage opportunities. There are two types of arbitrage that can occur which are detailed below. It is assumed that there are k asset returns for each of the s generated scenarios which are summarized in the s x k matrix **S**. In addition, it is assumed that the initial price of all assets is one.

Firstly, arbitrage occurs in a situation where at the investment horizon the portfolio has a non-negative value under any possible scenario but generates an initial positive cashflow. In this case arbitrage opportunities are found by optimizing the below problem.

$\min_{w} w\mathbf{1}, \mathbf{S}w \geq 0$ where 1 is a vector of ones representing the initial asset prices and w is the vector detailing the investment in each of the assets. If the objective

is negative then the investor receives an initial positive amount and arbitrage exists.

An example of arbitrage is given when $S=\begin{pmatrix}1.1 & 1.15\\ 0.9 & 0.95\end{pmatrix}$ i.e. a two risky asset problem where the second asset outperforms the first asset in both of the two possible scenarios. By selling short 1000 units of the first asset and buying 956.53 units in the second asset an investor receives an initial positive cashflow of 43.37. Whichever one of the two scenarios occur the portfolio value will be a non-negative amount, i.e. $\begin{pmatrix}1.1 & 1.15\\ 0.9 & 0.95\end{pmatrix}\begin{pmatrix}-1000.00\\ 956.53\end{pmatrix}=\begin{pmatrix}0.01\\ 8.70\end{pmatrix}$.

Secondly, arbitrage exists when a portfolio costs nothing to set up and the portfolio value at the investment horizon is non-negative in all the possible states and is strictly positive in at least one state.

As the number of assets and scenarios rises the probability of arbitrage opportunities existing decreases. Therefore, if arbitrage opportunities are found to exist, more scenarios should be generated until arbitrage is no longer possible. The limitation of long-only portfolios will also avoid the situation of unbounded optimal solutions where the arbitrage profit is unlimited. Arbitrage opportunities may also exist when derivatives are priced using a probability distribution that is inconsistent with that used to generate the asset returns.

2.2.2. Scenario optimization

Below I describe how scenario optimization provides an intuitive alternative to mean variance optimization, by addressing the limitations of mean-variance optimization mentioned above and the additional flexibility possible.

Addressing the limitations of mean-variance optimization

Scenario optimization provides a more realistic modelling framework than mean-variance analysis by allowing for non multivariate normal asset returns and

a choice in the specification of the utility function or risk measure used. Scenario optimization is particularly suitable when considering asset classes with non linear payoffs such as derivatives, when the model specification of appropriate return distributions and risk measures are likely to have a more significant impact on the investment strategy as compared to mean-variance analysis.

Scenario optimization allows for the uncertainty in model parameters by selecting the asset returns for each scenario from a particular probability distribution, although there is still a requirement to estimate the parameters that should be used in this probability distribution. The efficient frontier for all the simulated scenarios can then be calculated using stochastic optimization techniques. Rasmussen (2003) provides an example of how scenario optimization can result in an asset allocation which is much less sensitive to the input assumptions as compared with mean-variance optimization. Another advantage of scenario optimization is the specification of the efficient frontier ridge, which more realistically describes the optimal portfolio as a probability distribution of the efficient frontier rather than as a single line in return/risk space.

One may also argue that in contrast to mean-variance optimization, a stochastic scenario optimization approach may result in portfolio allocations that change more gradually as the level of portfolio risk is increased and also results in a more diversified portfolio for low and high levels of portfolio risk. It is likely that asset managers would be more comfortable implementing these more stable and diversified portfolio allocations that make better investment intuition than those produced by the mean-variance approach.

Flexibility of scenario optimization

Scenario optimization provides a flexible method for determining optimal financial decisions under uncertainty when several inputs with different and complex characteristics are employed simultaneously in a model, thereby providing a realistic model for portfolio allocation decisions. In addition, when

using scenario optimization it is possible to implement an objective function that is based on a particular risk measure.

Scenario optimization can utilize a wide range of statistical techniques in the process of estimating possible future asset return distributions. This may include the use of multivariate time-series analysis for the future asset return simulations, the use of Bayesian techniques to allow for data uncertainty by including prior return and correlation distributions, the use of robust statistics to reduce the impact of data outliers, or the use of adaptive non-linear techniques so that the data estimation process learns as time progresses.

Scenario optimization can be of benefit in active management by allowing an asset manager to include a detailed and complex specification of their views on the future return distributions of each asset classes which is not possible in the mean-variance setting. Scenario optimization also allows the fund manager to specify how they believe an asset class return distribution will evolve over time (e.g. a certain asset class may be forecasted to have a strong, neutral and then poor return performance over consecutive time periods 1, 2 and 3 respectively). Scenario optimization can also be used to determine both an optimal short term tactical investment strategy as well as the optimal long term strategic investment strategy.

Scenario optimization has added benefits when one wishes to perform a surplus optimization, where the level of surplus is defined as the value of the assets minus the value of the liabilities. In contrast to portfolio optimization where one wants to determine the optimal asset allocation in an asset-only modelling framework, surplus optimization also requires a framework to model liabilities. The method of scenario optimization can provide a flexible and realistic way to perform surplus optimization through the consistent statistical projections of assets and liabilities.

2.3. Methods for Multistage Portfolio Optimization

Mulvey & Ziemba (1995) and Mulvey (2001) detail some of the alternative approaches for solving dynamic multi-stage financial optimization problems.

Multi-stage stochastic programming

This method involves a large multi-stage stochastic program in order to determine the optimal investment at each time period as a function of the outlook for the assets in comparison with the investor's circumstances. Conditional decisions are made at each node of the scenario tree subject to the modelling constraints. One needs to be able to split the problem into a number of myopic sub-problems, so that each sub-problem can be solved separately. This approach has been developed by Kallberg, White & Ziemba (1982) and Kusy & Ziemba (1986). This method can address real-world features such as turnover constraint, transactions costs, risk aversion and limits on the groups of assets. It requires highly efficient solution algorithms due to the enormous number of decision variables. The disadvantage of this approach is the restrictive nature of the assumptions that can be specified in order to ensure that the problem remains manageable. Mulvey & Vladimirou (1992) used stochastic network programming to solve a multiperiod portfolio optimization problem.

Using decision rules to optimize Monte Carlo Simulations

The portfolio allocation at each time period can be determined by a decision rule function which takes the form

$$x_{j,t} = h(a^s_{j,t}, b^s_{j,t}, ...) \tag{2.1}$$

where $x_{j,t}$ is the investment in asset j at time t and the parameters a, b,... describe the driving factors. An example of a decision rule is the fixed-mix strategy, where the asset allocations of each asset class are rebalanced in line with fixed allocations at each time period. The best fixed-mix proportions can then be determined over the set of all simulated scenarios. Mulvey & Chen (1996) use

this approach to show that a fixed-mix strategy results in improved returns and reduced risk as compared to a buy and hold strategy.

Decision rules are relatively easy to implement and can be optimized without the need of large scale linear or non-linear programs. It also has the advantage that confidence limits can be provided on the recommendations as the method can be tested with out-of-sample scenarios. However, they often result in non-convex models where the global optimal solution is difficult to identify.

Stochastic control methods

This method forms a mesh over the state space for the driving variables at each time period, rather than discretizing the scenarios. This approach has been used by Samuelson (1969) and Merton (1992). Dynamic programming algorithms or finite element algorithms can then be used to solve this problem. This is a general purpose method that can be used when the state space can be kept to a limited number of driving factors.

Heuristic methods

There are many combinatorial optimization problems where it is computationally intractable to find exact optimal solutions, known as NP-hard. In such cases, heuristic methods can be used to find near-optimal feasible solutions to complex nonconvex optimization problems through adaptive stochastic search algorithms. Optimization problems often result in multiple local optima and discontinuities, which especially occur when non standard constraints are imposed. The main advantage of a heuristic approach is the high degree of flexibility possible when specifying the asset return distributions, objective function and required constraints. Heuristics are also relatively easy and quick to implement. In the next section I carry out a review of heuristic approaches that have been used to solve portfolio optimization problems.

2.4. Literature Review on Heuristic Approaches to Portfolio Optimization

Arnone, Loraschi & Tettamanzi (1993) use a genetic algorithm to solve an unconstrained portfolio optimization problem in a downside risk framework. Loraschi & Tettamanzi (1995) and Loraschi, Tettamanzi, Tomassini & Verda (1995) use an evolutionary synchronous stepping stone island model to solve a portfolio selection problem. Each risk aversion parameter represents as island where individuals are periodically allowed to migrate between adjacent islands. The reasoning for this is that a good solution under one risk aversion parameter is likely to remain a good solution under a similar adjacent risk aversion parameter. They show that this approach outperforms a sequential algorithm approach.

Glover, Mulvey & Hoyland (1995) gave an example of how Tabu search can be used for portfolio optimization when the portfolio is periodically rebalanced in order to maintain fixed asset allocation weights. Hiemstra (1996) showed how short-run fluctuations in excess returns and volatility change the standard optimization problem into that of a harder tactical asset allocation problem which can be effectively solved using genetic algorithms.

Speranza (1996) solved a cardinality constrained portfolio optimization problem by combining a mixed-integer linear programming model with a heuristic algorithm within a negative mean absolute deviation risk framework. Mansini & Speranza (1996) then used this approach to solve the problem with minimum transaction lot constraints. Mansini & Speranza (2005) solve a single-period mean-safety portfolio optimization problem with transaction costs and integer transaction round constraints. Their method requires the solving of linear and mixed-integer programming problems where the initial problem is partitioned into two subproblems and a simple local search heuristic is used to find an initial solution.

Chang, Meade, Beasley & Sharaiha (2000) use a heuristic approach to find the efficient frontier in a mean-variance portfolio optimization model. Their

model includes cardinality constraints, which limits the number of asset that can be held at any one time, as well as asset allocation weight constraints. The cardinality constraint may cause the efficient frontier to be discontinuous. As these practical constraints cannot easily be dealt with by quadratic programming, the methods used to solve this problem are genetic algorithms, tabu search and simulated annealing. They find that no method proved to be superior and concluded that a sensible approach is to pool the results from the three algorithms. Beasley, Meade & Chang (2003) use an evolutionary population heuristic to solve the index tracking problem (i.e. tradeoff between tracking error and excess return) where there is a constraint limiting the number of stocks that can be held. Their formulation also includes transaction costs and a limit on the total transaction costs that can be incurred.

Schlottmann & Seese (2004) combine quantitative models with genetic algorithms for the purpose of optimizing credit portfolios. Their memetic algorithm is found to reduce the possibility of becoming stuck in local optima. Kellerer & Maringer (2001) and Maringer & Kellerer (2003) use a hybrid local search algorithm that combines simulated annealing and evolutionary strategies to solve a cardinality constrained portfolio optimization problem. They find this an efficient, reliable and flexible approach. Streichert, Ulmer & Zell (2003) use an evolutionary algorithm for a cardinality constrained portfolio optimization problem. They show that using a local search heuristic to convert infeasible solutions into a feasible solution improves the performance of the evolutionary algorithm.

Gilli & Kellezi (2002a) use threshold accepting for portfolio optimization in a downside risk framework. Classical optimization methods are shown to be ineffective as the specified model constraints result in a complex optimization problem that has multiple local optima and discontinuities. Gilli & Kellezi (2002b) show that threshold accepting is an efficient optimization method to solve an index tracking portfolio optimization problem of minimizing the tracking error between a portfolio and its benchmark. They include the

constraints that the number of stocks allowed in the index must be less than number of stocks in the benchmark index and that transaction costs are incurred every time a rebalancing is made.

Ehrgott, Klamroth & Schwehm (2004) optimize a mixed-integer portfolio optimization problem using neighbourhood search, Tabu Search simulated annealing and genetic algorithms. They showed that for a large real case example the genetic algorithm method produces the best and most stable results.

Derigs & Nickel (2004) use a 2-phase simulated annealing heuristic to solve an index tracking problem. This involves minimizing the portfolio tracking error, subject to transaction costs, cardinality constraints, floor/ceiling constraints and bundle constraints. In the first phase one generates a feasible portfolio and then in the second phase one improves on this solution in order to find a near-optimal portfolio. Their method is found to provide good quality solutions that meet investment guidelines.

Tang & Wang (2004) solve a dynamic portfolio selection problem with probability criterion by embedding a genetic algorithm and neural network within dynamic programming. The criteria function is either the sum of the probability that the portfolio return over each period is not less than a given expected return or that the whole portfolio return is not less than a given expected return.

Chen, Zhang & Zhang (2006) show the benefits of using an adaptive genetic algorithm over a standard genetic algorithm when solving a mean-variance portfolio selection problem.

Fernandez & Gomez (2007) use the Hopfield neural network to solve a portfolio selection problem in a mean-variance framework that includes cardinality and bounding constraints. They compare the results to evolutionary algorithms, tabu search and simulated annealing and conclude that none of the four heuristic methods outperforms the others in all kinds of investment policies.

There have only been a limited number of papers on the use of heuristic methods for solving multi-stage portfolio optimization. Chi-cheong, Cheung,

Tang & Man-chung (2002) use a micro-genetic algorithm to solve a multi-stage portfolio optimization problem in a mean-variance framework. The micro-genetic algorithm uses a small population size to shorten the required computational time. Once the population loses its diversity it is randomly restarted with elitism used to maintain the best individual. Baglioni, Costa Pereira, Sorbello & Tettamanzi (2000) use an evolutionary algorithm to solve a multiperiod asset allocation problem. They describe how the financial regulation constraints of a pension fund can result in a more complicated objective function and constraints which cannot easily be dealt with using a quadratic optimization problem but can be easily optimized using a genetic algorithm.

More recently a number of papers have used multiobjective evolutionary algorithms to solve portfolio optimization problems.

Lin, Wang & Yan (2001) use a multiobjective evolutionary algorithm to solve a portfolio selection problem with fixed transaction costs and minimum transaction lots in a traditional mean-variance framework. The method is based on Deb, Pratap, Agarwal & Meyarivan's (2002) evolutionary multiobjective NSGA2 algorithm and Michalewicz & Janikow's (1996) GENOCOP algorithm. The GENOCOP is used to improve the algorithm's efficiency by ensuring that a feasible starting solution that undergoes genetic operators will remains feasible in the case of linear constraints.

Mukerjee, Biswas, Deb & Mathur (2002) use the multi-objective evolutionary algorithm NSGA2, to optimize the risk-return trade-off for a bank loan portfolio management problem. They show this multi-objective method of finding the Pareto frontier, i.e. the optimal tradeoff between risk and return, in a single run is a more computational efficient than solving the equivalent multiple single objective optimization problem.

Doerner, Gutjahr, Hartl, Strauss & Stummer (2004) use a Pareto ant colony optimization to solve a portfolio selection problem. They show that this

algorithm performs more favourably than either the Pareto simulated annealing or the non-dominated sorting genetic algorithm approach.

Fieldsend, Matatko & Peng (2004) use an evolution strategy to solve a cardinality constrained portfolio optimization problem. They use a parallel multiobjective evolutionary algorithm in order to simultaneously find a number of cardinality constrained efficient frontiers. This involves maintaining a Pareto frontier for each cardinality k, and using information from each of the Pareto frontiers to improve the search of the other fronts. This approach benefits from the increased efficiency of parallelization and is particularly useful in situations when an investor does not know as a priori the exact number of assets that they want in their portfolio.

Armananzas & Lozano (2005) use a multiobjective evolutionary algorithm to solve a portfolio optimization problem. The three optimization methods they use are greedy search, simulated annealing and ant colony optimization. They find that the simulated annealing and ant colony approach outperforms the greedy search.

Subbu, Bonissone, Eklund, Bollapragada & Chalermkraivuth (2005) develop a hybrid multiobjective evolutionary algorithm that integrates a multiobjective evolutionary algorithm with linear programming. They show that this approach can be used to solve a portfolio optimization problem with multiple measures of risk and return, where these measures may be nonlinear and nonconvex. A randomized linear programming finds initials potential solutions and a Pareto Sorting Evolutionary Algorithm and Target Objective Evolutionary Algorithms are then used to generate a Pareto frontier. The multiobjective optimization approach is that of iteratively integrating the search and multi-objective decision making in order to quickly find a small subset of efficient portfolios.

Diosan (2005) use three multiobjective evolutionary algorithms, Nondominated Sorting Genetic Algorithm 2 (NSGA2), Pareto Envelop-based Selection Algorithm (PESA) and Strength Pareto Evolutionary Algorithm 2

(SPEA2) to solve a portfolio optimization problem. The PESA algorithm is found to give the best Pareto frontier and the NSGA2 algorithm gives solutions that are most evenly spread out along the Pareto frontier.

Ong, Jih-Jeng & Tzeng (2005) use a multi-objective evolutionary algorithm to solve a portfolio optimization problem in the case of a small sample situation. They have three objectives of maximizing return, minimizing uncertainty risk (the measure of the possibilistic degree of the return rate) and minimizing relation risk (the measure of the trending degrees of the sequences). Numerical results show that this method can provide more flexible and accurate results.

Streichert and Tanaka-Yamawaki (2006) consider the effect of local search on the constrained portfolio selection problem. They use two algorithms, firstly the NSGA2 multiobjective evolutionary algorithm and secondly, a memetic algorithm that is a hybridization of this first algorithm with a quadratic programming local search heuristic. Memetic algorithms are a class of evolutionary algorithms that use local search or problem specific information. Both algorithms are also carried out with and without a Lamarckism evolution. They conclude that although their memetic algorithm may result in improved solution, it was also more likely to suffer from premature convergence. In both their algorithms, the inclusion of Lamarckism evolution was found to speed up initial convergence and worsen the algorithm performance due to premature convergence.

2.5. Conclusion

I have discussed the limitations of using a traditional mean variance approach when solving portfolio optimization problems and have explained why I am going to solve my optimization problem using a scenario generation of assets and a heuristic optimization method. This approach allows the realistic modelling of the assets and flexibility in the model constraints and risk measure

specification. By carrying out a literature review, I have also detailed heuristic methods that have been used to solve a variety of portfolio optimization problems.

Chapter 3

Heuristic Optimization Methods and Evolutionary Algorithms

Heuristic optimization methods have been widely used for solving a range of financial problems such as financial forecasting, financial trading and portfolio management. Heuristic optimization methods can be used to solve hard problems, where it is difficult to find local optima quickly and reliably. Another advantage of heuristic optimization methods are that they benefit from the ability to easily hybridize with other techniques. Heuristic methods are particularly useful when there is a well defined model and objective function where alternative soft computing methods (e.g. fuzzy logic or neural networks) are less appropriate.

The disadvantages of heuristic optimization are that solutions found are only near optimality and it is difficult to verify the optimality of a solution. Choosing the form of the problem representation, selection method and genetic operator probabilities can also be difficult and require a considerable level of judgment.

For the multistage portfolio optimization problem, the main advantage of using a heuristic approach over the use of an exact optimizer algorithm is the possible flexibility in the model specification of the asset return distributions, objective function and model constraints. A heuristic approach is needed if the objective function is nonconvex as the existence of many local maxima will make it difficult for the optimizer algorithm to find the optimal solution.

The heuristic approach I am focusing on is evolutionary algorithms due to them being a flexible less problem dependent method. They are more likely to reach a global optimum as it considers many potential solutions simultaneously

and has the unique crossover operator that makes it possible to merge the best attributes of current solutions.

3.1. Heuristic Optimization Methods

3.1.1. Single-Solution Heuristic Methods

Local search

Consider an optimization problem of the following form $\max_{x \in F} f(x)$. F is the set of solutions x that satisfy all the given constraints and is called the feasible region. A feasible solution $x^* \in F$ is called optimal if $f(x^*) \geq f(x)$ for all $x \in F$. Local search algorithms try to improve on a feasible solution x by applying a set of local perturbations within a defined neighbourhood n(x). The initial solution will be replaced with a better solution in its neighbourhood until no better solution is found and a termination criteria is met. Therefore, given a current solution $\bar{x}$, the local search algorithm searches for an alternative solution x^0 such that $f(x^0) = \max_{x \in N(x)} f(x)$. (3.1)

If $f(x^0) > f(\bar{x})$, then x^0 will be set as the new solution and the process will be iterated. If both $f(x^0) \leq f(\bar{x})$ and the termination criteria is met, then the search algorithm will be stopped. This last solution is then said to be locally optimal. This process is then repeated for many initial starting values of x, which may be chosen randomly from the search space, with the best locally optimal solution taken as the global optimal. A good introduction to local search can be found in Yagiura & Ibaraki (2002).

The three main components of a local search algorithm are detailed in Eiben & Smith (2003).

- Move strategy - details the order in which solutions in the neighbourhood are to be covered. Two move strategies are the first admissible move strategy (also called greedy ascent) and the best admissible move strategy (also called

steepest ascent). In the first admissible move strategy the evaluation of solutions in n(x) are calculated in some pre-specified order, and the first improved solution found is immediately accepted as the new solution. The order solutions in n(x) are tested may affect the quality of the solution and the computational time. In the best admissible move strategy the evaluation for all the solutions in n(x) are calculated simultaneously and the best solution is taken as the new solution. If the neighbourhood is large one may only wish to search a small randomly drawn sample from all the individuals in the neighbourhood.

- Depth of the local search – this is the number of improving steps which are to be taken before the algorithm is terminated
- Neighbourhood n(x) – this details how to generate candidate solutions from the current solution. For a portfolio optimization problem an appropriate definition of the neighbourhood may be solutions whose absolute or relative asset weights for individual assets or asset classes are similar. When choosing the size of the neighbourhood, the improved quality of the local optimal solution found from using a larger area needs to be offset against the increased computationally time searching this larger area will incur.

The effectiveness of the local search method may be explained by the 'proximate optimality principle' (POP), see Glover & Laguna (1997). The concept is that good solutions often have similar structures. Therefore, it is important that the neighbourhood is defined in such a way that it is able to find the hidden structures common to good solutions.

The pseudocode for a local search based on Eiben & Smith (2003) is detailed below.

Input: a starting solution i, fitness function f(x), and a neighbourhood function n(x)
Output: the final improved solution i
Function LocalSearch(i , f(x), n(x))
    best = i
    iterations = 0
    Repeat Until (depth condition is satisfied)
        count = 0
        // depth condition
        // either a fixed number of iterations
        // or until local optimally is reached: count = |n(i)| and best = i
        Repeat Until (move operator is satisfied)
            // move operator
            // for steepest ascent: count = |n(i)|
            // for greedy ascent: count = |n(i)| or best ≠ i
            Generate the next neighbour j ∈ n(i)
            count = count + 1
            If f(j) > f(best) **Then** best = j **End If**
        End Repeat
        i = best
        iterations = iterations + 1
End Repeat
Return i
End Function

The main drawback with the local search approach is that the optimal solution found is likely to be one of several local optima rather than the global optimum due to the specification that only improving perturbations, where $\Delta f > 0$, are accepted. Metaheuristics allow the move strategy in the local search to allow new non-improving solutions in order to escape from a locally optimal solution. Below, I consider common single-solution metaheuristic methods.

Simulated annealing

Simulated annealing, also called stochastic hill climbing, addresses the problem local improvement has, of not being able to escape from local optima, by allowing nonimproving perturbations in a specified way, see Kirkpatrick, Gelatt & Vecchi (1983). When trying to maximize the objective function, a perturbation

of the current solution where $\Delta f > 0$ will always be accepted and when $\Delta f < 0$ will be accepted with a probability

$$P(\Delta f, T) = \exp\left(\frac{\Delta f}{T}\right). \tag{3.2}$$

Therefore, the probability of accepting a nonimproving perturbation decreases for lower values Δf and higher values of T. When setting the value for T one has to consider the tradeoff between the need to exploit the current solution by improving it and the need to sufficiently explore the solution space. The temperature T at iterative step k can be specified by the following simple cooling schedule

$$T_k = \alpha T_{\alpha-1}\, 0 < \alpha < 1, \tag{3.3}$$

although it is usual to keep the temperature constant for the first few steps. The method is *first-improving* in that the entire neighbourhood of the current solution does not have to be searched before a potential solution is accepted. A rejection of a potential solution results in the selection of another potential solution in the neighbourhood of the current solution. The iterations are repeated until the termination criteria are satisfied. The termination criteria could depend on the permitted maximum number of iterations, the minimum temperature or the maximum number of steps in which the current solution is unchanged.

Threshold accepting

Threshold accepting is a deterministic analog to simulated annealing that was introduced by Dueck & Scheuer (1990). It is a local search procedure which escapes local optima by accepting new solutions that are not worse by more than a specified threshold. Both the number of iterations and the number of solutions explored at each iteration are fixed at the start. The threshold is decreased each iteration until it reaches zero for the last iteration.

Tabu search

Tabu search is a neighbourhood search based algorithm that aims to avoid being trapped in local optima while biasing the search towards good solutions and is described in Glover (1986) and Hansen (1986). Entrapment in cycles is avoided by excluding solutions which have been previously visited. Past solutions are identified by keeping a tabu list, which summarizes some key attributes of recent solutions. In a tabu search the best solution that is in the entire neighbourhood of the current solution and has not already been visited is accepted, even if this solution is a non-improving perturbation. The reasoning for this approach is that it is preferable to accept a new poorer solution in order to avoid revisiting a solution that has already been investigated. This helps to ensure that new regions of the solution space are explored. Berger, Glover & Mulvey (1995) used the Tabu search method to find the optimal asset allocation and found that this method resulted in significantly improved computation performance compared with the interior-point method.

3.1.2. Population Heuristics

Single solution heuristics, which I have just considered, work on improving a single current solution only. In contrast, population heuristics combine a number of current solutions in order to generate a population of new solutions. Population heuristics are based on intelligent probabilistic search algorithms where the evolutionary process increases the quality of the population over time. Beasley (2002) states "There is something about a population heuristic that means it is often capable of producing better quality solutions than single-solution heuristics". A disadvantage of population heuristics over single-solution heuristics is the often longer time it takes to implement. Examples of population heuristics are scatter-search algorithms, bionomic algorithms and evolutionary algorithms.

Scatter search

Scatter search is an evolutionary population based algorithm that combines existing solutions to produce new solutions. The scatter search method involves the following steps.

(i) Generate a starting set of diverse solutions. Apply a heuristic procedure to improve the starting solutions. Place the best solutions in a solution reference set. Best solutions are measured according to both the level of the objective value as well as the amount solutions add to the diversity of the reference set.

(ii) Produce new solutions as a weighted linear combination of the current reference set, with higher fitness solutions typically having higher weights. Unlike in standard genetic algorithms, the number of parents can be more than two. In order to keep the number of possible linear combinations of reference set members to a manageable level the reference set typically has at most 20 solutions.

(iii) Apply a heuristic procedure to ensure that the solution is feasible and improve the fitness of the solution.

(iv) Update the set of reference solutions with the best new solutions.

Steps (ii)-(iv) are then repeated until the set of reference solutions no longer increases or the target number of iterations have been reached. More details on the scatter search method can be found in Laguna (2002) and Glover, Laguna, & Marti (2003).

Bionomic algorithm

The bionomic algorithm was introduced by Christofides (1994). It uses a generational approach to combine new solutions from linear combinations of old solutions and can also make use of problem specific knowledge. The bionomic algorithm gives a better fitness individual greater probability of becoming a parent while preventing individuals that are too similar becoming parents simultaneously. The bionomic algorithm involves the following steps:

(i) Generate an initial population. Mature each individual member using either a problem-specific heuristic or any generic local search approach.

(ii) Parent selection. Use the below structured approach to construct parent sets from the current population:

a) Set F_i, the inclusion frequency, as the rank of each population member i, so that the highest fitness member in a population of size N has an F value of N. Set a distance threshold Δ that will be used to access whether two individual members are close enough to be considered similar. This parameter controls the degree of diversification in the population. Set Q, which contains the parent sets, as empty.

b) Generate a graph G that contains one node for each individual with $F_i \geq 1$ and an edge between any two nodes whose distance measure $\leq \Delta$.

c) Generate a parent set P as a maximal independent set of G. This is a set that contains no adjacent nodes where the set cannot be enlarged. Then set $Q = Q \cup P$. The reasoning behind this approach is that if parents are similar, the linear combination may result in solutions that are local to the parents. These regions of search space should have already been explored during the maturing of the previous children.

d) For all individuals where $i \in P$, set $F_i = F_i - 1$. Remove from G any nodes i where $F_i = 0$. If G is not empty go back to part (c).

e) Q now consists of N parent sets of varying sizes. All nodes should now have been selected in a parent set the same number of times as their initial inclusion frequency.

(iii) Child generation. Define an operator, e.g. weighted linear combination, that will generate one child from each parent set. For each parent set $P \in Q$, decide on the number of children that should be generated, and apply the operator to generate the required number of children. Mature each child. Replace a number of the worst population members with the best children. The number of solutions replaced at each generation will affect the speed of convergence.

3.2. Evolutionary Algorithms

3.2.1. Introduction to evolutionary algorithms

An evolutionary algorithm is a generic term used to describe any population based metaheuristic optimization algorithm that uses mechanisms inspired by biological evolution such as recombination, mutation and survival of the fittest. The aim of an evolutionary algorithm is to gradually improve the quality of a population of solutions until the optimal solution dominates the population. The evolution process is driven by the selection process which favours fitter solutions, and by variation operators (used to create new solutions from old solutions using recombination and mutation) which are used to facilitate the discovery of new fitter solutions. Good descriptions of evolutionary algorithms can be found in Back, Fogel & Michalewicz (1997) and Eiben & Smith (2003). The two main types of evolutionary algorithms are

- Evolution strategies developed by Rechenberg (1965) and Schwefel (1965). Evolution strategies modify real-valued vectors by typically using a normally distributed mutation operator with recombination being a secondary operator. Evolution strategies use random uniform (not biased by fitness) parent selection methods and deterministic survivor selection methods that are biased by fitness.
- Genetic algorithms developed by Holland (1975). Genetic algorithms emphasize recombination as the main search method with mutation considered a background operator. Stochastic parent selection methods, biased by fitness, are used and the representation of individuals is typically of a binary form. Survivor selection can either be generational, where all individuals are replaced, or steady-state, being deterministically biased by fitness. A good introduction to genetic algorithms can be found in Beasley, Bull and Martin (1993a and 1993b).

These two methods have greatly influenced each other and many ideas from each method have been applied to the other method. Differences between evolution

strategies and genetic algorithms can be found in Whitley (2001), Back, Hoffmeister & Schwefel (1991) and De Jong (2004). I will be using evolutionary algorithms, utilizing techniques developed from both evolution strategies and genetic algorithms, in order to solve the multistage portfolio optimization problem.

Stages in an evolutionary algorithm

Stages of an evolutionary algorithm approach are detailed below.

(i) *Representation.* A potential solution needs to be represented as a set of parameters. In genetic algorithms solutions are represented by their genetic makeup, called genotype. Each parameter value, called a gene, is represented as a binary string. A solution which comprises of several parameter values is then made up by joining these genes together to form a string. While genetic algorithms tend to use a binary rather than continuous representation, evolution strategies use a more direct real-valued phenotypic representation that expresses a solutions behavioural trait.

Fogel & Stayton (1994) found that there is no compelling reason, and it may actually be harmful, to use a binary representation with a large number of schemata when performing a real-valued optimization. Furthermore, Janikow & Michalewicz (1991) showed that floating point representation gave faster, more consistent and precise results. As the variables in my problem are continuous asset allocation weights it is simpler to represent them as floating-point numbers rather than having to convert them into a string of binary numbers.

(ii) *Creation of the initial population.* One needs to choose a set of population members that represent solutions to the given problem. In order to reach a robust solution the initial population should be diverse meaning that individual solutions are not too similar and the population spans a wide range of variable values. Evolution strategies typically have a population

size of up to 20 individuals compared to a typically value of 100 for genetic algorithms.

Alternative methods for creating the initial population are

- totally random initialization
- having some initial seeded solutions that either come from problem specific information or are the best solutions from multiple runs of a single point heuristic
- selective initialization – a large number of random solutions are created and the initial population selected according to the level of fitness and diversity over the search space (to avoid clustering).

(iii) *Evaluation.* Fitness of each population member is calculated according to the chosen objective function which is going to be optimized. The fitness function should be smooth and regular in order that solutions with a reasonable fitness have similar gene values to solutions with an improved fitness value. In addition the fitness function should be constructed in a way that it does not have too many local optima or a very isolated global optima.

(iv) *Parent Selection.* Evolutionary algorithms work by trying to improve the quality of the population of solutions over each successive generation. One way this is achieved is by making it more likely that good solutions will pass on their genetic makeup to the next generation, through the creation of offspring, than poor solutions.

(v) *Recombination* This is the stochastic operator which merges information from two parents to create an offspring. The principle behind recombination is that the offspring produced can combine the different desirable features of each of the parents.

(vi) *Mutation.* Random mutations of the population of the solutions are performed in order to give more opportunities for the population to reach good solutions and help to ensure population diversity.

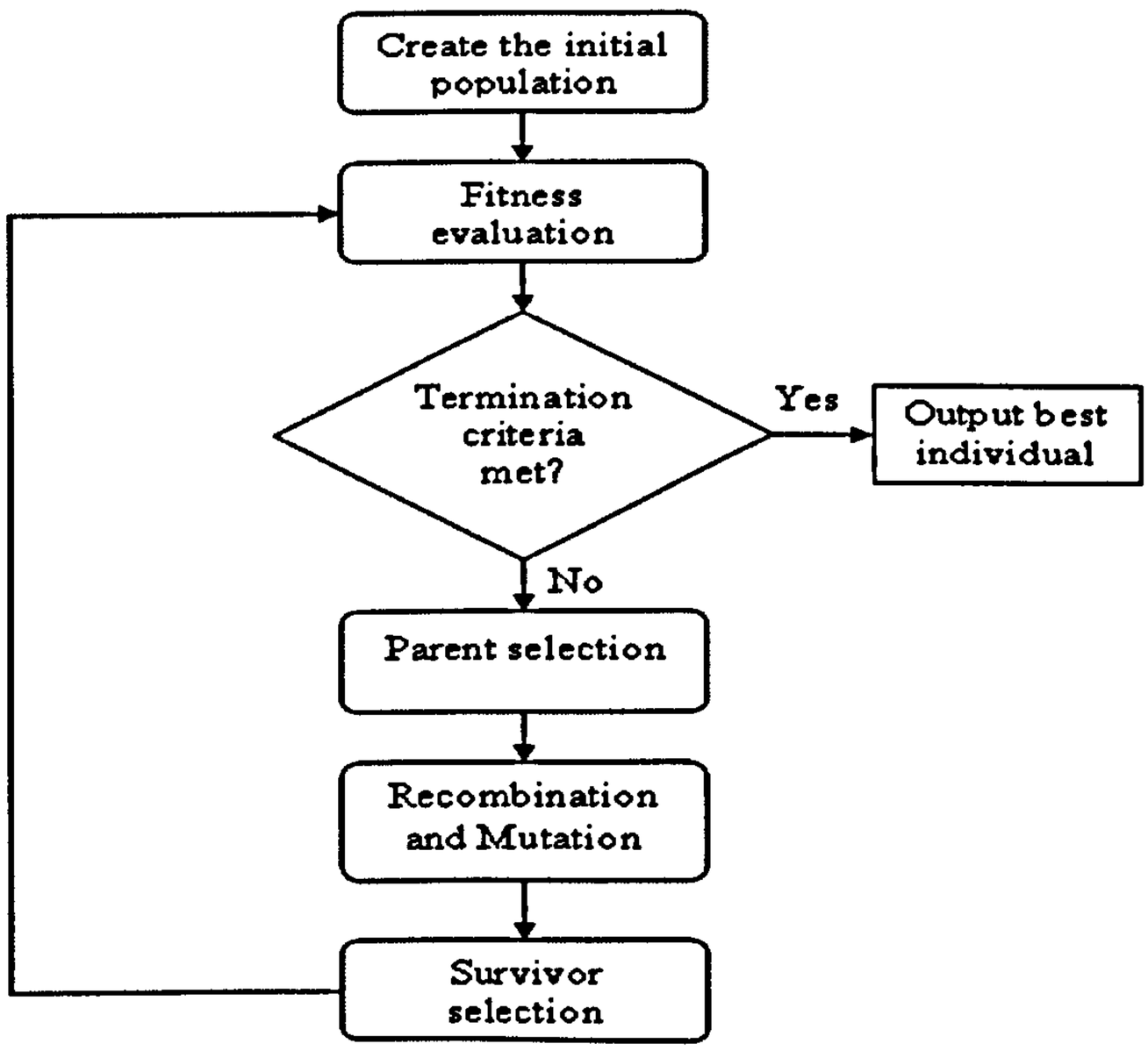
(vii) *Evaluation and survivor selection.* Once the processes of parent selection, recombination and mutation are complete, the resulting solutions are

evaluated. Survivor selection, which is required due to the fixed number of population members at each generation, is then used to determine which individuals will be kept for the next generation.

(viii) *Termination.* This above process of evaluation and applying selection and variation operators are repeated until the termination criteria are met. Termination criteria may be based on

- fitness improvement over a given number of generations remaining under a specified threshold level
- adequate population convergence (i.e. population diversity falls below a specified threshold level). For example in a binary representation, convergence is typically said to occur in a gene when 95% of the entire population share the same value. Population convergence then occurs when each gene in the population has converged.
- passing the maximum allowed number of generations, time or solution fitness evaluations.

Figure 3.1: Flow diagram of a basic evolutionary algorithm



The general form of the pseudocode for an evolutionary algorithm is:

```
Function EvolutionaryAlgorithm()
    Initialize Population
    Evaluate each solution
    Repeat Until (termination condition is satisfied)
        Select parents
        Recombine parents to produce offspring
        Mutate the offspring
        Evaluate the new population
        Select individuals for the next generation
    End Repeat
    Return Population
End Function
```

Why Genetic Algorithms work

Holland's (1975) Schema Theorem explains that Genetic algorithms work because they implicitly search for schemata, parts of strings, which have a higher fitness level. With a binary digit string representation each position in the string can take the value 0 or 1. In addition the "don't care" symbol # is used to denote when the digit could either have a value of a 0 or 1. Schemata are strings that include the # symbol. This means that in a 6-bit string there are $2^6 = 64$ individual strings and $3^6 = 729$ schemata. For example, the six digit string 00#11# matches the following four strings {000110, 000111, 001110, 001111}. The power of genetic algorithms stems from the fact that every time the fitness of an individual string is assessed a significantly larger number of schemata are implicitly assessed. This results in the genetic algorithm quickly being able to identify good schemata. Holland showed that the optimal method for exploring the search space is to choose parents for reproduction in proportion to that individual's relative fitness level. In contrast to genetic algorithms, evolution strategies have no notion of schema processing

Exploration and exploitation

An efficient evolutionary algorithm needs to find the right balance between exploitation (taking advantage of information already obtained) and exploration (searching for better solutions in new areas of the search space).

Evolutionary algorithms are able to effectively combine exploration and exploitation in a way that random search (only uses exploration) and hillclimbing (over concentrated on exploitation) are unable to achieve. Kingdon & Feldman (1995) showed that the balance between exploration and exploitation is greatly affected by the choice of the selection method.

In a population of finite size stochastic errors can cause genetic drift, as described in Booker (1987). This occurs if by chance a gene becomes more dominant in a population over several generations, until it has spread to the entire population. Once this occurs crossover is unable to change this gene's value which can only be changed by mutation. Therefore, increasing the mutation rate decreases the rate of genetic drift. In order for the evolutionary algorithm to effectively work the degree to which differentials in the fitness value is exploited needs to exceed the degree to which genetic drift occurs. Therefore, genetic drift is more likely to occur when the fitness function is relatively flat.

Population convergence

A classical problem that may occur in evolutionary algorithms is that of premature convergence of the population. This occurs when the population convergences to a local rather than a global optima. As a population becomes closer to converging, the offspring produced by crossover will become more and more similar to the parents. Mutation is then the only way to explore the search space, which is equivalent to a slow random search. In the case of genetic algorithms, the schema theorem says that the reproductive opportunities each individual gets should be in proportion to their relative fitness. However, in order for evolutionary algorithms to work for populations of finite size one needs to control the number of reproductive opportunities each individual gets through the choice of an appropriate selection method.

The converse problem to premature convergence is slow finishing. After a large number of generations the population will have almost converged. The average population fitness will be high, with a relative small difference between

the average and best fitness level. This means that there is an insufficient fitness differential to precisely locate the most optimal solution.

Speed of convergence can be measured by the

- average reduction in population variance
- change of the average fitness of a population as a ratio to the population variance
- mean time required for the population to converge to the best solution.

3.2.2. Selection Methods

In this section I will consider parent selection methods and survivor selection methods. Evolutionary algorithms can use selection methods that are either deterministic or stochastic. Deterministic selection will always select the same individuals from a particular population by ranking the population according to fitness and then replacing the worst solutions with the best solutions. This is easy to execute and has fast convergence but may fail to adequately search the solution space.

Beasley, Bull and Martin (1993a) divide parent selection methods into explicit fitness remapping and implicit fitness remapping. In explicit fitness remapping the fitness value of each individual is mapped onto a new scale which is used to determine the number of reproductive trials. In a generational evolutionary algorithm the average number of reproductive trials allocated to each individual needs to be one in order to keep the population the same size. Implicit fitness remapping achieves a similar effect without having to go through the intermediate stage of calculating an adjusted fitness value. In evolution strategies μ denotes the number of parents and λ the number of offspring that are produced each generation before selection is carried out.

Explicit fitness remapping

Selection is carried out on the current generation in order to produce an intermediate generation. One can then apply crossover and mutation on the intermediate population in order to create the next population. One generation step consists of moving from the current to next generation. The following methods can be used to construct the intermediate population from the current population.

1. *Proportionate selection:* Each individual solution is assigned a probability of being selected that is in proportion to their relative fitness. Each time we select a parent, the probability that each solution is chosen as the parent is equal to the proportion that the particular individual makes up the total fitness value. The probability of selecting the i^{th} solution is given by

$$P_i = \frac{F_i}{\overline{F}}, \tag{3.4}$$

where F_i is the fitness of the i^{th} solution and $\overline{F}$ is the mean fitness for all the current solutions in the population. One solution may be selected more than once and there are no guarantees that the current best solution will be chosen as a parent. This method can result in selection that too strongly favours the fittest individuals in the early generations. As more generations pass and the fitness of the population improves, the variation in fitness is likely to fall too significantly, causing a reduction in the selection pressure.

2. *Fitness scaling* - remaps raw fitness values into adjusted fitness values. The maximum number of reproductive trials that an individual can have is set at a specified level, typically 2. The way this is achieved is by subtracting the following fixed amount from the raw fitness value and then dividing by the average adjusted fitness value.

Max number of reproductive trials × average fitness value – max fitness value. (3.5)

BIBL LONDON UNIV.

When using this approach one needs to ensure that negative fitness values are not generated. Consider the example where the average raw fitness is 6, the maximum raw fitness is 7.8 and the maximum number of reproductive trials is set as 2. The ratio of the maximum raw fitness value to average raw fitness is 1.3. The fitness scaling will subtract 4.2 from each raw fitness value and divide the answer by 1.8. The maximum adjusted fitness value is 2 and the average adjusted fitness is 1 giving the desired ratio of 2. The usual result of fitness scaling is to compress the range of fitness values early in the run reducing the chance of premature convergence. The problem with fitness scaling is that the degree of compression is decided by a single individual, that with the highest fitness value. The existence of one extremely highly fit individual will result in overcompression and a relatively flat fitness function where many solutions have an adjusted fitness value close to one.

3. *Fitness ranking* - overcomes the problem fitness scaling has of overcompression when there are extreme fitness values. Individuals are ranked in order of raw fitness and the reproductive trials are assigned according to rank. In linear ranking, a single value β_{rank} is set so that $1 < \beta_{rank} \leq 2$, where β_{rank} is the expected number of offspring that is allocated to the highest ranking individual. Then the highest ranked individual receives a fitness bias of β_{rank}, the median individual a fitness bias of 1 and the lowest ranked individual a fitness bias of $\alpha_{rank} = 2 - \beta_{rank}$. The selection bias between the best and worst individuals remains constant, being independent of the generation number. For a population of size μ, the selection probability for individual i is then given by

$$\Pr(i) = \frac{\alpha_{rank} + \frac{rank(i)}{\mu - 1}(\beta_{rank} - \alpha_{rank})}{\mu} \tag{3.6}$$

More aggressive nonlinear ranking methods are possible e.g. selection probabilities are set proportional to the square of the rank or using

exponential probabilities based on rank. These selection methods may lead to premature convergence.

4. *Threshold selection.* The best Tμ individuals, where $0 < T \leq 1$, are given a uniform probability of being selected with the remainder of the population being discarded.

$$\Pr(i) = \begin{cases} 0 & \text{if } rank(i) < (1-T)\mu \\ 1/T\mu & \text{otherwise} \end{cases} \tag{3.7}$$

Muhlenbein & Schlierkamp-Voosen (1993) found that a value of T in the range of 0.1-0.5 should be chosen. This method is closely related to the (μ, λ) evolution strategy method except that the evolution strategy is usually selected deterministically.

Stochastic universal sampling detailed in Baker (1987) can be used in all selection methods as an unbiased method to convert the remapped non-integer number into an integer number. Initially, all the individuals have been set a probability Pr(i) that they will be in the mating pool based according to either their fitness value or ranking in such a way that the probabilities sum to one. The circle outside the roulette wheel is marked by points at equal spacing of 1/ λ, so that number of points are equal to the number of offspring λ. The percent of the area of the roulette wheel taken up by parent i is then Pr(i). The roulette wheel is spun once and the individuals on the roulette wheel that are adjacent to the points on the circle outside the wheel are selected to go into the mating pool.

For generational algorithms, where the number of offspring is the same as the number in the population, λ is set equal to μ. The number of offspring that Stochastic Universal sampling is expected to assign to individual i is λPr(i). Stochastic Universal sampling has been shown to be optimally efficient, only needing to make a single pass over each individual member of the population in order to make assign an individual to each offspring. Stochastic Universal selection also has less variance than a roulette wheel sampling method where

repeated calls are required. When $\lambda < \mu$ this method also has the advantage that one can easily ensure that no individual is selected as parent more than once by setting $(1/\lambda)$ larger than the probability of the fittest individual.

The pseudocode for Stochastic Universal Sampling based on Grefenstette (2000a) is given below.

Input: a probability distribution (ProbDist) and the total number of children (λ)
Output: $\mathbf{c} = (c_1, c_2, \ldots c_N)$ where c_i is the number of offspring each individual in the population has, where $\Sigma c_i = \lambda$.

```
Function StochasticUniversalSampling (ProbDist, λ)
    Sample u ~ U(0, 1/λ)
    Sum = 0.0
    Repeat for i = 1 to μ
        c_i = 0
        sum = sum + ProbDist(i)
        Repeat while u < sum
            c_i = c_i + 1
            u = u + 1/λ
        End Repeat
    End Repeat
    Return c
End Function
```

Implicit fitness remapping

Ranking can be indirectly achieved by using tournament selection as described in Goldberg & Deb (1991). In random tournament selection a fixed number of solutions q are randomly chosen, either with or without replacement, from the population. Either the highest fitness solution is automatically selected as the parent (deterministic selection) or is given a pre-specified more than 50% chance of being selected (stochastic selection). This is repeated until the desired number of parents are selected.

In the case when q = 2, on average each individual will be chosen twice. On average, the best (worst) individual will win (loose) both tournaments, and the median individual will lose one and win one tournament. Therefore, on average this tournament selection method is equivalent to a linear ranking with a bias of 2 for the best individual. In order to have a bias between 1 and 2 the individual with the highest fitness value will be only be selected with probability $0.5 < p < 1$. In

order to achieve a bias that is larger than 2, one can increase the fixed number of solutions compared at each tournament to more than two. This will more strongly favour the higher fitness solutions leading to faster convergence.

The pseudocode for tournament selection with replacement is given below:

```
Input: Population of size P (Pop), tournament size q
Output: Selected Population SPop
Function TournamentSelection(Pop, q)
    Repeat for i = 1 to P
        SPop(i ) = best individual from q randomly chosen individuals from Pop
    End Repeat
    Return SPop
End Function
```

Due to the fact that each tournament is performed separately, tournament selection, like roulette wheel sampling, suffers from sampling errors. Goldberg, Korb & Deb (1989) reduce the sampling error by taking a copy of the population and choosing pairs for the tournament from each population without replacement. Therefore, this method ensures that each individual is in exactly two tournaments and that an individual cannot be in a tournament against itself.

An alternative tournament selection approach can be used where one by one every individual in the population is compared with a random subset of the population of size q. The individual is then assigned one point for each of the subset that it has a higher fitness value. Selection is then made for the solutions with the highest assigned scores. This alternative tournament selection method ensures that every member of the population is evaluated, resulting in the best solution always being maintained. A value of $q \geq 10$ is considered a relative hard form of selection with $3 \leq q \leq 5$ being a soft form of selection, see Back, Fogel & Michalewicz (1997).

Survivor selection

Evolutionary algorithms can either be generational replacement or steady state. In both approaches a constant population size of N is specified. In a

generational evolutionary algorithm the entire population is replaced by the offspring population every generation. This is an example of an age-based replacement method, that is not biased by fitness, where each individual exists for only one generation. In steady state evolutionary algorithms, λ ($< \mu$) old individuals are replaced by λ new offspring. The generation gap is the proportion of the population replaced in each generation, which has value of one for a generational method and value λ/μ for a steady state method. Survivor selection can also be fitness-based an example of which is the steady state GENITOR method were the worst λ members are selected for replacement. An elitist policy can also be used to ensure that the best solution is not lost.

Schwefel (1981) describe two types of evolution strategies that have both multiple parents and multiple offspring. In both methods λ offspring are produced from μ parents. Firstly, in the (μ, λ) method, where $\lambda \geq \mu$, the λ offspring are ranked according to fitness with the best μ being deterministically selected to replace the parent population. Typically λ is set much larger than μ and this is similar to biological species that have many offspring where only few survive to reproduce. Secondly, in the $(\mu + \lambda)$ method all the $\mu + \lambda$ individuals are ranked and only the best μ individuals are deterministically selected for the next generation.

Goldberg & Deb (1991) found that the high initial growth rate achievable with a steady-state method could also be achieved using large size tournament selection or exponential fitness ranking. Grefenstette (1986) concluded that larger generation gaps improved the performance of the algorithm. De Jong & Sarma (1993) found that there is higher genetic drift in a steady state system than in a generational method, especially when the population size is small and there is a low generation gap. Fogel & Fogel (1995) concluded that a generational approach may be a more suitable approach to solve practical optimization problems.

Comparison of selection methods

When choosing the selection method one needs to consider the trade-off between the speed of convergence and quality of the solution found. If selection too strongly favours the best solutions then the evolutionary algorithm may not have adequate time to explore the search space which results in premature convergence.

Proportionate selection has been shown to be inferior to tournament selection e.g. Chidambaran, Lee & Trigueros (2002). The reason for this is due to the problems proportionate selection has in distinguishing between the performance of solutions when the range between best and worst solution is small. Methods that discard the worse solutions (e.g. evolution strategies, threshold selection and tournament selection with a large number of tournaments) can result is a loss of diversity due to the genetic material contained in the least fit individuals being lost. Whitley (1989) showed that ranking is a superior selection method to fitness scaling. Blickle & Thiele (1996) provide an extensive comparison of several selection schemes and detail parameter settings that give equivalent selection pressures.

Goldberg and Deb (1991) found that both fitness ranking and tournament selection gave similar performances when suitable parameter settings were used. Proportionate selection was found to be an inferior method. They also prefer tournament selection over ranking due to its lower time complexity, as tournament selection does not require sorting of the population. Tournament selection may also be more appropriate for parallel evolutionary algorithms as the tournaments can be performed independently of each other. For these reasons I will be using tournament selection in my evolutionary algorithms.

An elitism policy can be used to force a number of the best solutions to be passed on to the next generation without being disrupted by crossover or mutation. Novkovic (1998) shows the benefits of the using an elitist selection operator, although the beneficial effects quickly diminishes as the number of elites passed on at each generation increases.

3.2.3. Variation Operators

Recombination

This is the predominant technique used by genetic algorithms to efficiently explore the search space. Genetic algorithms use a crossover rate, with a typical value of 60-70%, to determine the number of solutions in the population that should be produced from crossover rather than being passed on to the next generation unaltered. Two pairs of solutions are combined in the hope that the offspring will result in a better solution than the parents. Crossover is achieved by randomly choosing two parents. The relative weighting of each parent in the pair can be specified to be proportional to the relative quality of the parents. In a binary representation a cut can be made at the same random place in the string of each parent with the broken strings then recombining to form two new offspring.

In the continuous representation, an offspring $\bar{z}$ can be created from two parents $\bar{x}$ and $\bar{y}$ in a variety of ways. Each component of the offspring z_i, for $i \in \{1,...,n\}$, could be produced according to one of the following methods:

$$z_i = \begin{cases} x_i \text{ or } y_i & \textit{discrete recombination} \\ (x_i + y_i)/2 & \textit{arithmetic average recombination} \\ \sqrt{x_i y_i} & \textit{geometric average recombination} \\ a x_i + (1-a) y_i & \textit{weighted blending} \end{cases} \tag{3.8}$$

A weighted blending of two whole individuals could also be used where the two offspring produced are $\bar{z} = a\bar{x} + (1-a)\bar{y}$ and $\bar{z} = (1-a)\bar{x} + a\bar{y}$.

An alternative recombination method is an extension operator where the difference in the values of two parents is either added to the parent with the higher value or subtracted from the parent with the lower value.

Mutation

In a binary genetic algorithm mutation usually occurs with only a small probability to ensure that the benefits of selection and crossover are not lost. In a

binary representation a mutation could simply flip bits, from a 0 to a 1 and from a 1 to a 0, with a certain probability (typically 0.001).

There are a variety of methods that could be used in a continuous representation. One method is to use random replacement where a parameter value is randomly replaced. A more common approach called creep, given in Davis (1991), adds or subtracts a small randomly generated amount to a parameter value. Geometric creep can also be used where a parameter value is multiplied by a random amount that is close to one.

The random variable typically has a zero expectation and can be generated from a number of distributions. One can use a uniform random variable U(a,b) where the lower limit a is often set equal to –b. However, this method has the drawback of entrapment when the step size is too small to escape from local optima. Fogel (1994) showed that unbound mutation operators such as a zero-mean Gaussian random variable guarantee asymptotic global convergence. This is because although approximately two thirds of the draws will be within one standard deviation there is a nonzero probability that the random variable will cause a very large change. The standard deviation then controls the likelihood of a relative small or large move in the solution occurring. A Cauchy distribution with heavier tails can also be used.

Evolution strategies often use a self-adaptive mutation step size σ which is encoded into the solution and also undergoes variation and selection thus coevolving with the solution $\bar{x} = \langle x_1,...,x_n \rangle$ part. The approach allows the mutation step size to vary according to both its local fitness landscape topology as well as progress of the overall algorithm. Eiben & Smith (2003) detail the following specification where the mutation step size is the same for all the components x_i of the solution. The solution $\langle x_1,...,x_n,\sigma \rangle$ then mutates into $\langle x_1',...,x_n',\sigma' \rangle$ according to the following formulas

$$\sigma' = \sigma \exp[\tau N(0,1)]$$

$$x_i' = x_i + \sigma' N_i(0,1) \qquad (3.9)$$

In order to ensure that the mutation step size does not fall below a specified threshold, where its effect will become negligible, the following boundary rule is also enforced.

$$\sigma' < \varepsilon_0 \Rightarrow \sigma' = \varepsilon_0 \tag{3.10}$$

The variable τ is typically set inversely proportional to the square root of the problem size. Back (1996) explains that the reason for mutating σ by multiplying it with a lognormally distributed random variable is that small changes will occur more frequently than large changes and the mutation will be neutral on average. An additional complexity would be to use a different mutation step size for each component. The reason for using this approach is due to the observation that the fitness landscape has different slopes in each component direction.

Recombination vs. mutation

In genetic algorithms mutation is often seen as a background operator with crossover being the main method for exploring the search space. Schaffer & Eshelman (1991) found that crossover gives faster results than mutation-only method. However, the inclusion of mutation does generally result in a better solution than a crossover-only method. Schaffer, Caruna, Eshelman, & Das (1989) found that mutation is a very important operator whose optimum probability is more critical than that for crossover. Davis (1991) found that as the population converges, mutation becomes more effective and crossover less effective.

3.2.4. Methods for Ensuring Population Diversity

As the generation number increases, the diversity of the population will tend to reduce as it becomes more likely that there are collections of solutions which are close to each other in search space. Niching and speciation methods can be used to ensure diversity in a population in order to avoid premature convergence. These methods are particularly relevant for multimodal problems

where there are many local optima. When the representation uses real-numbers the distance measure will be based on an Euclidean measure, but can also be designed to take account of the specific nature of the problem.

1. Fitness sharing (or niching) – Fitness sharing was introduced by Goldberg & Richardson (1987). The reasoning behind this approach is the competition exclusion principle which says that competing species do not coexist in the same niche. Fitness sharing works by sharing the fitness of each solution with that of other solutions that are within the same region of search space. The following fitness sharing example is given in Michalewicz & Fogel (2004). A fitness $F_i^{'}$ is calculated for each individual i, where there are n solutions in the population, F(i) is the evaluation function in the case of goal maximization and d(i,j) is a distance measure from the i-th to jth solution

$$F_i^{'} = \frac{F(i)}{\sum_{i=1}^{n} sh(d(i,j))} \text{ where} \tag{3.11}$$

sh(d) is a sharing function (or niche count) given by

$$sh(d) = \begin{cases} 1-(d/\sigma_{share})^{\alpha} & d < \sigma_{share} \\ 0 & d \geq \sigma_{share} \end{cases} \tag{3.12}$$

σ_{share} defines the size of the neighbourhood around the i-th solutions and α (typically set equal to 1) is a scaling parameter. The closer two solutions are to each other the more they will be penalized for occupying the same region of the search space. This method results in the populations being spread out across multiple local optima with each peak supporting a number of individual solutions that is proportional to its height.

2. Crowding – De Jong (1975) introduced the crowding method where during selection, new solutions that are similar to old solutions will simply replace the old solutions. This is achieved by comparing each offspring with several randomly chosen population individuals. This reduces the likelihood that

the population consists of too many similar solutions. Deb & Goldberg (1989) showed that fitness sharing is a superior method to crowding.

3. Mating restrictions can be used to create separately evolving subpopulations. This is achieved by only allowing individuals who are within a specified distance measure (so can be classed as the same or similar species) to mate.

4. Competition amongst populations – The evolutionary algorithm is designed so that entire populations compete against each other. A population's quality is based on both the fitness of each individual solution and the overall degree of diversity within each population. The algorithm will use variation operators to generate new populations from existing populations, and selection is used to eliminate entire populations. The drawback of this approach is the computationally difficulty of maintaining populations of populations.

5. Explicit memory structures – Specific information obtained during the search process is stored in memory so that it can be re-introduced into the population at a later point. The method used by Mori, Imanishi, Kita & Nishikawa (1997) stores the best individual at each generation in memory. Individual solutions in memory may then be chosen as parents. Memory is limited so that individuals with a poorer fitness that do not adequately contribute to the diversity of the memory are removed from the memory storage.

The disadvantages of using an explicit method to ensure population diversity are difficulties in choosing the parameter to control the optimal level of diversity and the increase in the computational time needed to run the algorithm.

3.2.5. Multiple Population Models

The main advantages of simultaneous multiple population models are their implicit ability of helping to maintain population diversity, improved

algorithm performance at finding the global optima and the possibility of faster computation time due to parallelization.

Island models

An island migration model (also called coarse-grain parallel evolutionary algorithms) is where the population is split into several subpopulations and migration can occur between the different subpopulations. In island models the only form of communication between the islands is from migration of individuals. Island models are gene flow models which can benefit from reducing the chance of genetic drift, decreasing the chance of premature convergence and controlling the selective pressure. Premature convergence is reduced due to the diversity of the genes in the different islands and the periodic migration between the islands. Island models benefit from the opportunity that each subpopulation has to follow a different search trajectory through the search space. Martin, Lienig & Cohoon (1997) gives an overview of island models.

Wright's (1964) shifting balance theory says that species are more adaptive to their environment when they are divided into small local subpopulations (demes) which are sufficiently differentiated from each other. Eldredge & Gould (1972) punctuated equilibria theory says rapid evolutionary change can take place in a population when there is an influx of new individuals from peripheral isolated subpopulations of the same species. Martin, Lienig & Cohoon (1997) detail the two main ideas of the punctuated equilibria model: "an isolated subpopulation in a constant environment will stabilize over time with little motivation for further development" and "continued evolution can be obtained by introducing new individuals from other, also stagnating subpopulations". The punctuated equilibria theory heavily influenced the island model of Cohoon, Hedge, Martin & Richards (1987). In island models, exploitation arises from the evolution of subpopulations in isolation and exploration from the infusion of migrants from other subpopulations.

The island model consists of N subpopulation $\{P_1, P_2,..., P_N\}$ with an overall population P of M individuals. The size of population P_i is μ_i. When the subpopulations are evenly partitioned, the size of each subpopulation is given by $\mu = M/N$ individuals. The optimal structure of the P_i's is problem dependent.

The interconnection between subpopulations is referred to as communication topology. This comprises the degree of connectivity (i.e. number of links between subpopulations where migration can occur), magnitude of migration and frequency of migration between the different subpopulations. High communication between the subpopulations results in an island model where the isolated evolution time for each subpopulation is small. This is similar to a single large intermixing population which may suffer from premature convergence due to reaching stable gene frequencies too quickly. Low communication between subpopulations results in an island model where the isolated evolution time for each subpopulation is high, which is similar to a model with several small independent populations. Each subpopulation will tend to concentrate on finding optima that is relatively close to its initial population due to selective pressure and is more likely to suffer from genetic drift. Therefore, the best balance between exploitation and exploration is likely to be achieved for intermediate degrees of communication.

Periods of isolated evolution are called an epoch. Migrations between subpopulations occur at the end of each epoch, except the last. The length of the epoch will determine the frequency of the migration between subpopulations and is specified by the number of generations (G_i) in which subpopulation i evolves in isolation.

The island model comprises E epoch iterations. During an epoch each subpopulation will independently execute a sequential evolutionary algorithm for G generations. At the end of each epoch a communication phase is carried out where individuals can migrate between neighbouring subpopulations.

The Pseudocode for an island model based on Martin, Lienig & Cohoon (1997) is given below.

Input: number of epochs (E), number of subpopulations (N), number of individuals in each subpopulation (μ)
Output: Best individual found
Function IslandModel(E, N, μ)
    Initialize Concurrently (P_i, μ) for each i = 1 to N subpopulations
    Repeat for epoch = 1 to E
        Repeat Concurrently for each i = 1 to N subpopulations
            SequentialEvolutionaryAlgorithm(P_i,G_i)
        End Repeat
        Repeat for i =1 to N
            For each neighbour j of i **Do** Migration(P_i, P_j) **End For**
            Assimilate(P_i)
        End Repeat
    End Repeat
    Return best individual of all subpopulations
End Function

There are two common ways migration of individuals between subpopulations can occur. Firstly, an individual could migrate from P_i to P_j where the individual is deleted from P_i and added to P_j. If the migration rates are symmetric across all subpopulations the size of each subpopulation would then remain constant after migration and assimilation has occurred. Secondly, a copy of an individual from P_i could be added to P_j, which was the approach used by Cohoon, Hedge, Martin & Richards (1991). In order to maintain the constant population size in P_j an individual from P_j would need to be deleted during the assimilation process.

In both the shifting balance and punctuated equilibria theories the subpopulations evolve in distinct environments. Normalization of fitness values within each subpopulation will cause distinct environments in the subpopulations to the extent that subpopulations have individuals with differing fitness value distributions. A more direct way of ensuring distinct environments is to have a different objective function for each subpopulation. This approach is less usual as it is typical to have only one objective function that is common to all subpopulations.

Martin, Lienig & Cohoon (1997) carried out several experiments to find the optimal routing in circuit design using a parallel island model and sequential single population model. They then investigated what were the optimal parameters for the island model. In order to fairly compare the performance of the parallel island model with a sequential single population model, both the number of offspring produced at each generation and the total number of generations were identical for both models. The sequential model had a total population size of 450 and the island model had 9 evenly partitioned subpopulations each of size 50. Each subpopulation in the island model had exactly four neighbours. Their results are detailed below:

- Number and size of the subpopulations – when 9 subpopulations were used the island model consistently outperformed the sequential model. Increasing the number of subpopulations above 9 while keeping the total population size unchanged, so that μ decreased, resulted in progressively worse results for the island model which eventually underperformed the sequential model. The reason given for this was that in order for the island model to be effective, μ needs to be above some critical mass, the level of which is dependent on the complexity of the problem.
- Number of migrants and epoch length – The best results were achieved when 2 migrants were sent to each neighbour and the epoch length was 25 generations. A higher number of migrants or a shorter epoch length reduced the diversity between the subpopulations and resulted in the subpopulations searching the same part of the search space (i.e. too close to a single population model). They found that migration of more than 25% of the subpopulation at one time is disruptive. A lower number of migrants or a longer epoch length meant that good genes were not adequately spread out amongst the whole population (i.e. too close to a model with independent subpopulations). A strictly isolated island model, with no migration allowed, still outperformed the sequential model. Therefore, on the assumption that the subpopulations had critical mass, dividing the total population into

independent subpopulations increases the chance that at least one of the subpopulations evolves into a better solution. An approximation of the optimal fixed epoch length will be the time it takes for individual subpopulations to converge.

- Migrant selection strategy – three migrant strategies of random, top 50% (above medium) individuals and the best individuals were investigated. The migrants were sent in a random order to the four neighbours, with each migrant only being allowed to be sent once. They found that selecting better or the best individuals led to poorer results and increases in the occurrence of premature convergence due to the subpopulations searching the same part of the search space concurrently.

Overall they found that the island model consistently outperformed the sequential model when both models searched an equivalent number of offspring. The additional computer time required to perform migration is also extremely small relative to the large amount of computer time needed for the evolution of each subpopulation. A key advantage of the island model is the fast computation time possible while using parallel processing.

Whitley, Rana & Heckendorn (1999) analyzed the effect of number of subpopulations and their sizes on the algorithm performance. Their findings suggested that island models can outperform a single population model for optimization problems where an increase in the population size does not result in fitter solutions.

Migration policies were studied by Cantu-Paz (2001) who found that it considerably affects the selection intensity and speed of convergence. They investigated the four migration policies of either random or fitness based approach for selecting both which individuals should migrate and which individuals they should replace. Using a fitness based approach to both select individuals that are to migrate and individuals that are to be replaced results in the

highest selection pressure and also corresponds to the most common approach used.

Skolicki & De Jong (2005) investigated the influence of migration size and interval length on the performance of island models. The best performance was achieved using moderate migration intervals and small migration sizes. They also found that the migration interval was a more dominant factor in the performance of the algorithm than the migration size. Their results were indicative of having a migration size of 5-10% every 10 generations.

Below I describe several specific types of island models that have been used. A stepping stone model is an island model where migration can only occur between neighbouring subpopulations. An example is given by Eby, Averill, Goodman & Punch's (1999) injection island model which has a different level of resolution of the problem representation, and correspondingly different sub-fitness functions, in each island. Different genetic algorithm parameters could also be used for each population. Approximate building block solutions were first found on the island with the simplest representation and highest exploration speed. Good solutions would then migrate to the island with a higher level of resolution and a greater exploration time in order to fine-tune the approximate solutions. This process then continues over several levels of higher resolution islands. The searching at various refinement levels was found to result in an efficient and robust algorithm. Skolicki & De Jong (2004) use an island model that has a different representation in each island, so that the solutions are either represented using binary or Gray code. During migration between the islands, the individuals are transformed from one representation to the other. This algorithm was found to perform at least as well or outperform the single representation model for several multi-model test functions. The explanation for this outperformance was that a local optima in one representation would not necessarily be a local optima in another representation, thus reducing the likelihood of becoming trapped in a local optima.

Diffusion models

Diffusion models (also called fine-grained parallel evolutionary algorithms) have overlapping fixed sized population and are described in Pettev (1997). Gorges-Schleuter (1989) identified that in evolution processes individual genes spread through the whole population in a similar way to a diffusion model. Individuals evolve in parallel, with selection and recombination typically performed within a local neighbourhood, see Collins & Jefferson (1991). In order to keep the time spent in communication to a minimum the size of the neighbourhood is usually kept small.

Parents can be chosen by performing standard selection techniques on a local population such as ranking or tournament selection. Other selection techniques that have been used include choosing an individual as the best individual in the neighbourhood, i.e. Farrell, Kieronska & Schulze (1994), or choosing each parent as the best individual found in a random walk, i.e. Collins & Jefferson (1991). De Jong & Sarma (1995) found that binary tournament selection performed worse than linear ranking even though the two methods have equivalent selection pressure. They conclude that 'these results emphasize the importance of an analysis of the variance of selection schemes. Without it one can fall into the trap of assuming that selection algorithms that have equivalent expected selection pressure produce similar search behaviour.

Diffusion models typically use standard recombination techniques to produce one or two offspring. When one offspring is produced it typically replaces the individual. In the model of Gorges-Schleuter (1989), the single offspring produced only replaces the individual if the offspring fitness is not worse than the least fit individual in the neighbourhood. When two offspring are produced typically the best individual will replace the individual, e.g. Manderick & Spiessens (1989).

In diffusion models the shape and size of the neighbourhood will be influenced by the underlying design architecture, the most common being a grid (or mesh). The neighbourhood is then normally defined in terms of the distance

between different points on the grid. For example, in a two-dimensional grid, with an individuals neighbours being individuals one link away, the neighbourhood would consist of 9 individuals (i.e. a 3 by 3 square on the grid). The size and length of the neighbourhood may also be random as in the random walk selection method used in Collins & Jefferson (1991).

3.2.6. Parameter Tuning and Parameter Control

The setting of the parameter values of the algorithm can be divided into parameter tuning and parameter control. Parameter tuning is the process of finding the optimal parameters for the algorithm before the run, with the parameters then fixed during the run. Parameter control is the process of using dynamic rather than static parameter values whose values may change during the run. The key components of an evolutionary algorithm are: representation of the individuals, objective function, population characteristics (e.g. size and topology), parent selection operators, variation operators and replacement operators.

The methods for setting the parameter values can be divided into the following three categories:

1. Deterministic – The parameter value is altered in a deterministic way, not depending on any feedback from the search process. Parameters could be either static or dynamic (e.g. using a time-varying parameter that depends on the generation number).
2. Adaptive - The current value of parameters is dependent on feedback from the search process. Examples of statistics that can be used to affect the change are the performance of operators and the diversity of the population.
3. Self-adaptive – Parameters are themselves encoded into the data structure of individual solutions. The solutions with the best choices of parameter values are more likely to survive and reproduce until they dominate the population of solutions.

Parameter tuning

Tuning the parameters in an evolutionary algorithm by experimentation has a number of drawbacks. It is a time-consuming process that has no guarantee of finding the optimum parameter values. It is normally only practical to tune one parameter at a time, even though the parameters are not independent. The performance of the genetic algorithm may also critically depend on the representation and fitness function used. However, Grefenstette (1986) concluded that for a well chosen representation and fitness function genetic algorithms are robust for a fairly wide range of algorithm parameters, such as crossover probabilities, mutation probabilities and population size.

Parameter control

Different values of the parameters may be optimal at different points in the evolutionary process. Therefore, static parameter can lead to a sub-optimal solution compared to when dynamic parameters are used. A simple dynamic parameter may only be a function of time (i.e. generation number), whereas a more advanced adaptive operator may also be a function of specific characteristic of the current population. The drawback of dynamic operators is that it is harder to determine the parameter function that should be used due to the increased range of possibilities.

As the optimal operator probability may vary as the generation number increases one may use dynamic operator probabilities. Fogarty (1989) decreases the mutation rate over successive generations, in a comparable way to which temperature is reduced in simulated annealing, in order to help convergence. Booker (1987) uses an adaptive variable crossover rate that depends on the variability of the fitness values. As the population converges, the crossover rate is reduced and more emphasis is placed on the mutation operator. Davis (1989) used adaptive operator probabilities that were directly based on the performance of the operators over the previous 50 matings at producing good offspring. When

using this technique one needs to be aware of the possibility that operators may be rewarded for helping to locate local rather than global optima. Lee & Takagi (1993) showed that using fuzzy logic techniques to dynamically control the parameter settings of genetic algorithms resulted in an improved performance over a simple static genetic algorithm.

Rechenberg's (1973) 1/5 success rule states that the ratio of successful mutations, where the offspring is fitter than the parent, to all mutations should be 1/5. If the ratio is greater than 1/5 the mutation step size needs to be increased to allow a wider exploration of the search space. If the ratio is less than 1/5 the mutation step size should be reduced to more closely focus on the area around the current solution. The mutation step size parameter σ should be adaptively reset at periodic intervals according to

$$\sigma = \begin{cases} \sigma / c \ \ if \ \ p_s > 1/5 \\ \sigma . c \ \ if \ \ p_s < 1/5 \\ \sigma \ \ if \ \ p_s = 1/5 \end{cases} \tag{3.13}$$

where p_s is the frequency of successful mutations measured over a specified number of trials. Schwefel (1977) uses a parameter value c that is in the range $0.817 \leq c \leq 1$.

3.2.7. Advantages of Evolutionary Algorithms

A key advantage of an evolutionary algorithm is the ability to incorporate the best features of many methods as well as making use of problem specific information. Techniques from population heuristics, single solution heuristics, neural networks and fuzzy logic can all be incorporated. For example, one can use an evolutionary algorithm to find the optimum solution and then use this as the starting point for a local improvement search. Simulating annealing can be thought of a particular case of an evolutionary algorithm where the population size is only a single individual and the selection mechanism is based on the temperature parameter. Similarly, the idea behind a tabu search can be

incorporated into an evolutionary algorithm by adding a memory structure detailing past solutions. Although evolutionary algorithms get inspiration from biological laws, they are not limited to proven biological laws e.g. can use Lamarckian evolution (covered in section 7.1) which has been rejected as a theory of genetic evolution or allow recombination between more than two individuals.

Initially, the algorithm parameters can be based on research carried out to identify standard parameter values that should be appropriate for a large range of problems. In addition to using an evolutionary approach to find the optimal solution, an evolutionary approach can also be used to find the best algorithm parameters at any stage of the run.

An evolutionary algorithm allows one to find multiple solutions, which is often useful when there are multiple objectives that cannot be easily aggregated with a normalizing function. One can allow for diversity of the population in order to avoid multiple solutions within a too close neighbourhood region. Evolutionary algorithms can be used to solve several problems simultaneously, while benefiting from the interaction of individual solutions between the different subpopulations.

Evolutionary algorithms are able to adapt to changes made to either the model or evaluation function. Although such an unexpected change will cause a short-time disturbance in the evolutionary process, the population will quickly adjust to the change. This is an advantage over algorithms that would need to be restarted each time an unexpected change occurs.

3.3. Incorporating Constraints

One of the key advantages of using a heuristic approach to solve the portfolio optimization problem is the high degree of flexibility possible when specifying the model constraints. This is due to the variety of methods that can be

effectively used to handle constraints when using a heuristic optimization approach.

A search space S consists of two disjoint subsets of feasible and infeasible subspaces given by F and U respectively. A simple way to allow for constraints is to use the death penalty method where any infeasible solution is removed. However, this approach will only work for a limited number of problems, in particular when the search space is convex. Michalewicz (1995) showed that this method was poor and less robust than the alternative techniques.

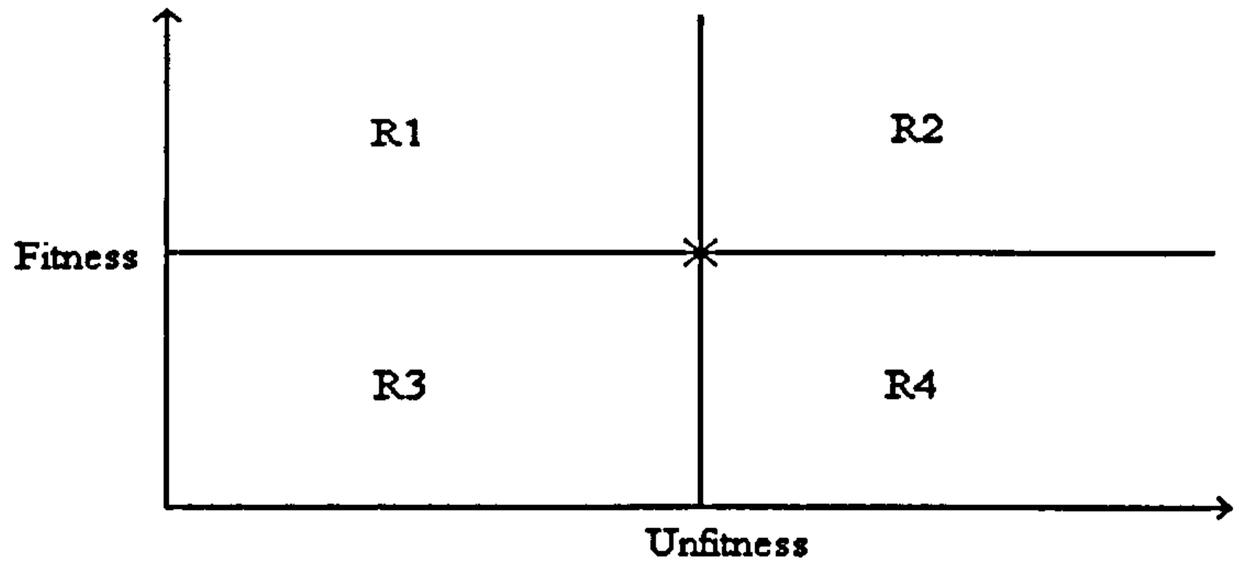
Beasley (2002) identifies four methods, given in a preference order, which can be used for incorporating constraints into a population heuristic optimization. The purpose of these methods is to ensure that individual population members always represent a feasible solution.

1. *Reducing the allowable search space.* Constraints can be imposed by reducing the allowable search space to feasible solutions thereby ensuring illegal solutions never occur. Michalewicz & Janikow (1991) also show that this is the preferable approach to use to deal with infeasibility where it is possible.
2. *Repairing infeasible individuals.* One may be able to design a heuristic repair operator that ensures that any infeasible solution can quickly be transformed into a feasible solution. Repairing infeasible solutions through a process of learning and evolution through a single generation is an example of the Baldwin effect (covered in section 7.1). The weakness of this method is that it is problem dependent as each problem will require a different heuristic repair operator.
3. *Separating individuals and constraints.* One can incorporate constraints by separately calculating the evaluation of fitness and infeasibility. Both an unfitness value (measuring the degree to which the constraints are violated) and a fitness value (measured by the objective function) are calculated for each population member. An unfitness value larger than zero means that the

population member is infeasible. Furthermore, the higher the unfitness value is, the more infeasible the solution will be. The optimal solution will be the maximum fitness value possible for solutions that are feasible (i.e. those where unfitness ≤ 0).

Population members can be ranked by comparing their unfitness and fitness values. Figure 3.2 details how this can be done by showing an example solution that is marked with a cross. R1 – R4 represent the four regions of the unfitness-fitness space. Solutions in R1 are always more preferable than the example solution as they have both a higher fitness value and a lower unfitness value. Similarly, solutions in R4 are always less preferable than the example solution. It is less clear whether we should prefer solutions in R2 or R4. A detailed discussion of this can be found in Chu & Beasley (1998).

Figure 3.2: Constraint handling by separating individuals and constraints



4. *Penalizing infeasible individuals.* If the above three approaches are not possible constraints can be included through the use of a penalty function on the objective function as described below. To apply a penalty function approach requires both a definition of unfitness, which is dependent on the severity of the constraint violation, and the setting of a penalty coefficient σ. The penalized fitness value is then of the form:

Penalized fitness value = original fitness value – σ(unfitness). (3.14)

The reason the penalty function method is inferior to method 3 is due to the fact that it requires more decisions to be taken due to the need to both define unfitness and set a penalty coefficient.

Below I detail how the penalty function approach can be used to solve a constrained optimization problem of the following form.

$$\begin{aligned} &\max f(\mathbf{x}) \\ &\text{s.t. } h_i(\mathbf{x}) = 0 \; i \in E \\ &\quad g_i(x) \leq 0 \;\; i \in I \end{aligned} \tag{3.15}$$

By using a penalty function, this can be approximated by the following constrained optimization problem.

$$\max \Phi(x,\sigma_1,\sigma_2) = f(x) - \sigma_1 \sum_{i \in E} h_i^2(x) - \sigma_2 \sum_{i \in I} \left[\max(g_i(x),0)\right]^2 . \tag{3.16}$$

If the penalty coefficients σ_1 and σ_2 are set too high the optimization algorithm will first concentrate on minimizing the penalty function and only thereafter focus on maximizing the objective function f. In order to avoid the convergence difficulties that results when using high penalty coefficients one can solve a sequence of unconstrained optimization problems by using the best solution of each subproblem as the starting solution for solving the next optimization problem.

When setting the level of the penalty coefficient one should consider Le Riche, Knopf-Lenoir & Haftka (1995) 'minimal penalty rule'. This says that the penalty coefficient should be set as low as possible, so that it is just above the threshold below which infeasible solutions would be optimal. A too low penalty coefficient will lead to infeasible solutions. A too high a penalty coefficient restricts the search to the feasible region, reducing the opportunities to take short cuts across the infeasible region and often leading to premature stagnation at a non-globally optimal feasible solution.

Richardson, Palmer, Liepins, & Hilliard (1989) detail how penalty functions should be constructed. They found that penalty functions which are based on the amount constraints are violated perform better than those just based on whether or not the constraint has been violated. They say that penalty functions should be constructed from the expected completion costs, which is the cost of turning an infeasible solution to a feasible solution.

Michalewicz & Fogel (2004) conclude that the choice of the penalty function may depend on

- The ratio between the size of the feasible search space to the whole search space as well as the topological properties of the feasible search space
- The type of the evaluation function and the number of variables and number of local optima
- The number of constraints, types of constraints (linear or non-linear, equality or inequality), and number of active constraints at the optimum.

This means that choosing a suitable penalty function is not a trivial problem and highlights the advantage of using a self-adaptive penalty function.

Joines & Houck (1994) gave an example of a dynamic penalty function that has a component $(C \times t)^{\alpha}$, where C and α are constants and reasonable parameters values are given as 0.5 and 2 respectively. As the generation number t increases, the penalty on infeasible solutions increases. They found that the factor $(C \times t)^{\alpha}$ may grow too quickly causing infeasible solutions to be over-penalized in later generations. This meant that the best solutions were frequently found in earlier generations.

Bean & Hadj-Alouane (1992) described how one can use an adaptive penalty function with component $\lambda(t)$, the value of which depends on the proportion of infeasible individuals as best solutions in previous generations. At each generation the value of λ would be increased (decreased) by a constant percentage amount if all the best solutions in the last k generations

were infeasible (feasible). The value of λ would be left unchanged if there were both infeasible and feasible best solutions in the last k generations.

3.4. Designing and Reporting on Heuristic Methods

Barr, Golden, Kelly, Resende & Stewart (1995) describe that contributions to a new heuristic method may include the following

- Gives a higher quality solution than existing methods
- Gives a high quality solution faster than existing methods
- Is simpler to implement or can be applied to a broader range of applications
- Is less sensitive to differences in the nature of the problem or the tuning of the algorithm parameters than existing methods.

They also listed the following steps which should be taken when designing and reporting on computational experiments with heuristic methods.

- Define the goals of the experiment – e.g. to compare the performance of different algorithms for a particular type of problem, to investigate the performance of a particular algorithm.
- Select performance measures – as evolutionary algorithms are stochastic, performance measures will be statistical in nature. The performance measure should consider factors such as the quality of the best solution found, time to reach the best solution, time to reach an acceptable solution, robustness of the method. Common performance measures are given below.
  - Success rate measure – is the percentage of runs that achieve a successful solution. As the optimal solution to my specific problem is unknown this performance measure is of limited use.
  - Mean best fitness measure. For each run the best fitness individual is identified. The mean best fitness is then the average of these best fitness values over all the runs.
  - Best-ever fitness and worst-ever fitness over all the runs. This gives information on the variability of the quality of the solutions the algorithm

produces and can be used to give statistical confidence levels on the performance.

In order to effectively compare the performance of different evolutionary algorithms, the above performance measures need to be calculated given a fixed pre-determined amount of computational time. The proxy for computational time is typically taken as the average number of solutions searched or evaluated over a number of runs. The pace of the progress of an algorithm can be determined by a progress plot of the generation number against the performance measure. The performance of the algorithm as the problem size increases can be determined by a scalability plot of the problem size against termination performance measure.

- Select factors that will be investigated – e.g. the influence of the size of the problem on the computational effort and the quality of the solution.
- Design and carry out the experiment
- Analyze the results and draw conclusions

Heuristic research is said to be beneficial if it is

- Revealing – giving an insight into reasons for the particular performance and characteristic of an algorithm and how best to use this information in future algorithm design.
- Gives theoretical insights - e.g. bounds on the quality of a solution.

3.5. Conclusion

In this chapter I have explained my reasons for using evolutionary algorithms to solve the multistage portfolio optimization problem. This is because it benefits from being a less problem dependent heuristic method which is also able to solve complex high dimensional problems. It is more likely to reach a global optimum as it considers many potential solutions simultaneously and has the unique crossover operator that makes it possible to merge the best attributes of current solutions. One is also able to measure the level of population

convergence towards the optimal solution. When choosing the algorithm design I have shown that the use of multiple population models can improve the efficiency of the evolutionary algorithm.

Chapter 4

Implementation of the Multistage Portfolio Optimization Model

The multistage portfolio optimization problem is a multiobjective problem due to the two objectives of maximizing expected return and minimizing the specified risk measure. Therefore, in order to optimize my problem, the evolutionary algorithm will need to find the portfolio with the maximum expected return for a given level of risk, and thus find the efficient frontier. The efficient frontier can then be used to assess the tradeoff between expected return and risk. Evolutionary algorithms provide a very natural way to perform multiobjective optimization and in this chapter I will describe alternative designs of multiobjective evolutionary algorithms.

To put this chapter in context I begin by describing the overall method of how evolutionary algorithms can be used to solve a multistage portfolio optimization problem. The aim of the evolutionary algorithm is to find the asset allocation weights at each time period t = 0, 1, 2,...T-1, for a dispersed number of points on the efficient frontier, where T is the number of time periods in the multistage problem. The method for solving the multistage scenario optimization problem is specified below.

(i) *Scenario generation.* A large number of simulations are generated in order to produce a representative scenario tree. For each point on the scenario tree the value of individual asset prices is calculated and these are used to value the equity options.

(ii) *Find the optimal asset allocations using evolutionary algorithms.* Given the scenario tree the multistage optimization problem now becomes a deterministic optimization problem which can be solved using evolutionary algorithms. Each individual solution in the population will consist of the genes that can be used to calculate the asset allocation weights at each possible rebalancing time (i.e. w_{it} for i = 1, 2 ,..., N, t = 0, 1, 2,..., T-1). These genes are chosen and modified in a way that the constraints of the model are met. I have taken this approach to constraint handling as limiting the search space to permitted asset weight combinations has been found to be the most efficient constraint handling method (as detailed in section 3.3). A multiobjective evolutionary algorithm will then be run until adequate solution convergence is reached in order to find the efficient frontier. Each point on the efficient frontier will detail for a certain level of risk aversion, the optimal asset allocation weight for each security at each rebalancing date.

This chapter begins by looking at possible specifications of the objective function including detail on alternative risk measures that can be used. I look at the three main approaches that can be used for multiobjective optimization, namely, multiple single-objective searches, single-objective simultaneous parallel searches and the Pareto dominance approach. I then go on to detail the multistage portfolio optimization model and the methods used for scenario generation. Finally, I explain the common features of my optimization problem and evolutionary algorithms.

4.1. The Objective Function

The two basic approaches that can be used when specifying the objective function are to either maximize expected utility or to maximize expected return less a specified risk measure. Utility functions have the advantage over the mean-

variance approach as they allow the arbitrary choice of the shape of the asset distribution and can allow for asymmetric portfolio performance. However, due to the common difficulties investors have in knowing how to specify their desired utility function I am going to use the flexible risk measure based approach.

The simple form of the objective function is to maximize expected wealth less a measure of risk as shown below

$$Max\left[\alpha \cdot \mathrm{E}(W_T) - (1-\alpha) \cdot RiskMeasure(W_T)\right] \tag{4.1}$$

where $\alpha \in [0,1]$ is a risk aversion parameter determining the level of risk aversion an investor has. The expectation and value of the risk measure are calculated across all of the simulated scenarios. $\alpha = 1$ results in the portfolio with the highest expected return, which when no asset allocation constraints are in place will consist entirely of the highest expected return asset. $\alpha = 0$ results in the portfolio with the lowest level of risk which will typically consist of several assets that form a diversified portfolio. By maximizing the objective function for different values of α one can build an efficient frontier. This was the technique used by Chang, Meade, Beasley & Sharaiha (2000).

Traditionally, the optimal portfolio is constructed using an optimization technique where portfolio weights are chosen by trading off expected risk against expected return. This optimal portfolio is then checked for risk budgeting constraints so that specific or aggregate risk exposures are within acceptable limits. However, when carrying out portfolio risk budgeting it is still important to allow for the optimal trade-off between expected risk and expected return. Therefore, a preferred approach would be a specification of the objective function that allows portfolio construction and risk budgeting to occur simultaneously, thereby ensuring the most optimal solution in terms of risk and return is reached.

4.1.1. Alternative Risk Measures

Below I give details of the alternatives scenario based risk measures that can be used.

Mean-Variance model

This is the most well known risk measure which originated in Markowitz's (1952) mean-variance framework and is given by

$$\sigma^2 = \frac{1}{m}\sum_{s=1}^{m}(r_s - \mu)^2, \tag{4.2}$$

where r_s is the portfolio return in scenario s, and m is the number of scenarios. Variance is an example of a symmetric risk measure as it can not distinguish between returns that are equally distanced above or below the mean. Symmetric risk measures will be less suitable in situations where returns deviate from normality or investors are highly risk-averse.

Mean Absolute Deviation

Mean Absolute Deviation (MAD) is given as the probability-weighted sum of the absolute deviations from the mean as given below,

$$MAD = \frac{1}{m}\sum_{s=1}^{m}|r_s - \mu|. \tag{4.3}$$

This model has been used by Konno & Yamazaki (1991) and Feinstein & Thapa (1993). Konno (1990) showed that when the returns are normally distributed the MAD model could be solved using linear programming. The implication of the MAD model is that an additionally unit of underperformance relative to the mean results in the same amount of disutility irrespective of the current size of the loss. One can also modify this risk measure so that a higher cost is placed on deviation above rather than below the mean.

Generalized Semi-variance

This model tries to address the limitation that variance has as a risk measure due to its symmetry property. Semi-variance is calculated based on only those scenarios whose returns are below the mean, in the case of lower semi-variance (sv^-) or above the mean, in the case of upper semi-variance (sv^+).

$$sv^{-} = \frac{1}{m}\sum_{s=1}^{m}(r_s - \mu)^2 \delta_s$$

$$sv^{+} = \frac{1}{m}\sum_{s=1}^{m}(r_s - \mu)^2 (1 - \delta_s)$$

$$\delta_s = \begin{cases} 0 & r_s > \mu \\ 1 & r_s \le \mu \end{cases}. \qquad (4.4)$$

The most common form of the semi-variance model uses the lower semi-variance measure. A more flexible approach was given by Hamza & Janssen (1996) who used a flexible generalized semi-variance model given as the weighted linear combination of the upper and lower semi-variances,

$$sv^{weighted} = wsv^{-} + (1-w)sv^{+} \text{ , where } w \in [0,1]. \qquad (4.5)$$

Probability-Based Risk Measures

Probability–based risk measures are based on the probability that a particular outcome occurs and is closely related to Value-at-Risk (VaR). Its popularity is due to the ease at which investors are able to understand probability-based measures, regulatory considerations when solvency is measured by a liability threshold and the goal of fund managers to outperform their given investment targets. Uncertain investment returns can be decomposed into a threshold return (γ) plus an upside measure, given by max[r-γ, 0] minus a downside risk measure, given by max[γ-r, 0]. This results in the following formula:

$$r = \gamma + \max[r - \gamma, 0] - \max[\gamma - r, 0]. \qquad (4.6)$$

Risk measures that focus on the downside (upside) of a return distribution are called lower (upper) partial moments and were introduced by Harlow (1991). Lower partial moments were also used by Feiring, Wong, Poon & Chan (1994). The lower partial moments gives the moments of the return distribution below the specified threshold γ and is given below.

$$lpm_{\gamma}^{k} = \frac{1}{m}\sum_{s=1}^{m}(r_s - \gamma)^k \delta_s$$

$$\delta_s = \begin{cases} 0 & r_s > \gamma \\ 1 & r_s \leq \gamma \end{cases} \tag{4.7}$$

Setting k to different values will yield the following results. k=0, the shortfall probability which is a popular with asset managers, k=1, the average shortfall, k=2, the shortfall variance, k=3, the shortfall skewness and k=4 the shortfall kurtosis.

The level at which the threshold return γ is set will depend on the specific objectives of an individual investor, e.g. zero for nominal capital protection, the inflation rate for real capital protection, the risk-free interest rate to meet a minimum target opportunity cost, the actuarial rate to meet actuarial funding requirements or a moving target rate to meet a time-varying target opportunity cost. The use of a low threshold will increase the size of the estimation error as the observations below the threshold will occur less frequently.

Differences between the optimal portfolio allocations when using variance compared to the lower partial moment as the risk measure will increase with the level of skewness, decrease with the level of return required and decrease with the level of the chosen threshold return.

The shortfall probability is the probability that the loss will fall below a specified threshold γ whereas VaR is the maximum loss that will occur with a specified significance level α. Therefore, both measures will coincide if we set $\gamma = VaR$.

Minimum Regret

An example of a minimum regret risk measure is given in Young (1998) who uses a minimax criteria of minimizing the maximum portfolio loss. This measure may be suitable for investors who have to ensure that they never incur a specified size loss. A disadvantage of the minimum regret measure is that the resultant portfolios are often too concentrated due to the exclusion of highly non-normal assets.

Value-at-Risk

The safety-first approach of Roy (1952) used a shortfall constraint defined as the probability that the end of horizon investment value falls below a specified disaster level is smaller than a specified disaster probability. A similar shortfall constraint approach was also used by Leibowitz & Kogelman (1991). This approach is generalized by the Value-at-Risk shortfall measure which says that the maximum expected loss would not exceed the VaR over a given investment horizon at a given confidence interval.

Conditional Value-at-Risk

The $CVaR_\alpha$ is defined by the expected loss conditional on being in the α worst cases. If the portfolio returns from all the scenarios are ordered so that $r_{1:m} \leq r_{2:m} \leq \ldots \leq r_{m:m}$ then the $CVaR_\alpha$, also called the expected shortfall (ES), is given by:

$$CVaR = \frac{1}{m\alpha} \sum_{s=1}^{m\alpha} r_{s:m} \tag{4.8}$$

Rockafellar & Uryasev (2000) use CVaR as a risk measure and Acerbi & Tasche (2002) discuss the use of expected shortfall as a risk measure and when it is appropriate. A disadvantage of CVaR is the high estimation error due to high outlier-dependence.

Risk measures that depend on wealth trajectory paths

The risk measures considered above only take into account the distribution of the portfolio returns over the whole investment period. However, one can structure the objective function so that it not only depends on the distribution of the level of wealth at time T over all the simulated scenarios, but also on the individual trajectories of the wealth processes over the entire period [0, T]. This may be desired because even if two scenarios result in the same level of wealth at time T investors are likely to prefer a scenario with

- low volatility in wealth levels over the period [0, T]

- a high value of the function $\min_{[0,T]} W(t)$
- w(t) not being below some critical level for long periods
- a high value of the average value of W(t) over the period [0,T].

Therefore, one could add an additional function to the basic form of the risk measure to take into account preferences for the actual wealth trajectory path.

An investigation into the volatility in the wealth levels over the time period [0, T] caused by different asset allocations may be of particular interest to a liability matching investor. Liability benchmarking would typically result in a higher concentration of assets which more closely track the liabilities in order that the surplus will be less volatile. The success of a particular portfolio allocation in tracking a liability benchmark can be assessed by calculating the tracking error, which is described below.

Active return is defined as the portfolio return (r_t) minus the benchmark return (R_t).

Tracking error which was introduced by Sharpe (1981), was defined as the variance of the active return. Beasley, Meade & Chang (2003) identified that a problem with this definition was that a portfolio that constantly underperformed the index by a fixed amount has a tracking error of zero. Therefore, they use a tracking error defined as

$$TrackingError = \frac{1}{T}\left[\left[\sum_{t\in S}(r_t - R_t)^{\alpha}\right]^{1/\alpha}\right] \text{ where } \alpha > 0. \qquad (4.9)$$

The parameter α sets the degree to which differences in the portfolio return and benchmark return are penalized. The set S could either comprise all the time periods, or only the time periods for which the portfolio return is less than the index return, resulting in a downside risk measure.

Tracking error is the predominant risk measure for passive investment, but may also be appropriate to use in other situations e.g. liability benchmarking, where the active return is set as the level of outperformance of assets over the liabilities. When incorporating tracking error as part of the objective function one

has to ensure that penalization for high tracking error figures are not detrimental to overall portfolio efficiency.

4.1.2. Coherent Risk Measures

Artzner, Delbaen, Eber & Heath (1997, 1999) describe the desirable properties that are needed for a statistic of portfolio returns to be considered a coherent risk measure.

1. *Monotonicous.* Larger losses should translate into an increase in risk.
2. *Positive homogeneous.* A linear multiplication of a holding size should cause risk to increase by the same linear factor.
3. *Invariant to translations.* Adding a constant profit or loss to our portfolio should not change the level of portfolio risk.
4. *Subadditive.* The risk of a portfolio should be at most the combined risks of the individual portfolio positions. This means that the level of risk in a portfolio that is made up from the combination of two individual portfolios must not be higher than either of the individual portfolio risks, i.e. ensures that portfolio diversification does not result in an increased risk level.

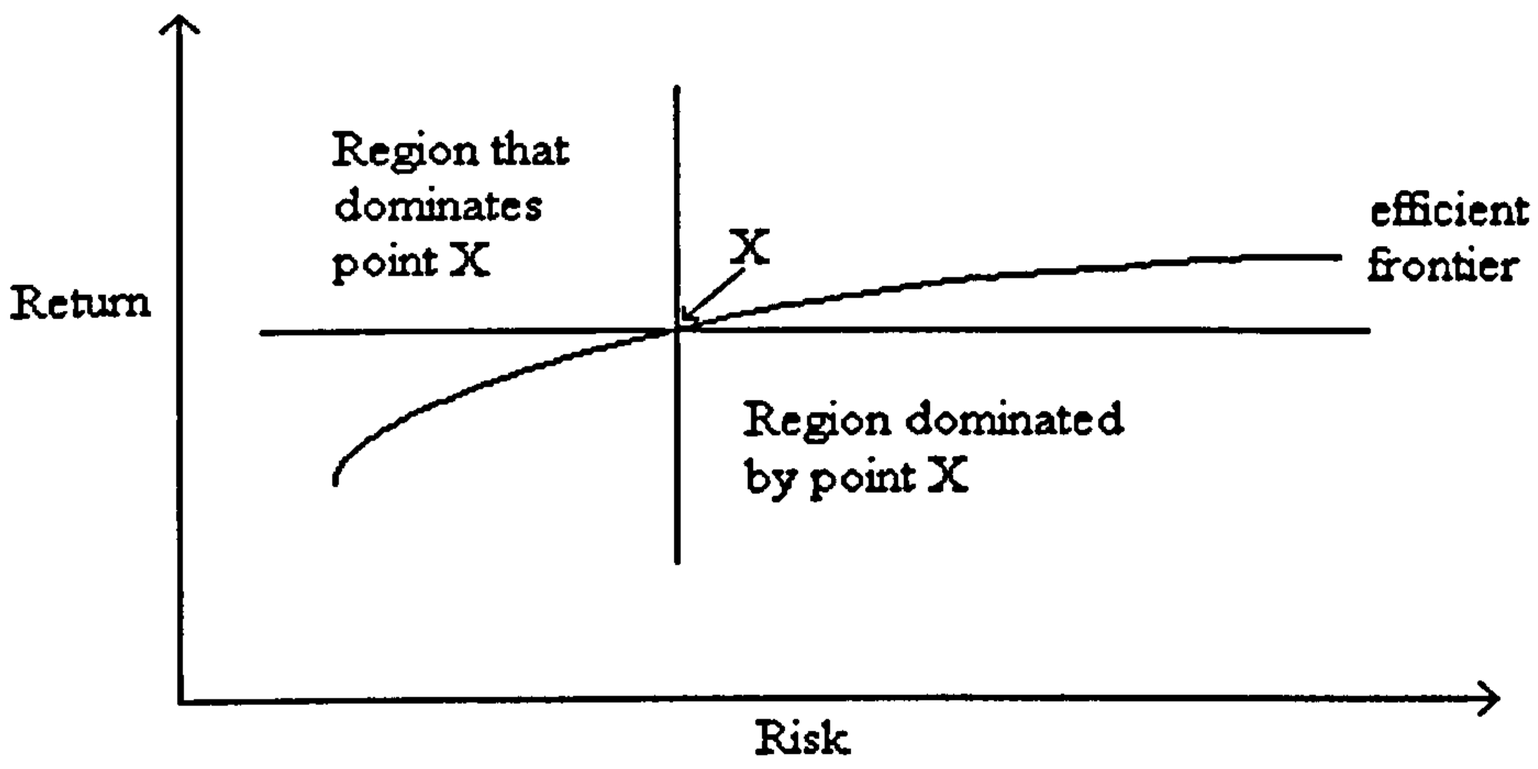
The condition of subadditivity is not met by VaR and shortfall probability, which are therefore not coherent risk measures. These two risk measures attach the same weight to losses below the specified threshold irrespective of the how far below the threshold the loss is. This means that it is possible to construct a combined portfolio that actually has a higher risk measure than any of its individual portfolios. Only in the special case of elliptical distributions will VaR satisfy the condition of subadditive. However, when portfolios contain non-standard assets that have high levels of kurtosis or skewness, these risk measures become unsuitable. CVaR was designed as an alternative to VaR that is a coherent risk measure meeting the condition of subadditivity.

4.2. Multiobjective Optimization

In the portfolio optimization problem the two objectives are to maximize expected return $Max[\mathrm{E}(W_T)]$ and to minimize the specified risk measure $Min[RiskMeasure(W_T)]$. This can be seen in Figure 4.1 where the region that dominates point X has both a higher expected return and a lower risk value. A solution is Pareto optimal if there is no other solution that dominates it. The set of all the nondominated solutions then constitute the Pareto optimal front, named after Pareto (1896). The aim of portfolio optimization is to either find the entire Pareto optimal front or a number of different points along the front.

The traditional approach of portfolio optimization is to either find the maximum expected return for a fixed risk level, or find the minimum risk level for a fixed expected return. This approach would need multiple executions in order to identify the Pareto frontier. In contrast, using a multiobjective approach may only require a single execution in order to identify the Pareto frontier.

Figure 4.1: Example efficient frontier



Deb (2001) states that the two main issues of multiobjective optimization are, firstly, to find a set of solutions as close as possible to the Pareto frontier, and secondly, to find a set of solutions as diverse as possible. Deb (2001) and Tan, Khor & Lee (2005) provide good references on multiobjective optimization.

In multiobjective optimization there are multiple objective functions, representing different characteristics of the problem, that must be simultaneously optimized. As one is typically unable to find a single optimal solution that is better than all other solutions (i.e. a solution where all individual objective functions are simultaneously optimal), tradeoffs need to be made between the different objective functions. This is because improving one of the objective functions may lead to the worsening of another objective function. The purpose of the optimization is then to find a set of nondominated solutions. These are solutions were it is not possible to find another solution that improves the value of one of the objective functions without simultaneously reducing the value for at least one of the other objective functions.

The general mathematical form of a multiobjective optimization is given below.

$X = (x_1, x_2,\ldots, x_n)$ the vector of variables of a problem

$F(X)= (f_1(X), f_2(X),,\ldots, f_m(X))$ the vector of functions that need to be optimized, where m is the number of attributes.

$\min\ f_1(x_1, x_2,\ldots, x_n),\ldots, f_m(x_1, x_2,\ldots, x_n),$ subject to $g_i(x_1, x_2,\ldots, x_n) \leq b_i$ (4.10)

A solution x dominates another solution y if $\forall i\ f(x_i) \leq f(y_i)\ \exists\ j \in 1,\ldots,n|$ $f(x_j) < f(y_j)$.

Subbu, Bonissone, Eklund, Bollapragada & Chalermkraivuth (2005) say that "Since evolutionary algorithms inherently work with a population of solutions, they are naturally suited for extension into the multiobjective optimization problem domain, which requires the search for and maintenance of multiple solutions during the search". They also say that evolutionary algorithms are not as sensitive to the shape or the continuity of the Pareto frontier compared to the traditional programming techniques.

Horn (1997) provides an overview of how evolutionary algorithms can be used to solve multicriterion decision problems. This type of problem typically involves two quasiseparable problem difficulties of search and multicriterion

decision making, where tradeoffs are made between conflicting objectives. Horn (1997) says that "evolutionary computation is in a unique position to address both search and multicriterion decisions because of its ability to search partially ordered spaces for multiple alternatives tradeoffs". The main difference between muticriterion and single criterion evolutionary algorithm applications is in the specification of the objective function. As search and multicriterion decisions are not independent tasks one needs to decide on the best way to integrate the search and multicriterion decision making, with the three main methods described below.

1. *Make multicriterion decisions before the search is carried out*

   This method changes the fitness landscape of the search space by aggregating the multiple objectives into a single objective. This single objective can then be used to totally order the solutions. A simple form of aggregation is a linear combination of the individual objective functions where the weights are set as constants before the search is carried out. The disadvantage of this method is that as the relative importance of the objectives is fixed prior to the search it does not produce a set of solutions on the Pareto frontier. Therefore, for portfolio optimization one is unable to find the market portfolio of risky assets using this method.

2. *Search before making multicriterion decisions in order to obtain the Pareto frontier*

   Pareto (1896) recognized that solutions can be partially ordered without needing to make any multicriterion decisions. The Pareto criteria says that one solution dominates another solution if it is at least as good in all attributes and better in at least one. However, some pairs of solutions are incomparable as neither solution dominates the other. Before multicriterion decisions are made a large number of dominated solutions are eliminated resulting in tradeoffs only being made between fewer solutions. The nondominated set of

solutions at any time is given by $P_{on\text{-}line}$. The aim of the method is to evolve $P_{on\text{-}line}$ towards the actual Pareto frontier P_{actual}.

Zitzler, Deb & Thiele (2000) found that the inclusion of elitism improves the performance of multiobjective evolutionary algorithms. Elitism in multiobjective optimization is more complex then in single-objective optimization due to the existence of an elite set rather than only an elite individual. Elitism can be achieved in an multiobjective evolutionary algorithm by maintaining an off-line set of nondominated solutions, $P_{off\text{-}line}$, found to date. This ensures that stochastic selection and variation methods cannot result in nondominated solutions being lost. The evolutionary algorithm could then either use an elitist policy to ensure that $P_{on\text{-}line} = P_{off\text{-}line}$ or occasionally insert members of $P_{off\text{-}line}$ back into $P_{on\text{-}line}$ during the search process. The Strength Pareto Evolutionary Algorithm (SPEA2) of Zitzler, Laumanns & Thiele (2001) is an example of an elitist method which maintains an archive of nondominated individuals.

The search before making multicriterion decisions approach is covered in the next three subsections which have been divided into

(i) Multiple single-objective searches

(ii) Single-objective simultaneous parallel searches and

(iii) Pareto dominance approach

In addition two further more advanced approaches are covered later in Chapter 7.

(iv) Cooperative coevolution approach

(v) Island model approach

Approaches (ii)-(v) use a cooperative population search, to simultaneously find multiple members on the Pareto frontier, due to the belief of greater efficiency due to the implicit parallel processing of schemata.

3. *Integrate the search and multicriterion decision making*

   This method starts by performing a multiobjective search in order to provide information on possible tradeoffs. Multicriterion decisions are then made in order to reduce the search space. This process of searching and multicriterion decision making is continued until only a single solution is produced.

Tan, Khor & Lee (2005) found that the Pareto dominance approach is the most popular approach used in multiobjective evolutionary algorithms. This was followed by a weights approach, where the multiple objectives are combined using weights into a single scalar objective. They also found the most popular supporting element for multiobjective evolutionary algorithms is the use of a distance measure in order to distribute individuals along the tradeoffs. This was followed by the use of elitism to assist the preservation of good solutions.

4.2.1. Multiple Single-Objective Searches

This method carries out multiple single-criterion independent searches in order to optimize different aggregations of the criteria. This approach might benefit from the concentration on specific areas of the criterion space. Tsoi, Wong & Fung (1995) use the linear aggregative approach to build up the whole efficient frontier by applying a single-criterion genetic algorithm that optimizes

$$F(x) = \beta f_1(x) + (1-\beta) f_2(x) \tag{4.11}$$

where β varies from zero to one in equal increments However, this approach is biased toward the convex portions of the frontier and may miss concave portions of the tradeoff curve, see Coello (2000).

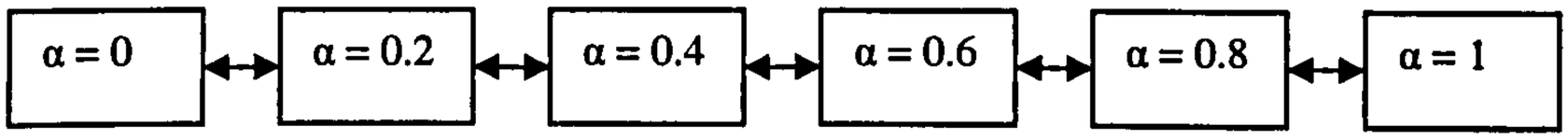
To find the efficient frontier in a portfolio optimization problem one could optimize the objective function

$$Max\left[\alpha \cdot \mathrm{E}(W_T) - (1-\alpha) \cdot RiskMeasure(W_T)\right] \tag{4.12}$$

for different values of the risk aversion parameters $\alpha \in [0,1]$ one at a time, e.g. as used in Chang, Meade, Beasley & Sharaiha (2000). However, when using an

evolutionary algorithm one can use an island migration approach with a distinct population for each risk aversion parameter that evolves in parallel. The advantage of this approach is that a good solution under one risk aversion parameter is unlikely too worsen so much that it no longer remains a good solution under an adjacent risk aversion parameter. This approach was used in the portfolio selection model of Loraschi & Tettamanzi (1995) and Loraschi, Tettamanzi, Tomassini & Verda (1995). In their stepping stone risk aversion island model each risk aversion parameter represents as island where individuals are periodically allowed to migrate between individuals of adjacent subpopulations, with their implementation using 20 such islands. I refer to their model as SSRAM(FM) with the 'FM' denoting the fixed migration rate between the adjacent islands. An example stepping stone island model with six islands in shown in the Figure 4.2.

Figure 4.2: Stepping stone risk aversion island model diagram



4.2.2. Single-Objective Simultaneous Parallel Search

This is a cooperative population search that does not use Pareto dominance in the selection process. Two common approaches are criterion selection and aggregation selection.

Criterion selection is a single-criterion parallel search in which fractions of the next generation of solutions are selected according to a single criteria at a time. An example is Schaffer (1985) which uses a Vector-Evaluated Genetic Algorithm (VEGA) to evaluate a fraction (1/m) of each new population based on the mth objective. The selected individuals from each of the m subpopulations are mixed and merged before recombination and mutation is carried out. This method is good for finding the extreme points on the Pareto frontier, but has the

drawback of underselecting middling individuals. Richardson, Palmer, Liepins, & Hilliard (1989) found that the process of mixing and merging the subpopulations is similar to fitness sharing each of the objectives.

The drawback of criterion selection is addressed by Murata & Ishibuchi (1995) aggregation selection method which uses a multiobjective genetic algorithm where the single criteria used for selection is based on a random linear combination of all the m criteria. The summed fitness value given by $F(X) = \sum_{i=0}^{m} w_i f_i(X)$, (4.13)

is then based on weights w_i's which are randomly varied, where $w_i \in [0,1]$.

An alternative way that the drawback of criterion selection could be addressed is by combining it with a Pareto dominance approach. This might be advantageous as criterion selection is good at finding extreme points on the Pareto frontier and Pareto dominance selection is good at finding middling points on the Pareto frontier.

4.2.3. Pareto Dominance Approach

This is a cooperative population search that uses Pareto dominance in the selection process and commonly uses niching to ensure diversity along the Pareto frontier. The beneficial effects of using niching is shown in Fonseca & Fleming (1995) as Pareto dominance selection alone without niching, results in individuals in a population tending to converge to arbitrary parts of the Pareto frontier rather than being evenly distributed.

Goldberg (1989) use a nondominated sorting method where the population is ranked according to Pareto optimality. The initial nondominated solutions in the population are given a rank of one and then removed from the population. The solutions remaining which are now nondominated are given a rank of two and then removed. This is continued until all the solutions in the

initial population are ranked. Niching methods are also used in order to ensure Pareto diversity.

Fonseca & Fleming (1993) use a Multiobjective Genetic Algorithm (MOGA) that ranks individuals according to their degree of dominance, i.e. the fewer solutions in the current population that dominate a particular individual the higher that individual's rank will be. The rank of individual i is given by rank(i) = $1 + q_i$ where q_i is the number of solutions that dominate individual i. The fitness is given by fitness(i) = inverse(rank(i)). Fitness sharing according to the niche count is then performed within each rank. The measure distance for the niche count is in the criterion space.

An alternative to ranking is given by Horn & Nafpliotis (1993). Their Niched Pareto Genetic Algorithm (NPGA) uses Pareto dominance tournaments with fitness sharing. Individuals in the current population compete in a binary tournament in which the dominate individual wins. If neither individual is dominant the winner is the individual with the lower niche count. The paper also identifies the need to balance selective pressure (domination) with diversity pressure (niching).

Zitzler & Thiele (1998) describe their Strength Pareto Evolutionary Algorithm (SPEA). At each generation, the nondominated solutions are stored in an external nondominated set and any dominated individuals are removed. Clustering is then applied to limit the number in the nondominated set. The fitness of an individual is determined only by reference to the external non-dominated set rather than whether individuals in the population dominate each other. All the solutions in the external nondominated set participate in selection and fitness sharing is also applied to maintain diversity of the population.

Deb, Pratap, Agarwal & Meyarivan (2002) use an elitist Nondominated Sorting Genetic Algorithm (NSGA2) which builds on the NSGA model of Srinivas & Deb (1995). The two main subroutines of the algorithm are a fast nondominated sorting algorithm based on Pareto dominance and a crowding distance algorithm. Their crowding distance algorithm has the advantage over

fitness sharing that there is no need to specify the fitness sharing parameter σ_{share}. It is also a less complex method being of order O(MPlogP) compared to fitness sharing which is of order $O(MP^2)$, where P is the number of individuals in the population and M the number of objective functions.

The crowding distance algorithm works by ordering the population according to fitness value for each objective function in turn. The first and last individual are assigned an infinite distance value, with the remaining individuals assigned a distance value equal to the average normalized distance of their two adjacent individuals. The crowding distance for each individual is then calculated as the summation of distance values across all the objectives. The larger an individual's distance measure is the less crowded their neighbouring area will be.

The selection method uses the crowding comparison operator to favour solutions that are uniformly spread out over the Pareto-optimal front. The solution with the higher rank is preferred to a solution with a lower rank. For two solutions with the same rank the one with the higher distance measure is preferred. This is similar to the method used in Horn & Nafpliotis (1993).

The overall structure algorithm works by initially creating a parent population ParPop, of size P. Each solution is given a fitness based on its rank, calculated using the nondominated sorting algorithm. The offspring population OffPop, of size P, is produced using binary tournament selection, recombination and mutation. From the second generation onwards elitism is used. The parent population and offspring population are combined to form ComPop of size 2P, which is then reduced to size P using the crowding comparison operator. The next offspring population is then produced and the process is repeated.

The effectiveness of NSGA2 has made it a very popular algorithm to use when solving multi-objective optimization problems. The NSGA2 algorithm was used by Lin, Wang & Yan (2001) and Mukerjee, Biswas, Deb & Mathur (2002) for the purpose of solving portfolio optimization problems.

The pseudocode for the nondominated sorting algorithm, crowding distance algorithm and overall structure algorithm for the NSGA2 method are given below, having been based on Tan, Khor & Lee (2005).

Input: A population (Pop) of size P
Output: Population ranking
Function NondominatedSorting(Pop)
Repeat for each individual i = 1 to P
Calculate n_i, the number of solutions dominating the individual i
Calculate Si, the set of individuals dominated by individual i
End Repeat
Q = set of individuals in Pop with $n_i = 0$
CurrentRank = 0
Repeat while Q is not []
R = []
CurrentRank = CurrentRank + 1
Repeat for each individual i in Q
Rank(i) = CurrentRank
Repeat for each solution j in set S_i
$n_j = n_j - 1$
If $n_j == 0$
Put individual j in R
End If
End Repeat
End Repeat
Q = R
End Repeat
Return Ranking
End Function

Input: A population (Pop) of size P
Output: Crowding distance
Function CrowdingDistance(Pop)
// $f_j^{\max} = \max(F(i,j) : \forall i = 1,2,...,P)$
// $f_j^{\min} = \min(F(i,j) : \forall i = 1,2,...,P)$
$CDtc(i) = 0 : \forall i = 1,2,...,P$
Repeat for each objective j
Idx = index vector based on the sorting of vector $F(i,j) : \forall i = 1,2,...,P$
CDtc(Idx(1)) = CDtc(Idx(P)) = ∞
Repeat for each individual i = 2 to P-1

$$CDtc(Idx(i)) = CDtc(Idx(i)) + \frac{F(Idx(i+1), j) - F(Idx(i-1), j)}{f_j^{\max} - f_j^{\min}}$$

End Repeat
End Repeat
Return CDtc
End Function

```
Output: Final ParPop
Function NondominatedSortingGA()
    Initialize generation counter: g = 0
    Initialize parent population ParPop of size P
    Initialize offspring population OffPop = []
    Repeat while stopping criterion is not met
        ComPop = ParPop ∪ OffPop
        Rank = NondominatedSorting(ComPop)
        ParPop = []
        rankP = rank of the Pth individual in ComPop sorted in ascending order of rank
        ParPop = individuals from ComPop with rank < rankP
        ParPopSize = number of individual in ParPop
        TPop = individuals from ComPop with rank value of rankP
        CDtc = CrowdingDistance(TPop)
        ParPop = ParPop ∪ {(P-ParPopSize) individuals with the largest CDtc}
        OffPop = GeneticOperators(ParPop)
        n = n + 1
    End Repeat
    Return ParPop
End Function
```

Typically values of the NSGA2 algorithm are population size 100, crossover rate 0.8 and mutation rate 0.01 e.g. as used by Zitzler, Deb & Thiele (2000) when testing the performance of different multiobjective evolutionary algorithms.

4.3. Multistage Portfolio Optimization Model

In this section I detail the general multistage portfolio optimization modelling framework which I shall use.

P_{it}^{s} denotes the price of asset i at the beginning of the time period t in scenario s.

w_{it}^{s} denotes the portfolio allocation weight in asset i at the beginning of the time period t in scenario s.

$w_{it}^{\min}$ and $w_{it}^{\max}$ are the minimum and maximum asset allocation weights for asset i at time t, such that $\sum_{i=1}^{N} w_{it}^{\min} \leq 1$ and $\sum_{i=1}^{N} w_{it}^{\max} \geq 1$.

A_{t}^{s} is the value of the assets at time t in scenario s.

The assumptions used are that the transaction costs of buying and selling security i is a percentage c_i of the traded asset value. In addition it is assumed that the investment strategy is self-financing.

The aim of the multistage scenario portfolio optimization is to find the efficient frontier such that the following conditions are satisfied

- *Full investment condition:* $\sum_{i=1}^{n} w_{it}^{s} = 1 \ \forall s \in S; \forall t$. (4.14)

- *Asset allocation limit constraints:* $w_{it}^{\min} \le w_{it}^{s} \le w_{it}^{\max} \ \ \forall s \in S$ (4.15)

  Lower and upper limit constraints may also be placed on asset allocation classes. Relative constraints may also be placed on individual asset allocations.

- *Initial full investment constraint:* $\sum_{i=1}^{N} w_{i0} P_{i0} = A_0$. (4.16)

- *Self-financing constraint:*

$$\sum_{i=1}^{N} w_{i,t-1}^{s} P_{i,t-1}^{s} (1 + r_{i,t-1}^{s}) - c_i \sum_{i=1}^{N} \left| w_{i,t-1}^{s} - w_{it}^{s} \right| P_{it}^{s} = \sum_{i=1}^{N} w_{it}^{s} P_{it}^{s} = A_t^{s} \qquad (4.17)$$

  $\forall s \in S; t = 1,...,T$.

- *Nonanticipativity condition:* At each node of the scenario tree nonanticipativity needs to be ensured so that the asset allocation weights are the same for all scenarios that are still undistinguishable. If $\{s\}_t$ denotes the set of scenarios which are indistinguishable from each other at time t then the nonanticipativity constraint can be written as

  $w_{it}^{s} = w_{it}^{s'} \quad \forall i, t, s, s' \in \{s\}_t$

  Each scenario is assumed to occur with the same probability.

4.4. Scenario Generation

4.4.1. Equity factor models and derivative pricing

Equity Factor Models

When one needs to model the relationship between a large number of equities it is more computational efficient to represent the returns of individual equities using a factor model. This is a linear model where individual equity return are represented as the sum of risk factor returns and weighted sensitivities to these risk factors. Factors can either be macroeconomic, fundamental or statistical.

Sharpe (1964) introduced the Capital Asset Pricing Model (CAPM) model which is given by

$$r_{it} = \alpha_i + \beta_i X_t + \varepsilon_{it} \tag{4.18}$$

where r_{it} is the return of asset i at time t, and X_t is the market return at time t. The values of α_i's, the relative mispricing of stock i, and β_i's, the sensitivity of stock i to the market risk factor, are calculated using linear regression. In the CAPM, market risk represents the undiversifiable risk that is common to all stocks, whereas stock specific risk can be eliminated through investment in a diversified portfolio. Therefore, the value of a stock's beta, which equates to the equity's covariance with the market, measures the riskiness of the stock relative to the market, so is the only factor that is priced by the market.

A main limitation of the CAPM is that a one factor model is unable to capture the large number of risk factors that affect the price of equities. Therefore, Ross (1964) introduced the Arbitrage Pricing Theory (APT) model which extended the CAPM to a more general multi-factor model given by

$$r_{it} = \alpha_i + \beta_{1i} X_{1t} + \beta_{2i} X_{2t} + ... + \beta_{ki} X_{kt} + \varepsilon_{it}, \tag{4.19}$$

where k is the number of factors specified.

This model has flexibility in the number and types of factors that can be used to model equity returns. The factors that can be used in this model can be classified

as macroeconomic factors (e.g. interest rate, inflation rate, GDP) and fundamental factors (e.g. market value, dividend yield, price/earnings ratio, book-to-price ratio, debt/equity ratio).

Statistical factor models are the main alternative to macroeconomic and fundamental factor models. These models identify the statistically most significant factors without needing to explain the meaning of the resultant factors e.g. principal component analysis (PCA). The high degree of collinearity between equity returns means that it should be possible to identify only a few important driving factors that can adequately model the evolution of equity prices. This is a computational efficient method due to the reduced dimension of using only the first m principal components and that the principal components are independent.

The method described in Alexander (2001) for implementing PCA is described below.

(i) Summarize historical returns data with T observations on k correlated assets or risk factors in a $T \times k$ matrix $\mathbf{Y}$. Normalize the $\mathbf{Y}$ data into a $T \times k$ matrix $\mathbf{X}$ where each column is standardized to have mean zero and variance one.

(ii) Let $\mathbf{W}$ be the matrix of eigenvectors of $\mathbf{X'X/T}$, and $\mathbf{\Lambda}$ be the associated diagonal matrix of eigenvalues which have been ordered according to decreasing magnitude of eigenvalue. The principal components of $\mathbf{Y}$ are given by the $T \times k$ matrix $\mathbf{P} = \mathbf{XW}$. This linear transformation of the original risk factor returns results in transformed risk factor returns $\mathbf{P}$ that are orthogonal and have variances equal to the eigenvalues in $\mathbf{\Lambda}$.

(iii) The $\mathbf{P} = \mathbf{XW}$ equation is equivalent to

$$\mathbf{x}_i = w_{i1}\mathbf{p}_1 + w_{i2}\mathbf{p}_2 + \ldots + w_{ik}\mathbf{p}_k \qquad (4.20)$$

or when using the first r of the k principal components only

$$\mathbf{y}_i = \mu_i + w^*_{i1}\mathbf{p}_1 + w^*_{i2}\mathbf{p}_2 + \ldots + w^*_{ir}\mathbf{p}_r + \varepsilon_i \qquad (4.21)$$

where $w^*_{ij} = w_{ij}\sigma_i$ and the error term is due to the approximation of only using the first r of the k principal components.

(iv) Taking variances of the formula for the y_i's gives

$$\mathbf{V} = \mathbf{ADA} + \mathbf{V}_\varepsilon \tag{4.22}$$

where the diagonal principal component covariance matrix is given by

$$D = \text{diag}(V(\mathbf{p}_1), \ldots, V(\mathbf{p}_r)), \tag{4.23}$$

$A = (w^*_{ij})$ is the $k \times m$ matrix of normalized factor weights and $\mathbf{V}_\varepsilon$ is the covariance matrix of the errors.

Ignoring $\mathbf{V}_\varepsilon$ gives the approximation $\mathbf{V} \approx \mathbf{ADA'}$.

Another advantage of PCA is the ability to allow for volatility clustering by extending this method to the Principal Component GARCH (also know as Orthogonal GARCH, O-GARCH) model of Alexander (2002).

Autoregressive Conditional Heteroscedasticity (ARCH) models were introduced by Engle (1982) as a way of capturing the time-varying nature of volatility present in financial time series. This was extended by Bollerslev (1986) into the Generalized-ARCH or GARCH model. GARCH models are able to capture volatility clustering that are seen in financial time series and are implied by positive autocorrelation in squared returns.

A univariate GARCH model consists of two equations. Firstly, a conditional mean equation, which is often of the form

$$r_t = c + \varepsilon_t \ , \tag{4.24}$$

although one may specify a time-varying conditionally mean. If the time series has a significant level of autocorrelation the Autoregressive AR(1) model

$$r_t = c + \alpha r_{t-1} + \varepsilon_t \tag{4.25}$$

may be used. Secondly, a conditional variance equation which describes the nature of the time-varying variance.

In the case of the O-GARCH model the principal components covariance matrix D is estimated using a GARCH model for each of the principal components. For example a GARCH(1, 1) model with the following specification may be used

$$\sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + \beta\sigma_{t-1}^2 \tag{4.26}$$

where $\omega > 0$ and $\alpha, \beta \geq 0$. The market reaction parameter α and the volatility persistence parameter β should sum to less than one in order to achieve convergence. The orthogonal GARCH model is a generalization of a factor GARCH model e.g. Engle, Ng & Rothschild (1990) that by design ensures that the factors are orthogonal. The independence of the principal component factors means that one is only required to estimate univariate GARCH models. Due to the small GARCH effects when generating weekly equity return series I will not be incorporating GARCH effects into my simulations.

Equity Derivative Pricing

Black & Scholes (1973) showed one can value European call and put equity options on a non-dividend paying stock. Their model assumed that the stock price follows a geometric Brownian motion given by

$$dS = \mu S dt + \sigma S dw \tag{4.27}$$

with constant mean μ and volatility σ parameters. They also assumed that the interest rate is constant, no transaction costs are incurred, short selling is allowed, security trading is continuous and all securities are perfectly divisible. By using the argument that there should be no riskless arbitrage opportunities their derived price for a call (c_t) and put (p_t) option at time t was,

$$c_t = S_t N(d_1) - Xe^{-r(T-t)} N(d_2)$$

$$p_t = Xe^{-r(T-t)} N(-d_2) - S_t N(-d_1)$$

$$d_1 = \frac{\ln(S_t / X) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}$$

$$d_2 = \frac{\ln(S_t / X) + (r - \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}} = d_1 - \sigma\sqrt{T-t} \tag{4.28}$$

where X is the option strike price and N(x) is the cumulative probability distribution function for a variable that has a standard normal distribution.

The delta of a call option is defined as

$$\Delta = \frac{\partial c}{\partial S}.$$

The value of the delta can be used by an investor to calculate how much stock needs to be owned in order to hedge against changes in the value of a call option caused by small changes in the underlying stock price.

Volatility smiles

Empirical data shows that the implied option volatility is not constant across different strike prices. The implied option volatility on the underlying stock increases as the stock price falls. The result of this is that the volatility used to price a low strike option is higher than that used to price an equivalent high strike option. The implied probability distribution of a stock price given by equity options has a fatter left tail and a thinner right tail than would be expected if the stock price followed a lognormal distribution.

Possible reasons that have been given for the existence of volatility smiles are detailed below.

- *Company leverage.* As the equity price of a company falls its debt-to-equity ratio increases making it a more risky investment, causing its volatility to increase.
- *Equity correlation.* During an equity market crash, the correlation between equities increases, reducing the benefits of diversification.
- *Risk Premium.* Higher levels of volatility increase the risk of equity investment. This justifies a rise in the equity risk premium, which is achieved by falling equity prices.
- *Crashphobia.* Investors fear of an equity market crash may be higher than that justified by historical data.
- *Investor wealth.* As equity markets fall, investors may become more risk averse due to their falling wealth levels. This may cause investors to become more nervous resulting in increased trading volumes in response to new market information causing volatility levels to increase.

Volatility term structure

The volatility used to price options also depends on the options time to maturity. Empirical data shows that equity prices are mean-reverting, which has two effects when calculating the appropriate volatility to use for option pricing. Firstly, if the current equity volatility is unusually low (high), there is an expectation that the volatility will increase (decrease) in the future, resulting in an upward (downward) sloping level of volatility for increasing maturity. Secondly, if the current equity volatility is equivalent to its normal expected level, then equity mean reversion will tend to cause the volatility over the medium to long term to decrease for increasing maturity. Similarly, if a stock has positive short term autocorrelation, this would also tend to cause the volatility over the short term to decrease for increasing maturity.

4.4.2. Data summary

Generating the simulations

In each simulation the Total Return Index (TRI) of each of the 10 industrial sectors of the FTSE ALL Share Index are calculated over weekly intervals over a one year period using principal component analyses. Using information on the initial weights that these sectors make up of the FTSE All Share index and the TRI for each industrial sector, the TRI for the FTSE All Share Index is calculated at quarterly intervals. The Black-Scholes formula is then used to calculate the price of call and put equity options on the FTSE All Share Index, using the assumption of constant interest rate and volatility. This is then repeated for the required number of simulations.

The data I am using in my investigations are the weekly Total Return Indices (TRI), which includes both capital gains and dividend payments, for the following UK equity indices: FTSE All Share, FTSE Actuaries 350, FTSE Small Cap, FTSE 350 Growth, FTSE 350 value, FTSE Actuaries 350 Higher Dividend,

FTSE Actuaries 350 Lower Dividend and the 10 FTSE All Share industrial sectors. The data period I am using is from the 22 August 1997, as this was when the Growth and Value indices began, to the 31 December 2005.

I have made the assumption that the risk free interest rate and expected return on the FTSE All Share are 4.5% p.a. and 7.5% p.a. respectively. Using the beta values for each industrial sector I have calculated the expected return on each industrial sector which is shown in Table 4.1. This was calculated using the CAPM and massaging the expected returns half way towards the market mean.

Table 4.1: Equity return statistics

Index Name	Percent composition of the FTSE All Share at 31/12/2005	Standard deviation (% p.a.)	Beta	Expected return (%p.a.)
FTSE All Share	100.0	15.7	1.00	7.5
FTSE Actuaries 350	96.7	16.0	1.02	7.5
FTSE Small Cap	3.3	14.2	0.62	6.9
FTSE 350 Growth	40.1	17.9	1.09	7.6
FTSE 350 Value	56.7	15.6	0.92	7.4
FTSE Actuaries 350 High Yield	52.1	16.1	0.91	7.4
FTSE Actuaries 350 Low Yield	44.6	18.6	1.11	7.7
FTSE All Share Basic Materials	6.7	19.7	0.81	7.2
FTSE All Share Consumer Goods	9.2	26.2	0.96	7.4
FTSE All Share Consumer Services	12.2	17.5	0.95	7.4
FTSE All Share Financials	27.4	21.1	1.17	7.8
FTSE All Share Health Care	8.5	18.0	0.72	7.1
FTSE All Share Industrials	7.4	23.0	0.97	7.5
FTSE All Share Oil & Gas	16.6	24.0	0.97	7.4
FTSE All Share Technology	1.1	41.3	1.68	8.5
FTSE All Share Telecom	7.0	27.4	1.14	7.7
FTSE All Share Utilities	4.0	15.1	0.40	6.6

The eigenvalues produced when PCA is applied on the series of 10 industrial sectors are shown in Tables 4.2. The first three factors are significant due to their eigenvalue being greater than 1, and combined these three factors explain 74% of the variability in the return series. Using these three principal components the resultant principal component factors are shown in Table 4.3.

Table 4.2: Principal component eigenvalues

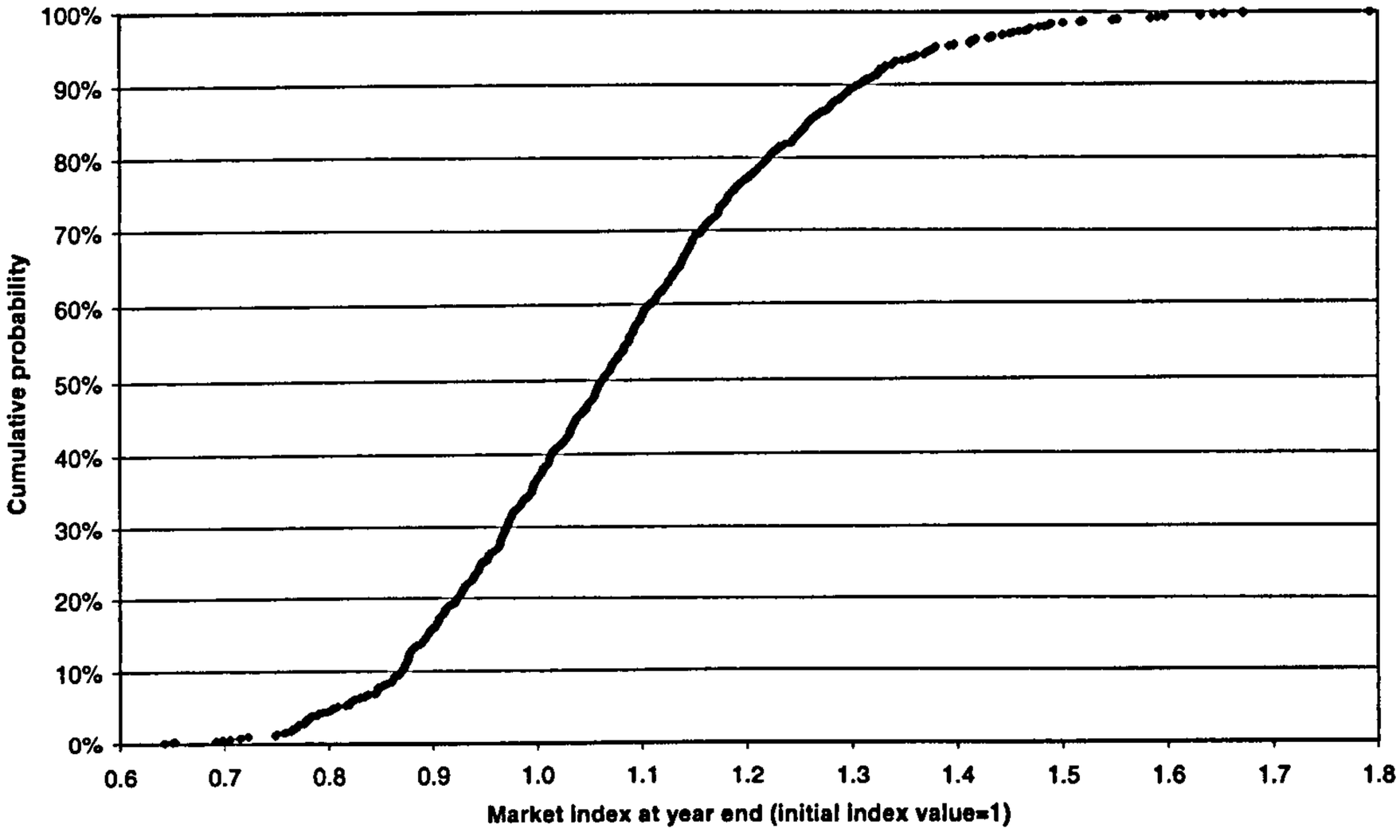
Principal component number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Eigenvalues	4.91	1.31	1.19	0.61	0.52	0.41	0.34	0.3	0.25	0.17
Explanatory power of each principal component	0.49	0.13	0.12	0.06	0.05	0.04	0.04	0.03	0.02	0.02
Cumulative explanatory power	0.49	0.62	0.74	0.8	0.85	0.89	0.93	0.96	0.98	1.00

Table 4.3: Principal components for the FTSE All Share industrial sectors

Equity Index	**Principal component number**		
	1	**2**	**3**
FTSE All Share Basic Materials	0.0099	0.002	-0.0084
FTSE All Share Consumer Goods	0.0091	0.0023	-0.0092
FTSE All Share Consumer Services	0.0108	-0.0074	0.0021
FTSE All Share Financials	0.0104	0.0029	0.0023
FTSE All Share Health Care	0.0067	0.0149	0.0098
FTSE All Share Industrials	0.0101	-0.0002	-0.0070
FTSE All Share Oil & Gas	0.0077	0.0089	-0.0040
FTSE All Share Technology	0.0081	-0.0188	0.0034
FTSE All Share Telecom	0.0063	-0.0157	0.0121
FTSE All Share Utilities	0.0050	0.0185	0.0102

Figure 4.3 shows the cumulative distribution of the market index value at the end of the year for the 810 simulations produced. The mean annual return and standard deviation for the simulations are 7.7% and 17.7% respectively.

Figure 4.3: Simulated one year equity market returns



4.5. Common Features of the Portfolio Optimization Problem

Assets

My generalized portfolio optimization problem consists of the following 17 assets.

- Equity indices: Assets 1 to 10 are the 10 industrial sectors which make up of the FTSE All Share Index. Therefore, the total equity asset allocation at time t is $Equity_t$ and is given by

$$\sum_{i=1}^{10} w_{it} = Equity_t \tag{4.29}$$

- Equity index call options: Assets 11, 12 and 13 are call options, with maturity at the end of the 1 year scenario projection period, on the FTSE All Share Index at strikes of $0.85MI_0$, MI_0 and $1.15MI_0$ respectively. MI_0 is the initial price of the FTSE All Share Index, i.e. market index price at time t = 0.

- Equity index put options: Assets 14, 15 and 16 are put options, with maturity at the end of the 1 year scenario projection period, on the FTSE All Share Index at strikes of $0.85MI_0$, MI_0 and $1.15MI_0$ respectively.
- Cash: Asset 17 is cash. The expected return used in the simulation is 4.5%p.a.

Asset allocation constraints

- Equity industrial sector indices: non-negativity constraint with relative bounds that are ±15% of the respective industrial sector weights at time 0 (w_0^i) in the FTSE All Share Index, so that

$$\max(w_0^i - 0.15,\ 0) \ \le\ \frac{w_{i0}}{Equity_0} \ \le\ \min(w_0^i + 0.15,\ 1) \quad . \tag{4.30}$$

$$\forall i = 1,\ 2,\ ...,\ 10,$$

- The constraints on the equity index call options, equity index put options and cash depend on the problem specification and are detailed in the following chapters.

Risk measure

The three risk measures I will be using are variance, generalized semi variance and lower partial moment (with a threshold level of -15% and moment of 2). I have chosen these risk measures as they are standard type risk measures that together can be used to compare the changes to the optimal asset allocation and efficient frontier caused by varying the level of importance placed on downside risk by an investor.

Trading costs

In order to compare the degree to which a dynamic trading strategy can mimic derivatives I am assuming frictionless markets. The cost of trading equities has been set to zero, i.e. $c_i = 0 \ \ for\ i = 1,2,...,10$. As my investigation is aimed at large equity portfolio management problems, it is also not unrealistic to assume

that there are no minimum trading unit constraints and no requirement to trade in integer units

4.6. Common Features of my Evolutionary Algorithms

Initial population

Two approaches have been used in the literature to ensure that an initial solution satisfies the simple case of minimum and maximum asset allocation weight bounds for each individual asset. In both methods the genotype for the initial portfolio is given by $g = (g_1, g_2, ..., g_N)$ where g_i is an integer between 0 and a positive constant g_{max}.

The first method is detailed in Baglioni, Costa Pereira, Sorbello & Tettamanzi (2000). The non-normalized weight vector and normalized weight vector is given by $w = (w_1, w_2, ..., w_N)$ and $\overline{w} = (\overline{w}_1, \overline{w}_2, ..., \overline{w}_N)$ respectively. The normalized weight vector **w** is then given by

$$w_{i0} = w_{i0}^{\min} + (w_{i0}^{\max} - w_{i0}^{\min}) \frac{g_i}{\sum_{j=1}^{N} g_j}, \tag{4.31}$$

This **w** vector will satisfy the $w_{i0}^{\min}$ and $w_{i0}^{\max}$ constraints but will not necessarily satisfy the constraint that $\sum_{i=1}^{N} w_i = 1$. The below procedure can then be used to obtain the normalized weight vector $\overline{w}$ from the vector **w**.

$$\text{Set } \Delta = 1 - \sum_{i=1}^{N} w_i \text{ and } r_i = \begin{cases} w_i^{\max} - w_i & \Delta \geq 0 \\ w_i^{\min} - w_i & \Delta < 0 \end{cases} \tag{4.32}$$

$$\text{The normalized weight vector is then given by } \overline{w}_i = w_i + r_i \frac{\Delta}{\sum_{j=1}^{N} r_j} \tag{4.33}$$

The second method is given in Chang, Meade, Beasley & Sharaiha (2000) who ensure that both minimum and maximum asset allocation constraints are

enforced by automatically ensuring that the lower limit constraints are satisfied and using an iterative process to ensure that the upper limit constrains are satisfied. The following pseudocode is based on their approach in the situation where there are no cardinality constraints.

Input: The genotype for the initial portfolio is given by $g = (g_1, g_2, ..., g_N)$ where g_i is an integer between 0 and a positive constant g_{max}. The minimum and maximum initial asset allocation weights given by w_{i0}^{min} and w_{i0}^{max} respectively. Q is the set of i assets.

Output: Feasible asset allocation

Function FeasibleInitialAssetAllocation()

$$L = \sum_{i=1}^{N} g_i$$

$$F = 1 - \sum_{i=1}^{N} w_{i0}^{min}$$ //F is the free portfolio proportion

$$w_{i0} = w_{i0}^{min} + g_i F / L \; \forall \; i \in Q$$ //g_i is interpreted as relating to the share of the free //portfolio proportion

R=empty set //R is the set of i whose proportions are fixed at w_{i0}^{max}

Repeat while there is an i $\in$ Q with $w_{i0} > w_{i0}^{max}$

For all i $\in$ Q **If** $w_{i0} > w_{i0}^{max}$ **Then** R=R $\cup$ [i]

$$L = \sum_{i \in Q-R} g_i$$

$$F = 1 - \left(\sum_{i \in Q-R} w_{i0}^{min} + \sum_{i \in R} w_{i0}^{max} \right)$$

$$w_{i0} = w_{i0}^{min} + g_i F / L \; \forall \; i \in Q - R$$

$$w_{i0} = w_{i0}^{max} \; \forall \; i \in R$$

End Repeat

Return $\overline{w} = (\overline{w}_1, \overline{w}_2, ..., \overline{w}_N)$

End Function

Both methods ensure that the asset allocation constraints and full investment criteria are met. However, the second approach is preferred due to its ability to more adequately cover the feasible search space. This can be shown by the simple example shown Table 4.4. There are 10 assets each of which has a 5% minimum weight bound and 20% maximum weight bound and the g_i's have been randomly generated using g_{max}=1000. The problem with the first method is that the

normalization process too strongly pulls the weight vector towards a value of 10%, whereas the second method allows more variability in each asset weight depending on the individual's g_i value.

Table 4.4: Example of two methods used for bounding asset allocations

Asset number (i)	w_{i0}^{min}	w_{i0}^{max}	g_i	$\frac{g_i}{\sum_{j=1}^{N} g_j}$	First method Normalized weight vector ($\overline{w}_i$)	Second method Normalized weight vector ($\overline{w}_i$)
1	5%	20%	974	21.20%	11.2%	15.6%
2	5%	20%	170	3.70%	9.3%	6.9%
3	5%	20%	956	20.80%	11.2%	15.4%
4	5%	20%	284	6.20%	9.6%	8.1%
5	5%	20%	325	7.10%	9.7%	8.5%
6	5%	20%	382	8.30%	9.8%	9.2%
7	5%	20%	403	8.80%	9.9%	9.4%
8	5%	20%	61	1.30%	9.0%	5.7%
9	5%	20%	524	11.40%	10.2%	10.7%
10	5%	20%	507	11.10%	10.1%	10.5%
Total			**4586**	**100.00%**	**100.0%**	**100.0%**

Table 4.5 details the general algorithm features and parameters that apply across the different evolutionary algorithms used to solve the portfolio optimization problem.

Table 4.5: Common features and parameters of my evolutionary algorithms

Representation	Floating point numbers. The representation chosen also ensures that all the solutions produced satisfy the problem constraints
Initial population	Random initialization
Population size	100
Parent selection method	Deterministic tournament selection: a copy of the population is made and pairs for the tournament from each population are randomly chosen without replacement
Recombination type	Uses a combination of discrete and weighted blending
Recombination rate	0.7
Mutation type	Uses a combination of random replacement, random arithmetic creep and random geometric creep
Mutation rate	0.05
Survivor Selection	Type SSRAM algorithms use a generational approach where the entire population is replaced by the offspring each generation. Type NSGA2 algorithms use Pareto dominance and crowding distance measure to maintain the fixed population size.
Termination criteria	When number of evaluations is reached
Main Output	Optimal asset allocation and the efficient frontier

4.7. Conclusions

My scenario generation method uses the computationally efficient method of principal component analysis to generate the equity return series. I have described how my scenario portfolio optimization problem can be divided into the two objectives of maximizing expected return and minimizing risk. Two common multiobjective optimization methods that can be used to find the efficient frontier are multiple single-objective searches for different risk aversion parameters (e.g. type SSRAM algorithms) or using a Pareto dominance approach (e.g. type NSGA2 algorithms). Finally, I have detailed common features of my optimization problem and my evolutionary algorithms that are applied in the following three chapters.

Chapter 5

Incorporating Derivatives in Portfolio Optimization

In this Chapter I investigate how the inclusion of derivatives shifts the efficient frontier and optimal asset allocation in a single period equity portfolio optimization problem as well as the degree to which this depends on the risk measure used.

5.1. Benefits of using Derivatives in Multistage Portfolio Optimization

Margrabe (1997) lists the main reasons for using equity derivatives as: reducing transaction costs, reducing taxes, selling short, mimicking a dynamic trading strategy, expressing views on correlation and volatility, creating non-linear payoffs, managing accounting and governmental rules and regulations, increasing leverage. I will be investigating how the inclusion of derivatives with non-linear payoff profiles shifts the efficient frontier for different risk measures. In Chapter 6, I will also be looking at how well a dynamic investment strategy can mimic a strategy that excludes the use of derivatives.

The ability to use derivatives in a portfolio optimization problem will be beneficial in situations where the inclusion of derivatives results in an outward shift in the efficient frontier, allowing an increase in expected return for the same level of risk. Below I give a summary of the two general situations when the use of derivatives may result in such an outward shift in the efficient frontier.

Using derivatives to approximate a dynamic investment strategy

The ability to carry out a dynamic investment strategy, where the multistage optimization allows asset allocation weights to be constantly altered, results in an improved efficient frontier over the situation where a single period optimization is performed. However, it is not always possible to implement an optimal dynamic investment strategy due to certain practical considerations such as lack of ability to trade continuously and the costs of implementing the dynamic strategy.

Haugh & Lo's (2001) closed-form results shows that under certain conditions a buy and hold portfolio of stocks, bonds and options can provide an excellent approximation to the comparable identified optimal dynamic investment strategy of holding varying amounts of stocks and bonds. Therefore, in situations where it is not possible to implement a given optimal dynamic investment strategy the addition of a well chosen portfolio of derivatives can result in an improvement in the efficient frontier of a similar order that would be achieved if it were possible to follow a dynamic investment strategy. In their theoretical model the objective was to maximize expected utility and their stock asset was modeled by a diffusion process.

Using derivatives in an incomplete market setting

Black & Scholes (1973) showed that in complete markets derivatives can be perfectly replicated through investment in equity and cash. When markets are complete the inclusion of derivatives in the portfolio optimization problem will not improve the efficient frontier as in complete markets derivative securities can be replicated by existing assets so are redundant assets. However, when markets are incomplete, due to infrequent trading or the presence of additional sources of uncertainty it becomes suboptimal to exclude derivatives from the optimization problem.

The paper by Liu & Pan (2003) show that allowing the use of derivatives in portfolio optimization makes it is possible to gain exposure to the risk factors

of volatility risk and jump risk, resulting in a possible improvement to the efficient frontier. In their setup an at-the-money option provides exposure to volatility risk and a deep out-of-the-money put option to negative jump risk. Their closed-form results show that derivatives which provide access to volatility risk can be used by non-myopic investors to take advantage of the time-varying nature of their opportunity set. Similarly, by providing access to jump risk, derivatives can be used by investors to disentangle their simultaneous exposure to diffusive and jump risks in the stock market. Their theoretical setup can be extended to include any other additional risk factors which result in market incompleteness.

5.2. Specific parts of the problem specification and evolutionary algorithm

Asset allocation constraints

- When equity index call options are allowed: The portfolio may be short call options up to a total amount of 100% of the equity exposure. This allows an investor to enhance their income by selling potential upside in their equity portfolio.
- When equity index put options are allowed: The portfolio may be long put options up to a total amount of 100% of their equity exposure. This allows an investor to protect their entire equity portfolio against a fall in equity prices.
- Cash: Non-negativity constraint with either an upper bound of 30% (i.e. $0 \leq w_{17,0} \leq 0.30$) or in a particular case 0%.

In the most general case, the combination of long stock and ability to short call options/go long put options, results in a cash like payoff when both options have the same strike. As I can already invest in cash, it is an undesirable feature to be able to achieve a cash like payoff from a combined stock and option strategy. Therefore, in order to avoid this I am placing the constraint that at least one of the call or put options in any of the same strike option pairs must have a zero asset

allocation weight. The same strike option pairs are $(w_{11,0}, w_{14,0})$, $(w_{12,0}, w_{15,0})$ and $(w_{13,0}, w_{16,0})$.

Given this constraint the combination of long stock, long put and short call can still create a rich variety of payoff profiles e.g.

- Long stock
- Short put like payoff: long stock and short call
- Long call like payoff: long stock and long put
- Bull spread: long stock, short high strike call, long low strike put
- Bear spread: long stock, short low strike call, long high strike put

Varying the degree to which derivative assets are a permitted asset

The following four degrees to which derivative assts are permitted will be investigated:

- Allowing one to sell equity call options and buy equity put options.
- Allowing one to sell equity call options but excludes the possibility to buy equity put options: $w_{it} = 0 \;\; for\; i = 14,15,16 \;\; \forall t$
- Allowing one to buy equity put options but excludes the possibility to sell equity call options: $w_{it} = 0 \;\; for\; i = 11,12,13 \;\; \forall t$
- Excluding both call and put options: $w_{it} = 0 \;\; for\; i = 11,12,...,16 \;\; \forall t$

The derivative strategies are limited to annual rebalancing (i.e. only the initial asset allocation needs to be set).

Problem representation

My problem solution is represented using gene values that hold all the information necessary to calculate the solution's asset allocation weights. The advantage of this approach is that the gene values are set up in a way that they can be initialized, and have variation operators applied to them, while ensuring that the resultant asset allocation weights are within the specified constraints.

Initializing the gene values

- Equity asset allocation split. Calculate the gene values $g_{i,0}$ for i=1,2,...,10 detailing the split of the equity asset allocation among the 10 industrial sectors using the method detailed in Chang, Meade, Beasley & Sharaiha (2000). This method ensures that the sum of these gene values is 1 and that all the equity industrial sector asset allocation constraints are satisfied.
- Strike indicators. For each of the three strike levels randomly choose either a 0 or 1 to indicate whether one can either sell calls or buy puts at that strike level. This is to ensure that the constraint that at least one of the call or put options in any of the same strike option pairs has a zero asset allocation weight is satisfied.
- Equity call and put options relative asset allocation weights. For each call option (i.e. i = 11, 12 and 13), assign $g_{i,0} = 0$ for each individual call option that was just given a zero asset allocation in the previous step and for the remaining call options assign a randomly generated U(0,1) value. Use the same method to generate the three gene values that will be used to calculate the put relative asset allocation weights.
- Equity call and put exposure. Generate two U(0,1) random variables $g_{call_exposure,0}$ and $g_{put_exposure,0}$ that assign the percentage of the upside potential to be sold, and percentage of the downside protection to be bought respectively.
- Cash asset allocation. Produce a random uniform number, $g_{cash,0}$ = U(0,1), that will be used to calculate the cash asset allocation.

Converting from gene values to asset allocation weights

(i) The cash asset allocation at time t is given by

$$w_{17,t} = w_{17,t}^{\min} + g_{cash,t}(w_{17,t}^{\max} - w_{17,t}^{\min}). \qquad (5.1)$$

(ii) The relative call weight (relcallwght) of each call option is calculated as $relcallwght_{i0} = \frac{g_i}{\sum_{i=11}^{13} g_i}$, (5.2)

which satisfies the requirement that $\sum_{i=11}^{13} relcallwght_{i0} = 1$. The same approach is used to find the put relative asset allocation weights.

Using $g_{call_expsoure,0}$ $g_{put_expsoure,0}$ the relative call weights and relative put weights and the initial price of the different equity options, calculate the individual asset allocation for each call and put option at each of the three strike levels.

(iii) Now that the cash allocation and equity call option asset allocations are known, the equity class asset allocation is calculated so that the sum of the asset allocations is 1. Using the equity class asset allocation and the asset split between the equity industrial sectors, the asset allocation in each industrial sector is calculated.

5.3. Computational Results

For my investigations in Chapter 5 and Chapter 6, my preferred algorithm for finding the efficient frontier and optimal asset allocations is the SSRAM(FM) algorithm rather than the NSGA2 algorithm. The main reason for this is the advantage of the type SSRAM algorithms at being able to directly compare points on the efficient frontier, and their resulting asset allocations, at a particular risk aversion parameter, under different risk measures and problem specifications.

While the SSRAM(FM) is found to produce a reasonable distribution of points along the efficient frontier when either variance or lower semi-variance is the risk measure, it fails when lower partial moment (with a threshold return of –15% and moment of 2) is chosen as a risk measure. Therefore, I have used the SSRAM(FM) algorithm when either variance or lower semi-variance is the risk

measure and the NSGA2 algorithm when lower partial moment is the risk measure. This particular short-coming of the type SSRAM algorithm, in that there is no guarantee that the points along the efficient frontier will be evenly distributed, is discussed in more detail in Chapter 7.

5.3.1. Asset allocation weights for different risk measures

In this section I compare how the asset allocation weights vary as the risk aversion parameter changes for the investment setup where both the shorting of call options and purchasing of put options are allowed. The risk aversion parameter (α) takes values between 0 and 1, where 0 results in the lowest risk portfolio and 1 the highest expected return portfolio.

Table 5.1 shows the asset allocation weights when variance is the risk measure. For a risk aversion parameter of 0 the asset allocation is 82% equities, -14% call options, 2% put options and 30% in cash (i.e. its maximum allowable allocation). As the risk aversion parameter is increased from a value of 0 to 0.3, the asset allocation of cash quickly falls to a zero allocation with the equity allocation increasing to its peak value of 115% when $\alpha = 0.3$. Over this range, the allocation to call options remains fairly constant having peaked at -17% for $\alpha = 0.2$, while the allocation to put options has fallen to a zero allocation by the time $\alpha = 0.3$. As the risk aversion parameter is increased further from 0.3 to 1.0, the equity allocation falls to a fully invested 100% and the call option a zero allocation by the time $\alpha = 0.9$. Over this range, there remains a zero allocation to put options and cash.

For high values of α the full investment given to equities is explained due to direct equity investment providing the highest expected return of all the asset classes. It is interesting to note that a significantly higher allocation is given to shorting call options than is given to buying put options when the risk measure of variance is used. One reason for this is that shorting call options allows one to

gain from the high expected returns equities offer through an increased equity allocation.

The composition of the call options between the low strike, at the money strike and the high strike shifts as the risk aversion parameter is increased. The results show that when $\alpha \leq 0.3$ it is preferable to short the low strike call option. As the risk aversion parameter is increased the allocation shifts firstly in favour of the at the money strike call option when $0.4 \leq \alpha \leq 0.5$ and then to the high strike call option for $\alpha \geq 0.6$. This behaviour can be explained by the property that shorting a call option with a lower strike level provides greater downside protection due to the higher premium received by the writer, but also gives up more upside equity potential.

Table 5.1: Asset allocations when variance is the risk measure

Asset	**The optimal % asset allocation for risk aversion parameter**										
	0	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
Overall portfolio allocation											
FTSE All Share	81.9	86.1	113.9	114.7	109.2	105.1	103.2	102.5	101.5	100.0	100.0
Call Options	-13.7	-14.6	-17.3	-14.7	-9.2	-5.1	-3.2	-2.5	-1.5	0.0	0.0
Put options	1.8	0.4	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Cash	30.0	28.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Equity portfolio allocation											
Basic Mats.	18.6	15.5	10.4	12.0	3.7	0.0	0.0	0.0	0.0	0.0	0.0
Cons. Gds	4.1	3.7	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0
Cons. Sers.	17.2	17.6	14.5	7.5	6.4	5.7	4.8	1.6	1.8	0.0	0.0
Financials	12.5	12.6	12.5	12.4	12.5	13.2	22.5	34.0	35.3	38.2	39.1
Health	5.5	0.0	3.0	4.5	5.1	12.8	21.3	23.5	23.4	22.1	21.2
Industrials	7.1	0.7	1.6	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0
Oil & Gas	14.3	21.6	31.1	31.3	31.6	27.0	10.4	1.6	1.6	1.6	1.6
Technology	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	6.7	16.1	16.1
Telecom	8.4	10.7	13.5	17.0	21.6	22.0	22.0	22.0	21.9	22.0	22.0
Utilities	12.4	17.7	13.4	15.2	19.0	19.0	19.0	17.2	9.4	0.0	0.0
Call option portfolio allocation											
Low strike	100.0	100.0	88.1	80.3	39.8	0.0	0.0	0.0	0.0	N/A	N/A
ATM strike	0.0	0.0	11.9	12.6	46.9	54.7	0.0	0.0	0.0	N/A	N/A
High strike	0.0	0.0	0.0	7.1	13.3	45.3	100.0	100.0	100.0	N/A	N/A
Put option portfolio allocation											
Low strike	0.0	0.0	0.0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
ATM strike	53.6	45.5	0.0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
High strike	46.4	54.5	100.0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Below I discuss how the asset allocation in the ten industrial sectors varies as the risk parameter is increased from 0 to 1, which is shown graphically in Figure 5.1. The sectors are grouped into three categories according to whether the allocation rises, falls or exhibits a hump as the risk aversion parameter is increased.

- *Sectors whose asset allocation fall as the risk aversion parameter is increased* - Basic materials, consumer goods and industrials have allocations when $\alpha = 0$ of 19%, 4% and 7% respectively, which falls to a zero allocation for $\alpha \geq 0.5$. The allocation to consumer services of 17% when $\alpha = 0$ also falls as the risk aversion parameter is increased, but only reaches a zero allocation when $\alpha \geq 0.9$. The explanation for this behaviour is that these sectors have neither a sufficiently high expected return nor diversification property to make them attractive to an investor whose main concern is to maximize expected return.
- *Sectors whose asset allocation exhibit a hump as the risk aversion parameter is increased* - The allocation to oil/gas starts at 14% when $\alpha = 0$ then peaks at the maximum allowable allocation of 32% when $\alpha = 0.4$, thereafter falling to the minimum allowable allocation of 2% for $\alpha \geq 0.7$. The allocation to the utility sector starts at 12% when $\alpha = 0$ rising to the highest allowable allocation of 19% for $0.4 \leq \alpha \leq 0.6$ then falls down to zero allocation for $\alpha \geq 0.9$. The low correlation that utilities have with the other sectors explains the reason why investment in utilities is desirable for medium risk aversion investors. The sharp fall in the allocation to oil/gas at high levels of α is likely to be due to the preference for investment in the highly correlated and higher expected return financials sector.
- *Sectors whose asset allocation rise as the risk aversion parameter is increased* – Financials start at the minimum allowable allocation of 12% when $\alpha \leq 0.4$. As the risk aversion parameter is increased the allocation to financial increases to a 39% allocation when $\alpha = 1$. The allocation to healthcare is below 6% for $\alpha \leq 0.4$ then rises to an allocation above 21% for

$\alpha \geq 0.6$. For telecom stocks, the allocation is 8% when $\alpha = 0$, rising to its maximum allowable amount of 22% for $\alpha \geq 0.4$. The technology sector starts with a zero allocation for $\alpha \leq 0.7$. As α is increased above 0.7 the allocation to technology increases quickly and reaches the maximum allowable allocation of 16% for $\alpha \geq 0.9$. The explanation for this behaviour is due to the higher expected returns these sectors offer, with the allocation to health care also being influenced by low correlation with the other sectors.

Figure 5.1: Equity portfolio split for the strategy where derivatives are allowed when variance is the risk measure

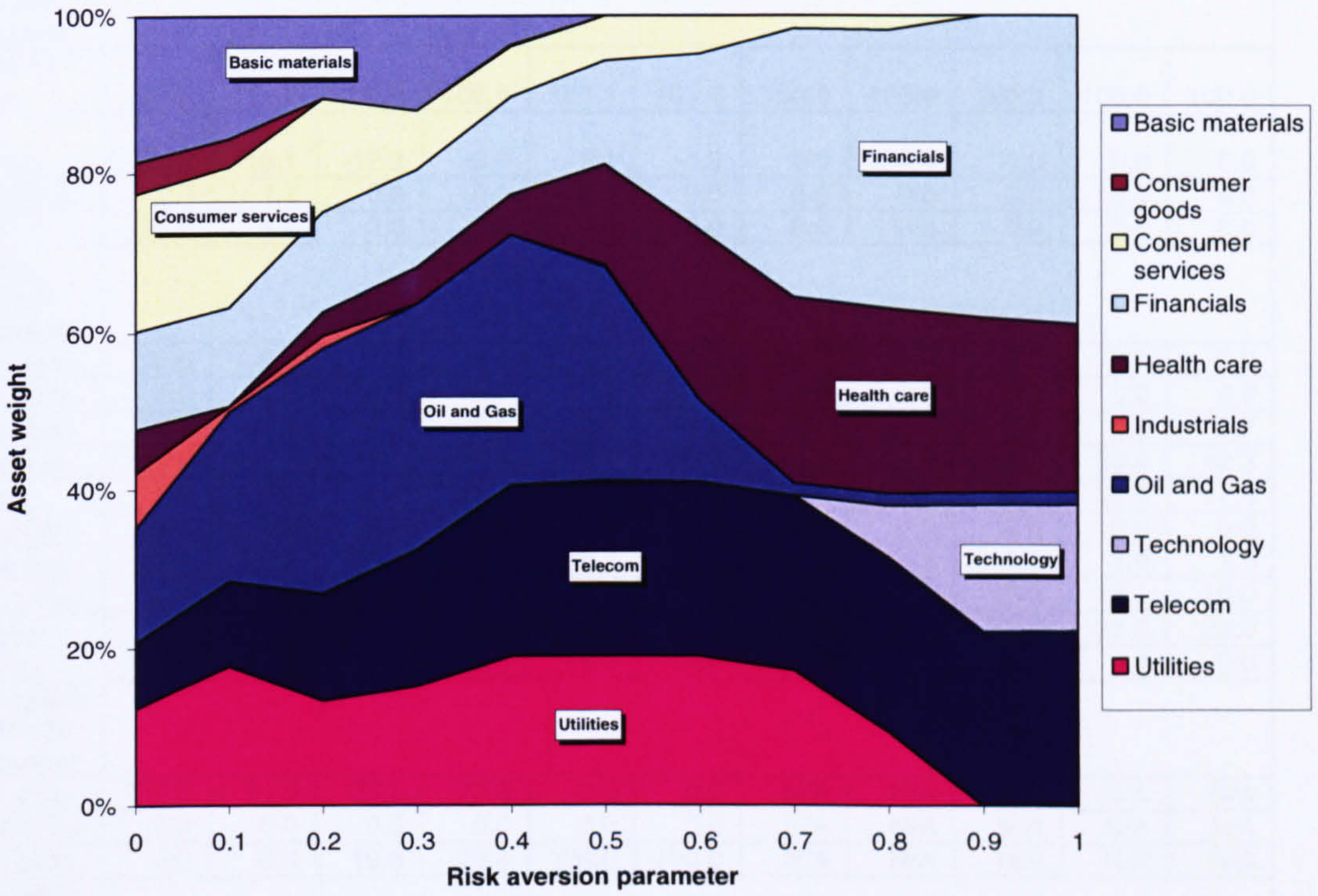


Table 5.2 shows the asset allocation weights when lower semi-variance is the risk measure. Below I describe the differences with the asset allocation when lower semi-variance rather than variance is used as a risk measure.

Portfolio asset allocation: There is a larger allocation to put options in the lower semi-variance case with the largest difference, from 0.4% to 1.7%,

occurring when $\alpha = 0.1$. The amount of money used to short call options also falls in the lower semi-variance case.

Equity asset allocation: The most significant change in the case of lower semi-variance is a shift from exposure in the oil/gas sector to the health care sector which occurs when $0.2 \leq \alpha \leq 0.6$. The largest fall in oil/gas is 23% and the largest rise in heath care 18% when compared to the variance case.

Table 5.2: Asset allocations when lower semi-variance is the risk measure

Asset	The optimal % asset allocation for risk aversion parameter										
	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Overall portfolio allocation											
FTSE All Share	81.4	111.9	112.2	108.6	103.1	101.7	100.0	100.0	100.0	100.0	100.0
Call Options	-13.5	-15.1	-12.8	-8.6	-3.1	-1.8	0.0	0.0	0.0	0.0	0.0
Put options	2.2	1.7	0.6	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Cash	30.0	1.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Equity portfolio allocation											
Basic Mats.	19.6	8.0	7.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Cons. Gds	2.9	1.2	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Cons. Sers.	15.4	14.5	7.0	6.2	9.0	3.0	7.8	0.0	0.0	0.0	0.7
Financials	12.5	12.7	12.4	12.4	18.1	26.3	21.4	30.9	36.1	39.4	42.3
Health	6.9	0.0	9.4	21.5	23.5	23.5	23.5	23.4	23.5	20.9	17.4
Industrials	9.9	2.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Oil & Gas	11.1	28.9	26.1	19.6	8.5	3.3	1.6	1.6	1.6	1.6	1.6
Technology	0.9	0.0	0.0	0.0	0.0	2.9	4.8	15.3	16.1	16.1	16.0
Telecom	7.0	14.2	18.9	21.6	22.0	22.0	22.0	22.0	22.0	22.0	22.0
Utilities	13.9	17.6	19.0	18.6	19.0	19.0	19.0	6.8	0.7	0.0	0.0
Call option portfolio allocation											
Low strike	100.0	91.6	89.2	79.6	0.0	0.0	N/A	N/A	N/A	N/A	N/A
ATM strike	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	N/A	N/A
High strike	0.0	8.4	10.8	20.4	100.0	100.0	N/A	N/A	N/A	N/A	N/A
Put option portfolio allocation											
Low strike	0.0	0.0	0.0	0.0	43.2	51.6	N/A	N/A	N/A	N/A	N/A
ATM strike	64.4	100.0	100.0	100.0	56.8	48.4	N/A	N/A	N/A	N/A	N/A
High strike	35.6	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	N/A	N/A

Composition of call/put options: In the case of lower semi-variance, the changing preference to higher strike call options as the risk aversion parameter increases remains, but no allocation is ever given to at the money strike call options. In the lower semi-variance case, a larger allocation is given to at the

money put options. Unlike in the variance case, exposure is also given to the low strike put option which can protect more effectively against downside risk.

Table 5.3 details the asset allocation weights when lower partial moment is the risk measure (with a threshold parameter of -15% and moment of 2). In this case the allocation in equities is in the range of 97% to 100% over the entire efficient frontier. Unlike in the case when either variance or lower semi-variance is used as the risk measure, it is never optimal to short any call options. This means that it is not possible for the equity allocation to ever exceed the 100% level.

Table 5.3: Asset allocations when lower partial moment is the risk measure

Asset	**The optimal % asset allocation for points on the efficient frontier ordered with increasing levels of risk**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Overall portfolio allocation										
FTSE All Share	97.6	99.2	99.4	99.4	99.6	100.0	100.0	100.0	100.0	100.0
Call Options	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Put options	0.9	0.8	0.6	0.6	0.4	0.0	0.0	0.0	0.0	0.0
Cash	1.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Equity portfolio allocation										
Basic Mats.	16.7	5.9	0.0	0.9	0.0	0.6	0.0	0.0	0.1	0.0
Cons. Gds	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.0	0.1	0.0
Cons. Sers.	10.8	4.9	0.0	0.1	0.1	0.1	0.0	0.0	0.1	0.0
Financials	12.4	15.5	13.9	33.4	30.6	32.2	23.8	19.9	26.9	35.5
Health	4.6	5.5	22.8	23.5	23.5	23.5	23.5	23.5	23.5	23.2
Industrials	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.0	0.1	0.0
Oil & Gas	29.0	30.0	23.6	1.7	1.7	2.6	1.6	1.6	1.7	1.6
Technology	0.0	0.0	1.3	0.1	4.3	0.0	9.9	16.0	16.1	16.1
Telecom	14.3	20.8	21.4	22.0	22.0	22.0	22.0	22.0	22.0	22.0
Utilities	12.3	17.3	17.1	18.1	17.7	18.8	19.0	16.9	9.4	1.6
Put option portfolio allocation										
Low strike	100.0	100.0	100.0	100.0	100.0	N/A	N/A	N/A	N/A	N/A
ATM strike	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	N/A	N/A
High strike	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	N/A	N/A

Allocation to buying put options is given to points on the lower risk half of the efficient frontier. The put option portfolio is made up entirely of the low strike put option, which gives protection to equity falls that are greater than 15%

(i.e. the strike level corresponds to the level used in the lower partial moment calculation). The low strike of this put makes it cheap to purchase, which means that a relatively small percentage portfolio allocation is able to protect a high proportion of the equity portfolio for falls that are greater than 15% while allowing the majority of the portfolio to be invested in equities. The effectiveness of the put to cover this downside risk also means that even the lowest risk portfolio only has a minimal 2% allocation to cash.

5.3.2. Efficient frontiers for the different derivative strategies

In this section I am comparing, under the different risk measures, the four efficient frontiers produced using the following degrees to which derivatives are allowed in the portfolio:

(i) where one is allowed to short calls and buy put options
(ii) can only short call options
(iii) can only buy put options
(iv) or can neither short call options or buy put options.

Figure 5.2 show the efficient frontier when variance is the risk measure and the maximum allowable cash allocation is 30%. One can clearly see that the best tradeoff between risk and return is achieved when one is allowed to short call options, with there being little difference as to whether or not one is also allowed to simultaneously purchase put options. The extra expected return achievable using strategies (i) or (ii) over strategy (iii) varies from greater than 1%p.a. for a target variance of 0.2%p.a. to a value of around 0.5%p.a. for a variance target of 2%p.a. It is not until a variance target of over 3% p.a. is set that there is no benefit of allowing the shorting of call options. The benefit achieved from allowing the shorting of call options is even more pronounced compared to the case when one is not allowed exposure to any option. This is because the extent

of the efficient frontier is drastically reduced in strategy (iv) with the lowest achievable variance being only 1.1%p.a.

When comparing the efficient frontiers of strategies (iii) and (iv), where one is unable to short call options, it can be clearly seen that there is a significant benefit of allowing the purchase of put options in cases where the target risk level is below 1.2% p.a.

Figure 5.2: Efficient frontiers for the different derivative strategies when variance is the risk measure and the maximum allowed cash allocation is 30%.

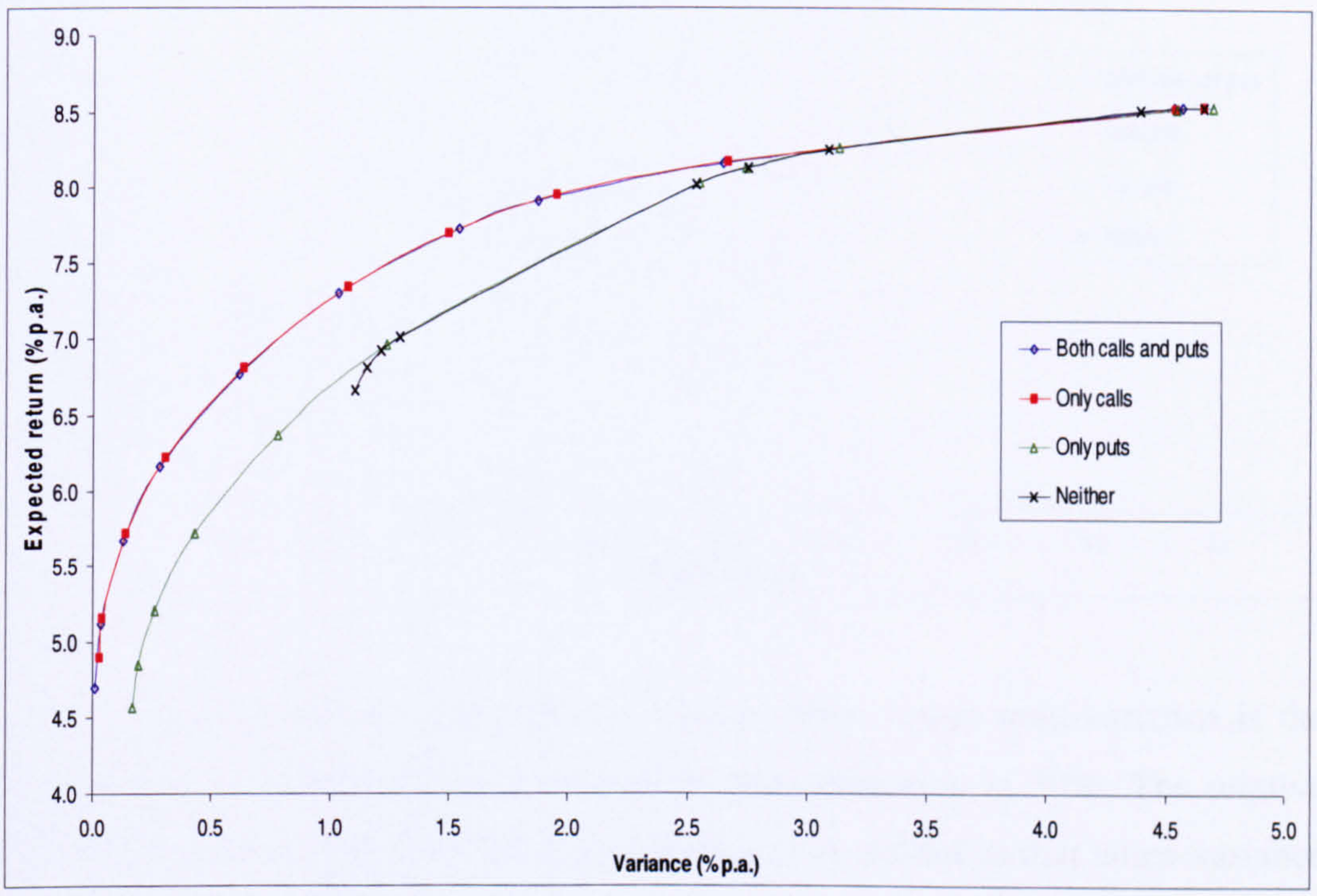


Figure 5.3 shows the efficient frontier when variance is the risk measure and the maximum allowable cash allocation has now been reduced to 0%. The most notable difference caused by restricting any investment in cash is the shift of the efficient frontier for strategy (iv) to the right with the lowest achievable variance having approximately doubled to 2.2%p.a. The benefit of allowing the shorting of call options (i.e. strategy (i) or (ii)) over strategy (iii) has increased for

the lower variance points on the efficient frontier. The lowest achievable variance for strategy (iii) has also approximately doubled to 0.35%p.a.

Figure 5.3: Efficient frontiers for the different derivative strategies when variance is the risk measure and no cash allocation is allowed

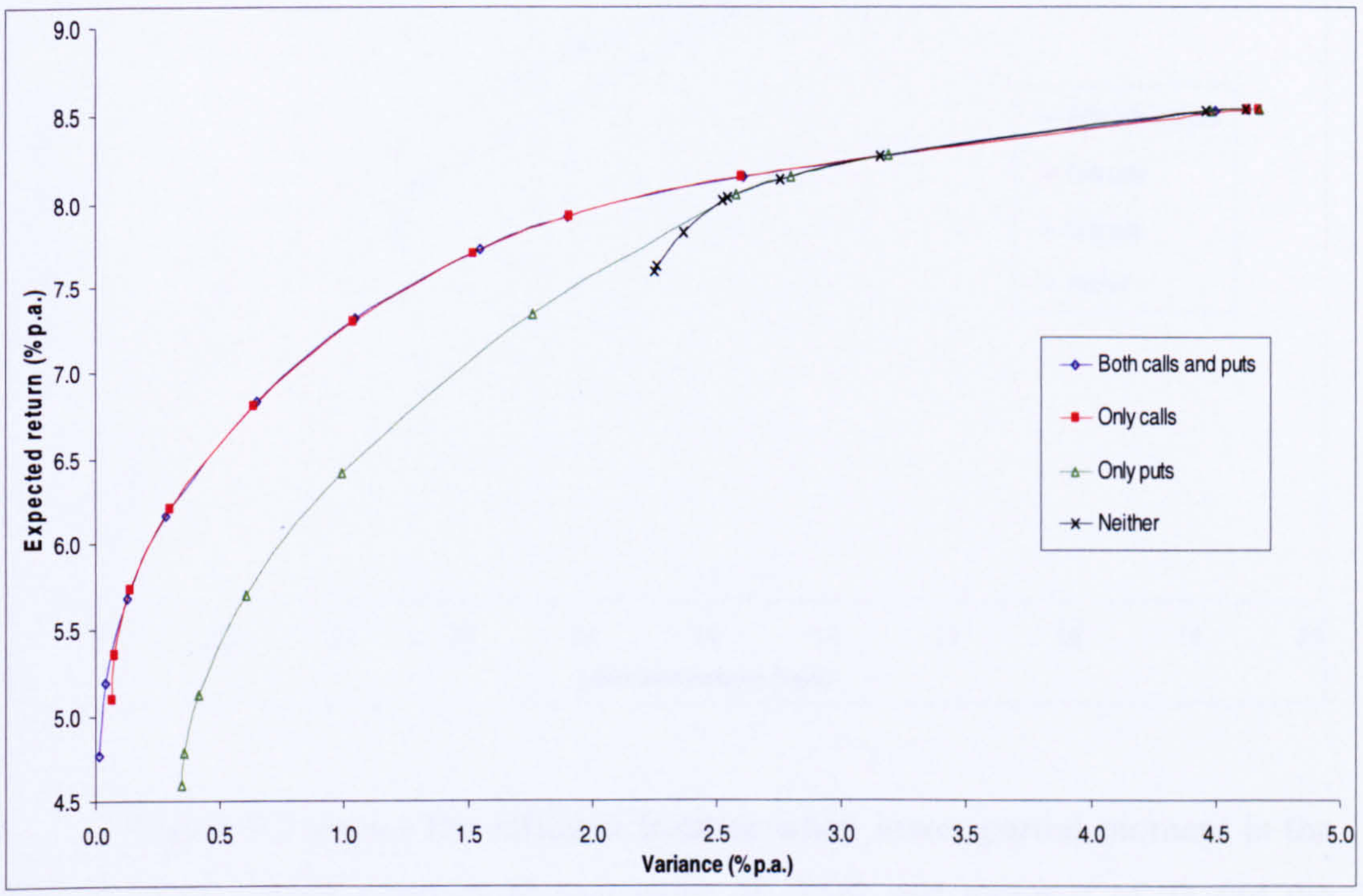


Figure 5.4 shows the efficient frontier when lower semi-variance is the risk measure and the maximum allowable cash allocation is 30%. The relative performance of the different derivative strategies is similar to that when variance is used as the risk measure. There are clear advantages of allowing the purchase of put options, i.e. strategy (iii), to the no derivatives strategy (iv). The efficient frontier also improves further when one is allowed the sale of call options (i.e. strategies (i) or (ii)).

Figure 5.4: Efficient frontiers for the different derivative strategies when lower semi-variance is the risk measure

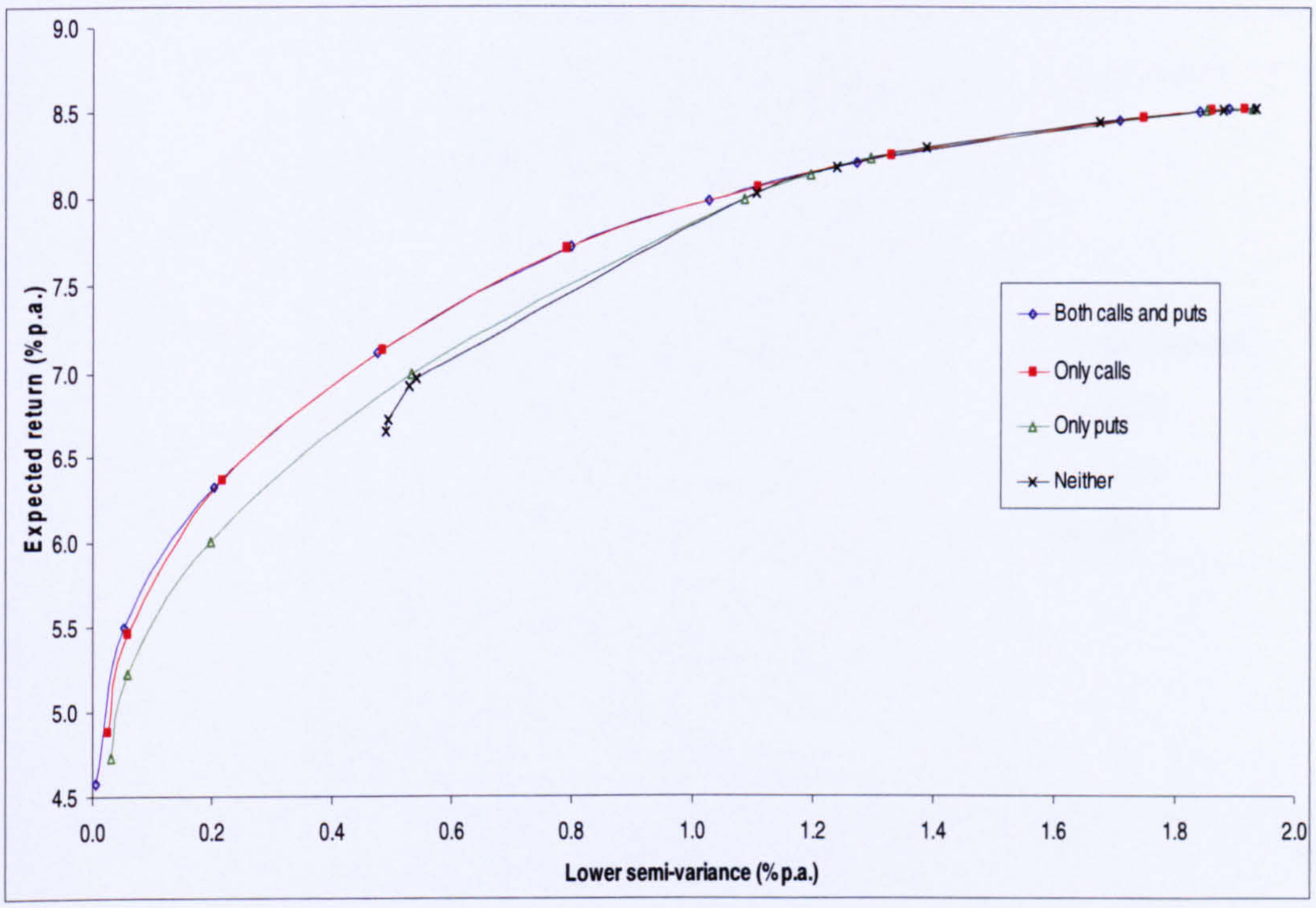
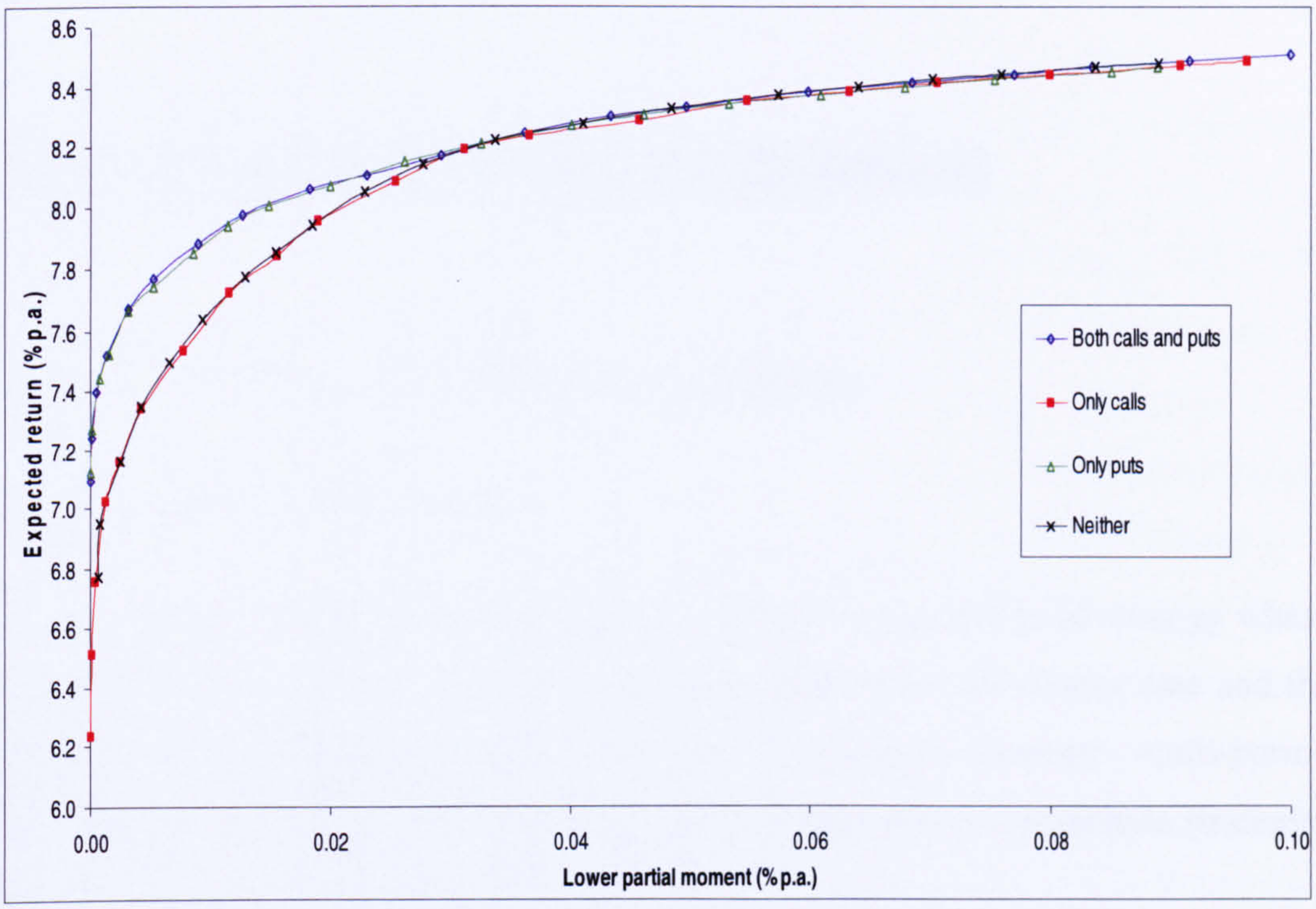


Figure 5.5 shows the efficient frontier when lower partial moment is the risk measure (using a threshold parameter of -15% and moment of 2) and the maximum allowable cash allocation is 30%. It shows that the best efficient frontier is produced when an investor is allowed to buy put options (i.e. strategies (i) and (iii)). The benefit in a higher expected return is 0.9%p.a. when the lower partial moment target is 0, which gradually falls to no extra benefit when the lower partial moment target reaches a level of 0.03%p.a. Unlike the previous cases, where either variance or lower semi-variance is the risk measure, the best efficient frontier is now achieved when the purchase of put options (rather than the sale of call options) is allowed. This is intuitive as the risk measure is now completely concerned with large equity market falls which can best be protected against by the purchase of put options. There is only a marginal improvement in the efficient frontier for an investor who is allowed to buy call options (i.e. moving from strategy (iv) to (ii) or from strategy (iii) to (i)).

Figure 5.5: Efficient frontiers for the different derivative strategies when lower partial moment is the risk measure



5.4. Conclusions

In this chapter I have demonstrated how the inclusion of derivatives can significantly improve the efficient frontier. I have shown how through the sale of call options and/or purchase of put options an investor is able to improve on the achievable risk/return tradeoffs over certain parts of their efficient frontier. To my knowledge the benefits of including call and put options within an equity portfolio under a heuristic optimization method has not yet been demonstrated elsewhere in the current literature. I have also shown that when placing a higher importance on downside risk within the risk measure an investor will increase their allocation in put options within their portfolio.

Chapter 6

Multistage Portfolio Optimization

6.1. Dynamic asset allocation strategies

6.1.1. Multi-period models

A static single period investment strategy is a buy and hold strategy where no rebalancing of assets is allowed in between the initial investment date and the end of the specified investment horizon. In contrast dynamic multi-period investment strategies allow rebalancing of asset allocations and include strategies such as constant-mix and portfolio insurance.

Trippi & Harrif (1991) define dynamic asset allocation as "a class of investment strategies that shifts the content of portfolios between two or more asset classes in response either to changes in the value of the portfolio and/or external economic states, on a more or less continual basis". The main benefits of dynamic asset allocation are to alter the distribution of the portfolio return at a future investment horizon and to exploit any predictable regularities such as market timing and tactical asset allocation strategies.

Below I discuss the benefits of multi-period models and describe some of popular asset allocation strategies.

Comparing the optimal asset allocation for investors with different investment horizons

In a single period model, where rebalancing is not allowed, one may wish to understand how the optimal portfolio is affected by the length of the investment horizon. One can show that if asset returns are uncorrelated, non-time

varying and normally distributed the mean-variance optimal portfolio will not depend on the time horizon. This is due to the fact that expected excess returns and variances will be scaled up by the time period T.

Having looked at the case when rebalancing is not allowed, the following cases allow rebalancing. Samuelson (1969), Merton(1969, 1971) and Mossin (1968) show that under the following two sets of conditions the optimal portfolio choice in the short-term for both a short-term and long-term investor will be identical. Firstly, if the investor has both power utility and returns are independently identically distributed (IID). As an investor with power utility has a constant relative risk aversion, portfolio allocation will not depend on the level of wealth and will therefore not depend on past asset returns. Furthermore, if asset returns are independently identically distributed, investment opportunities are not time-varying, so period returns are not predictable. Secondly, an investor who has a log utility function even if asset returns are not IID. A long-term log utility investor will have an optimal asset allocation that maximizes the expected long-term log return, which is simply the sum of the single time period log returns. Assuming that the portfolio allocation can be chosen freely at each time period, the long term portfolio will only be optimized when each of one period log utilities are maximized.

In the situation where rebalancing is allowed, it is assumed that there are no transaction costs, so the amount of rebalancing will not affect the overall portfolio return. In addition it is also assumed that there is no uncertainty in the estimated parameters, as uncertainty in mean returns might cause investors to reduce their allocation in the riskier assets as the time horizon increases.

A study by Thaler, Tversky, Kahneman & Schwartz (1997) found that investors have myopia loss aversion, which results in risky assets becoming less attractive as the frequency of portfolio evaluation increases.

Advantages of multi-period models

In practice investors do not implement a buy and hold strategy over a long term horizon, but tend to rebalance their portfolio according to changes in market conditions, their circumstance and risk tolerance resulting in the need of multi-period models. Therefore, one of the disadvantages of mean-variance analysis is the static single-period nature of the approach.

Brennan, Schwartz & Lagnado (1997) show that dynamic asset allocation strategies can take into account the time-variation in expected returns on different asset classes and the investor's time horizon. Mulvey, Pauling & Madey (2003) show that multi-period portfolio models result in improved portfolio performance over single time period models. Barberis (2000) show that an investor's optimal portfolio will change if investors update their return forecasts as time progresses and realizations occur.

The advantages of setting the asset selection problem in a multistage setting are summarized below.

(i) The ability to take advantage of changes in the expected level and volatility of asset returns (i.e. exploit any return or volatility predictability). For example, assets that have a serial negative correlation, or mean-reversion, over the time-periods considered will tend to have a higher optimal allocation. This is because of the lower level of risk over the entire investment horizon due to a high return period tending to be followed by a low return period.

(ii) The ability to alter the optimal asset allocations as the investor's circumstances or risk tolerance change, possible due to the varying levels of overall wealth.

(iii) Multistage models are able to more realistically account for the impact of transaction costs, including taxes, on the optimal dynamic asset allocations.

Insights into dynamic asset allocation strategies

Hodges (1995) discusses the effect of equity risk premium on asset allocation under the assumption that the level of the equity risk premium is negatively related to the level of the market. Investors with a long term horizon will require a smaller equity risk premium than short term investors. The result of this is a tendency for long term investors to follow contrarian investment strategies and short term investors to benefit from portfolio insurance.

Cox & Leland (2000) show that when using the assumptions behind Black-Scholes option pricing framework, such as constant equity volatility and frictionless markets, a dynamic asset allocation strategy is only efficient if it is path independent. Efficiency is taken as first order stochastic dominance, which is achieved when a desired portfolio return distribution is purchased as cheaply as possible. A path independent strategy occurs when the value of a portfolio at a future date is independent of the route taken by the market to reach that point. Using this result one can characterize any path independent dynamic portfolio strategy in terms of its contingent return profile at the investment horizon. A multi-period portfolio asset allocation problem can then be solved by calculating the optimal contingent return distribution at the investment horizon and then applying option pricing theory to determine the portfolio allocation at any intermediate date during this investment period.

Examples of inefficient path dependent strategies given by Dybvig (1988) are stop-loss (selling equities if equities fall below a certain prespecified level), lock-in (selling equities if equities rise above a certain prespecified level) and repeated short term portfolio insurance (where portfolio insurance is rolled over at regular specified intervals).

Constant proportion strategies

Constant proportion strategies determine the amount of equity investment according to the following equation,

$$\text{Equity Investment} = m(\text{Portfolio Value} - \text{Floor}), \tag{6.1}$$

where m is the specified multiplier.

The three main types of constant proportion strategies are:

- *Buy and hold strategies.* This is a constant proportion strategy that has a multiplier of one and a floor equal to the value of the risk free bond investment.
- *Constant-mix strategies.* Constant-mix strategies rebalance a portfolio in order that the asset allocation in equities is a constant proportion of total wealth. They have a multiplier between 0 and 1, corresponding to the percentage of assets invested in equities, and a floor of 0. In general, this strategy will buy (sell) equities after they fall (rise) in price relative to the other assets in the portfolio. This strategy has no downside protection and will perform well during periods when equities oscillate up and down in price.
- *Constant Proportion Portfolio Insurance (CPPI).* This is a constant proportion strategy that has a multiplier greater than one and a floor smaller than the initial portfolio value. It is detailed further in the next section.

6.1.2. Portfolio Insurance

Portfolio Insurance (PI) is one of the most common types of derivative-type strategies taken by investors. Its aim is to provide investors with a proportion of the upside exposure in risky assets while guaranteeing a minimum level of return over a specified time period and results in positively skewed portfolio returns. As investors tend to prefer positively skewed portfolio returns, in market equilibrium, investors will have a tendency to buy stocks when the market is high and sell stocks when the market is low. Three alternative ways it can be implemented are Option Based Portfolio Insurance (OBPI), see Leland (1988),

Constant Proportion Portfolio Insurance (CPPI) used in Perold & Sharpe (1995), and a dynamic replicating hedging strategy.

The OBPI strategy consists of an investment in a risky asset and the purchase of a put option written on the risky asset. This strategy ensures that at the maturity of the option the overall portfolio is at least equal to the strike price of the option.

The CPPI strategy provides investors with a simple risky asset allocation rule depending on the price of the risky asset given. Initially the investor sets a targeted floor level and uses this to calculate a 'cushion' (or risk capital) which is the difference between the current portfolio value and the present value of the targeted floor. A multiplier is then also specified, where the allocation to the risky asset is given by the multiplier times the risk capital. In this way, the CPPI guarantees that the floor will be protected if there is an instantaneously market drop in the risky asset of one over the multiplier. For example, consider the case where the value of the portfolio is 100, the present value of the target floor is 80 (i.e. risk capital of 20) and the multiplier is 4. The allocation in the risky asset will be 80%, so even if the market falls by one over the multiplier, i.e. 25% the overall value of the portfolio will be equal to the present value of the targeted floor.

Gradual falls (rises) in the risky asset will result in disinvestment (investment) in the risky asset, so this is a trend-following strategy. As the cushion approaches zero, the exposure to the risky asset also approaches zero. The larger the multiplier, the higher the turnover costs and volatility risk will be. High multipliers will be desired in periods of rising equity prices and modest volatility. It can be shown that the dynamic replicating delta hedging strategy is a generalization of the CPPI strategy where the size of the multiplier depends on the price of the risky assets.

Leland (1988) describes the type of investors who should buy PI. Firstly, investors who, as the level of wealth increases, have a risk-aversion that falls faster than the risk-aversion of the market. This is because such investors will

want to increase their equity allocation as their wealth level and equity prices increase. A typical example would be a liability driven investor, e.g. a pension fund, who is likely to have a risk aversion that increases faster with falling wealth (due to pressures on funding requirements) than that of the average investor in the market. Secondly, investors with above average expectations of future equity performance will also benefit from PI, as this allows them to invest more aggressively in equities while limiting their downside risk.

Option Based Portfolio Insurance vs. synthetic replication

There are several risks inherent in the CPPI and dynamic replicating hedging strategies that are not present in OBPI.

- *Jump risk.* In the CPPI strategy an instantaneous fall in the risky asset of more than one over the multiplier of the risky asset will fail to protect the targeted floor. One of the key assumptions underlying the dynamic replicating hedging strategy is that equity prices are continuous.
- *Volatility risk.* If realized volatility is higher than expected volatility the costs of implementing CPPI and synthetic replication increases due to the higher amount of rebalancing needed. A higher amount of rebalancing raises the number of times equity must be bought at a high price and sold at a low price as well as pushing up turnover costs.
- *Interest rate risk.* If risky assets fall significantly in a crash scenario, interest rates may also fall due to the increased demand for risk free assets. However, in this case, the lower reinvestment rates available may mean that the targeted floor level is not reached.

Synthetic replication will be preferred over OBPI in situations where the investor views these risks are less probable or personally damaging as compared to the average option writer. For example, an investor who believes that the expected ex post realized volatility will be lower then the ex ante expected volatility used by option writers to price the put option will be more likely to

prefer to synthetic replicate the put option rather than to purchase it directly from an option writer.

Resetting dynamic asset strategies

The characteristics of a dynamic asset strategy can be greatly affected by the resetting rules used. A rolling OBPI strategy that involves rolling the horizon so that it is always a fixed time period away from the beginning combined with resetting the horizon floor level as a constant proportion to the assets is equivalent to a constant mix-strategy. This would not achieve the aim of portfolio insurance as constant-mix strategies offer no downside protection. A rolling OBPI strategy can be achieved by ensuring the floor is kept at a nominal level.

When using the CPPI strategy one possible resetting policy is to carry out the initial specified rules when markets are flat or falling, in order to reduce downside risk, and lock in profits by raising the floor in rising markets. However, this will often result in selling equities in both falling and rising markets.

Limitations of Portfolio Insurance

- PI can only be effectively applied to risky assets that are very liquid due to increased difficulty and costs of hedging less liquid assets. By excluding investment in less liquid assets, an investor is foregoing the diversification benefits and liquidity premium that would be achieved by including less liquid risky assets in their portfolio.
- If PI is periodically rolled over, the central limit theorem says that, the desired single-period positively skewed returns will become lognormally distributed in the long run.
- Overhedging will arise when PI is only applied to a portion of an investor's risky asset portfolio, as this portion will not be perfectly positively correlated with the remaining risky assets.

Portfolio Insurance strategies for liability driven investors

Liability driven investors need to ensure that the PI strategy they undertake is most suitable in light of their individual liability structure. The standard PI strategy aims to protect an investor's total wealth level and consists of an investment in equities protected by a put option. This strategy results in a synthetic cash position as equity markets fall. However, cash has a zero duration so is a very risky asset in the context of an investor with long-term liabilities. The significant risk for a liability driven investor is that the risky asset performs badly relative to a bond with an equal duration to that of the average liability. Therefore, the worst case scenario occurs when equities markets fall and bond markets rise simultaneously. Therefore, a more appropriate strategy would be to buy an option that has a payoff equal to the outperformance of long-term bonds over equities in situations when bonds outperform equities. The aim of this strategy is to protect the investor's solvency ratio, rather than the total wealth level.

6.2. Methods for Generating Scenarios for Multistage Optimization

Gulpinar, Rustem & Settergren (2004) describe three alternate ways of generating price scenarios in order to solve a multistage portfolio optimization problem. Firstly, in the simulation based approach scenarios are the centroids of simulation clusters generated either sequentially or in parallel. Secondly, in the optimization approach economic scenarios are generated to ensure that their statistical properties match with the desired return distributions. This is achieved by either solving a non-linear optimization problem at each node or a large non-linear programming problem. Thirdly, in the hybrid approach of simulation and optimization, simulated prices are fixed as the centroids of the simulation clusters and optimization is used to find the probability of branches in the scenario tree. I am going to use the simulation based approach which allows a more realistic

account of the model uncertainties and is an efficient approach for the generation of large scenario trees.

Simulation approach to multistage scenario generation

Below is a summary of the simulation approach to scenario generation given by Gulpinar, Rustem & Settergren (2004). It is based on a procedure of simulation and randomized clustering to generate the event tree which can then be imputed into the multistage optimization problem.

(i) *Initialization* – create a root node with N scenarios and initialize all the scenarios with today's economic prices.

(ii) *Simulation* – simulate economic returns over a single time period for each of these N scenarios.

(iii) *Randomized seeds* - randomly choose a number of distinct scenarios (seed points), one for each of the desired branches in the tree, around which to cluster the remaining scenarios.

(iv) *Clustering* – group each of these remaining scenarios with the seed point which is the closest to its center according to a distance measure. The suitability of the clusters is determined according to a set of criteria e.g. the relative sizes of the clusters and the statistical properties of the clusters. If the resulting clustering is unacceptable, return to stage (iii). Execution times can be improved through the use of intelligent seed replacement that is linked to any constraints being violated.

(v) *Centroid selection* – once an acceptable clustering is found, each cluster is represented as a single point, called the centroid, which is determined as the scenario closest to the center of the cluster.

(vi) *Queuing* – a child scenario tree node for each cluster is created, with probability proportional to the number of scenarios in the cluster, and the clusters scenarios and centroid are installed. The stages (ii)-(v) are then repeated until the determination of the final tree which will consist of the centroids of each node and their branching probabilities.

Parallel and sequential simulations

The two methods of scenario simulation are parallel and sequential. Both methods can benefit from the use of pseudo-random numbers and low-discrepancy quasi-random sequences.

The parallel simulation method is equivalent to generating a large number of N linear scenarios for the entire time horizon before the process of clustering them into a scenario tree even begins. The purpose of clustering the scenarios is to ensure that the non-anticipativity condition is achieved. As you progress further down the tree the size of each cluster becomes smaller, which can be justified by viewing the need for accurate forecasts nearer the end of the time horizon as being less critical. The decrease in the cluster size as time progresses means that the number of scenarios needs to be set high enough in order to ensure that the leaves of the tree have sufficient resolution.

In sequential simulation, the scenarios of a node are initialized to that node's centroid before updating. As this cannot be done until the clustering at the previous level has been determined the centroid scenarios cannot be generated out to the time horizon before clustering. The number of scenarios at each node is constant.

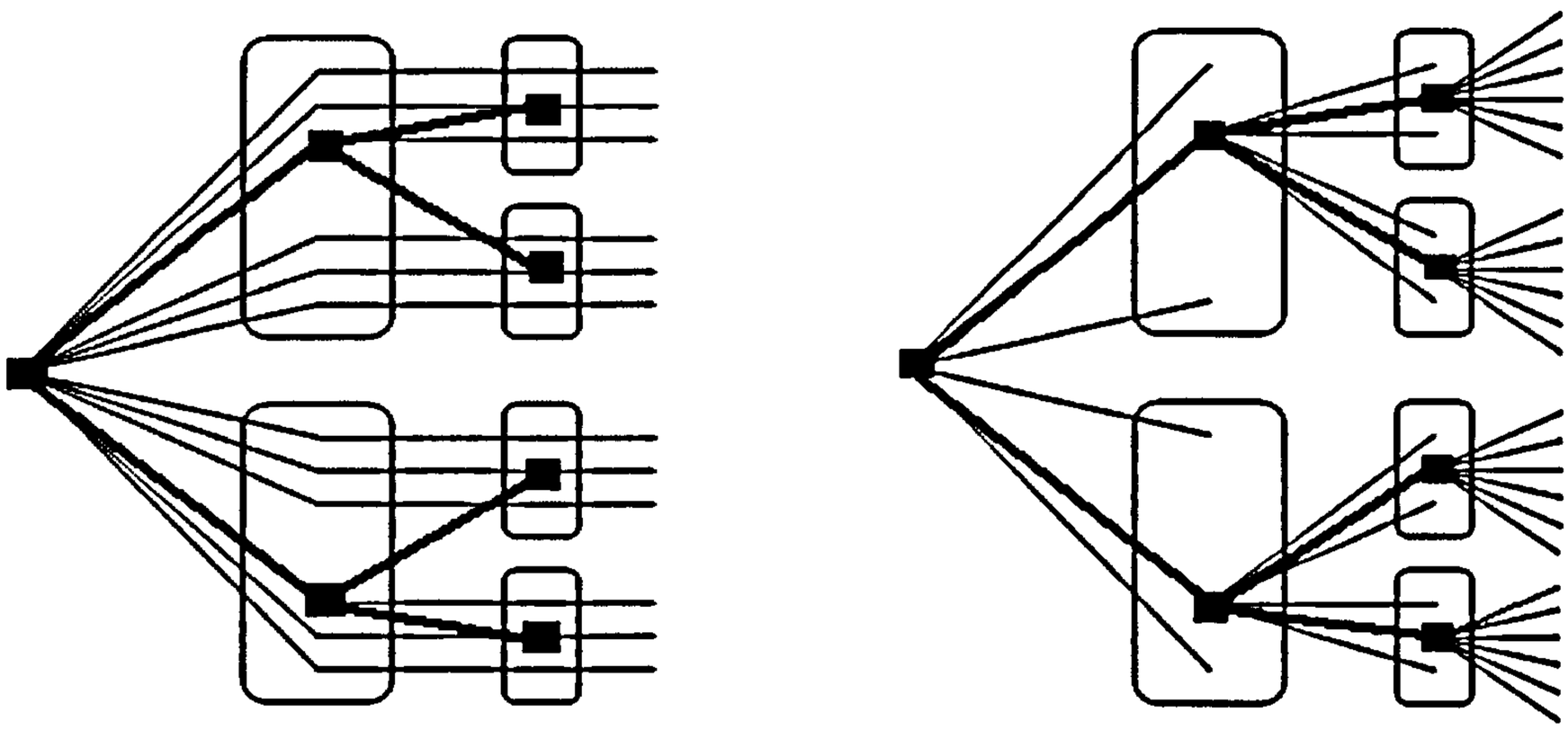
The parallel simulation method will result in a tree with more realistic extreme events due to the combination of the small cluster sizes at the leaves and the fact that in the sequential simulation method the simulations are restarted from the centroid. The two approaches can also be combined, so that the parallel method is initially used with a smaller number of N scenarios and then the sequential method is used once the number of scenarios in each cluster becomes too small to divide into sub-clusters of appropriate size.

Figure 6.1 shows an example of how to produce a two-stage scenario tree using either the parallel simulation or sequential simulation approach. Thin lines are used to represent the simulation paths and ovals represent simulations that are

clustered together. Each cluster point, denoted by a black square, will have the same asset allocations in order to ensure that the nonanticipativity condition is met. The heavy lines that link up the cluster points represent the scenario tree.

The parallel simulation example shows how a scenario tree can be produced from 12 simulations where the cluster size halves as we move along the tree. The example for the sequential simulation uses a constant number of simulations (i.e. 6) that are projected from each cluster point on the scenario tree.

Figure 6.1: Parallel simulation (left diagram) and sequential simulation (right diagram) example



Generating the scenario tree from the simulations

I am using a parallel simulation approach, with 810 simulations, in order to generate three scenario trees covering a 1 year projection period for annual, semi-annual and quarterly portfolio rebalancing frequencies. Each scenario tree is made up from a different structure of these same 810 simulations. One of the reasons for favouring parallel simulation over sequential simulation is that the use of these same simulations makes the computational results more easily comparable across the varying rebalancing frequencies. At each valid rebalancing date, each scenario cluster branches into 3 new clusters, each one broadly representing either a market rise, market neutral or market fall scenario.

The structure of the scenario tree in terms of the number of scenario clusters and number of simulations in each scenario cluster at each time period is shown in Table 6.1 for each of the three rebalancing frequencies.

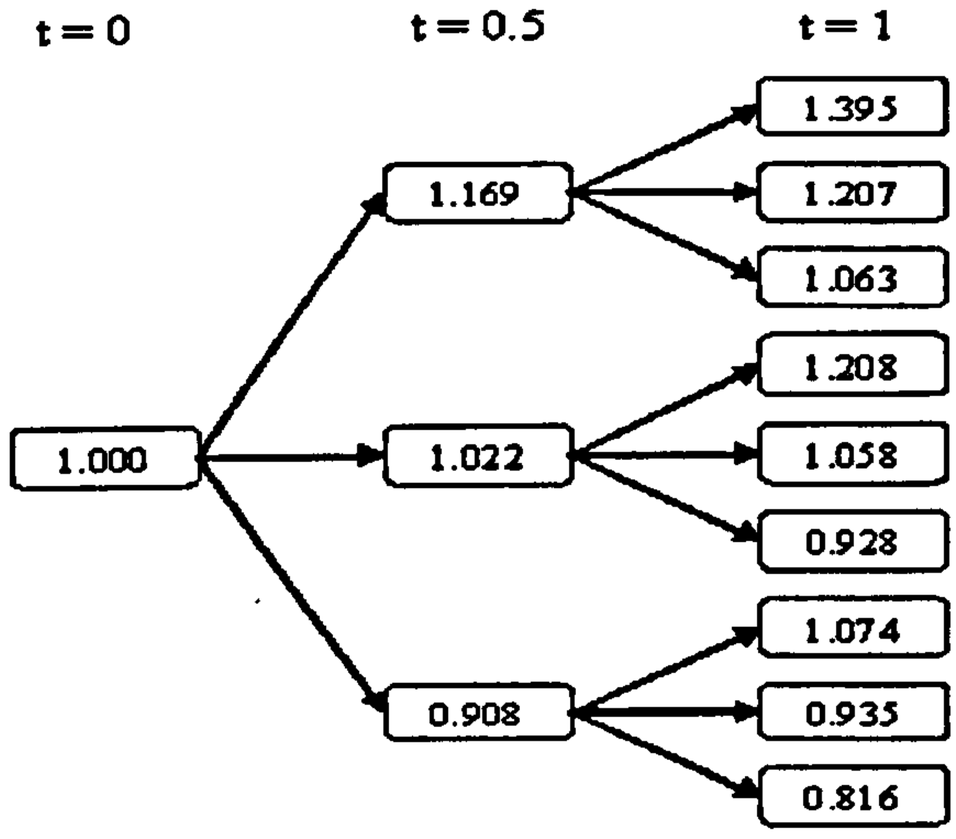
Table 6.1: Scenario tree structure

Time t (in years)	Annual rebalancing (1-stage tree)		Semi-annual rebalancing (2-stage tree)		Quarterly rebalancing (4-stage tree)	
	Number of scenario clusters	Number of simulations in each cluster	Number of scenario clusters	Number of simulations in each cluster	Number of scenario clusters	Number of simulations in each cluster
0.00	1	810	1	810	1	810
0.25	1	810	1	810	3	270
0.50	1	810	3	270	9	90
0.75	1	810	3	270	27	30
1.00	1	810	9	90	81	10

The annual rebalancing just involves setting the initial asset allocation. The semi-annual rebalancing involves both setting the initial asset allocation and the three asset allocation at time 0.5 for each of the scenario clusters. The quarterly asset allocation involves setting 40 asset allocations (1 at time 0, 3 at time 0.25, 9 at time 0.5 and 27 at time 0.75).

Figure 6.2 shows the 2-stage scenario tree of the equity market (initial value of 1) that summarises the actual 810 simulations into cluster groups that are used for the semi-annual rebalancing. Each cluster at $t = 0$ and $t = 0.5$ are divided into three clusters at the following time step, where the clusters are arranged in terms of decreasing equity market returns. At the end of the 1-year simulation horizon, six clusters show a positive overall return with the remaining three a negative return. The scenario tree can also be seen to exhibit positive skewness (i.e. by comparing the highest cluster return of 39.5% and the lowest cluster return of -18.4% around the mean return of 7.7%).

Figure 6.2: 2-stage scenario tree of the equity market return



6.3. Specific parts of the problem specification and evolutionary algorithm

The main aims of the investigations in this Chapter are firstly, to analyze the effects of allowing a more frequent rebalancing period for an equity strategy that excludes the use of derivatives and secondly to compare the ability of a multi-period equity strategy that excludes the use of derivatives to a single period equity strategy that allows derivatives. Therefore, to achieve these aims while keeping the problem specification as simple as possible I have disallowed the use of derivatives in the multi-period asset allocation strategies,

i.e. $w_{it} = 0 \;\; for\; i = 11,12,...,16 \;\; \forall t$. The implication of this choice is that I will not be investigating the possibility of a combined benefit of allowing both the inclusion of derivatives and a dynamic investment strategy in the portfolio optimization problem.

The two variations of the rebalancing period used are semi-annual (i.e. one rebalancing date) and quarterly (i.e. 3 rebalancing dates). The following three investment strategies are investigated.

(i) Constant mix strategy – The solution genes consist of the equity asset allocation split between the 10 industrial sectors and the multiplier (g_M) value, which can take values between 0.7 and 1. This range of possible multiplier values ensures that the cash allocation is in the range $0 \leq w_{17,0} \leq 0.30$. The equity investment at each node in the scenario tree is then given by

$$\text{Equity Investment} = g_M(\text{Portfolio Value}) \quad (6.2)$$

which is equivalent to a constant equity asset allocation after each rebalancing. The relative split between the industrial sectors is also fixed as the initial split given by the solution genes.

(ii) Constant Proportion Portfolio Insurance strategy (CPPI) – The solution genes consist of the equity asset allocation split between the 10 industrial sectors the multiplier (g_M) value, (which can take values between 1 and Multiplier(max)) and the floor (g_F) value (which can take values between Floor(min) and 1). I have set Multiplier(max) = 4 and Floor(min) = 0.5. The equity investment is then given by

$$\text{Equity Investment} = g_M(\text{Portfolio Value} - \text{Floor}) \quad (6.3)$$

This means that the initial equity investment can be between 0 and 200%. The rate of borrowing is assumed to be at 5.5%p.a. which is 1%p.a. higher than the risk free saving rate. The relative weights of the industrial sectors in the equity portfolio will remain constant through the scenario, with only the split between cash and the equity portfolio being altered by the dynamic investment strategy.

For the CPPI strategy the equity allocations at each rebalancing date is calculated, using the multiplier and floor parameter values, for each scenario cluster using the average value of the portfolio for all those simulations within that scenario cluster. This means that for a CPPI dynamic investment strategy at a particular rebalancing date, a scenario cluster that represents a market rise will have a higher equity asset allocation than a scenario cluster that represents a market fall. This is in

contrast to the constant mix investment strategy, where the rebalanced asset allocation is independent of both the time of the rebalancing and the cluster scenario.

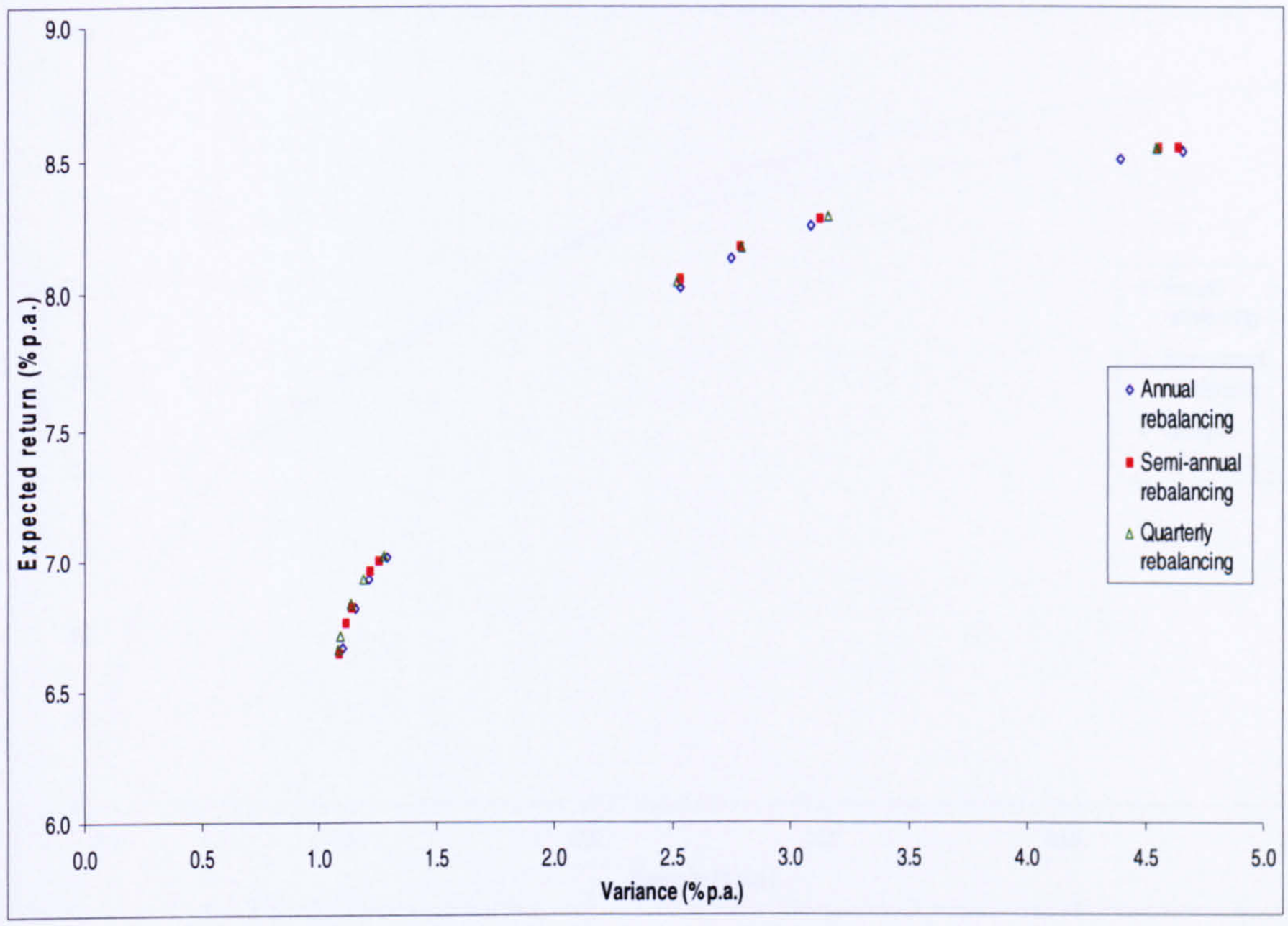
(iii) Fixed equity cluster strategy. In this strategy, the solution gene consist of the equity asset allocation split between the 10 industrial sectors and the equity asset allocation at each of the nodes on the scenario tree (i.e. 4 for semi-annual rebalancing and 40 for quarterly rebalancing). The equity asset allocation is allowed to be set between 0 and 150% at each point, with the borrowing rate at 5.5%p.a.

6.4. Computational Results

6.4.1. Constant Mix Strategy

Figure 6.3 shows the efficient frontier produced for the three different rebalancing frequencies for a constant mix investment strategy where variance is used as the risk measure. It shows that changing the rebalancing frequency makes little difference in the efficient frontier produced. The same effect is also seen when the risk measure is changed to either lower semi-variance or lower partial moment. This result can be explained by the fact that the method for producing the equity market returns should mean that they do not inherently exhibit autocorrelation of returns. For example if equity markets were mean reverting over the time horizon considered then a constant mix strategy, that sell assets after they rise and buys them after they fall, should result in a rebalanced strategy having a better efficient frontier. Therefore, on the modelling assumptions used, it is not beneficial to rebalance the portfolio at either semi-annual or quarterly periods. Figure 6.3 provides an example on how the points on the efficient frontier produced by the SSRAM(FM) algorithm are not always evenly spaced out.

Figure 6.3: Efficient frontier for a constant mix strategy when variance is the risk measure



6.4.2. Constant Proportion Portfolio Insurance Strategy

Figures 6.4 shows the efficient frontiers for the CPPI strategy for the three rebalancing frequencies considered when variance is the risk measure. It shows that there is only a small improvement in the efficient frontier when a more frequent rebalancing frequency is allowed. However, moving from annual rebalancing to quarterly rebalancing frequency yields an additional expected return of 0.5% p.a. when the variance target is above 15% p.a.

Figure 6.4: Efficient frontier for a CPPI strategy when variance is the risk measure

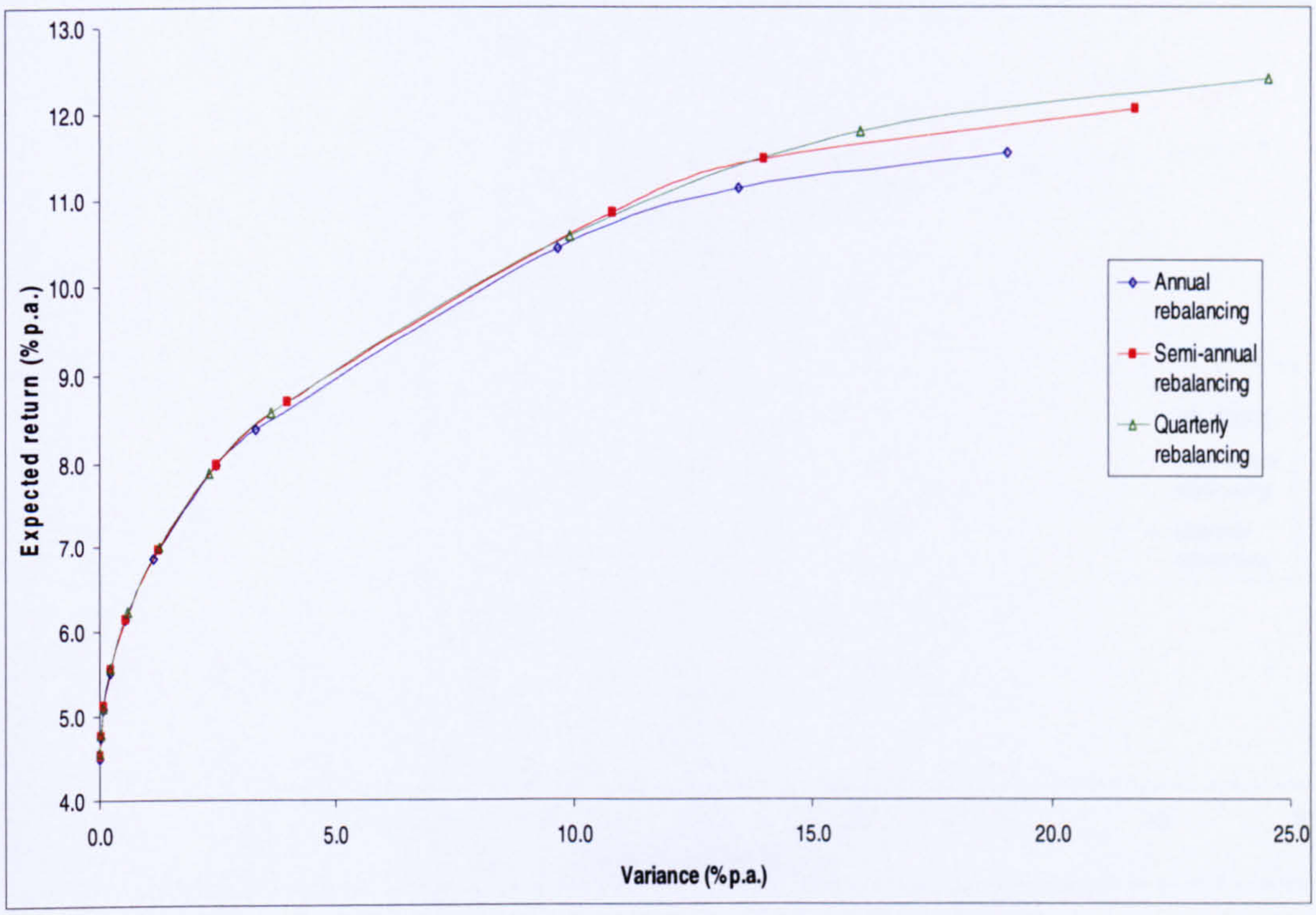
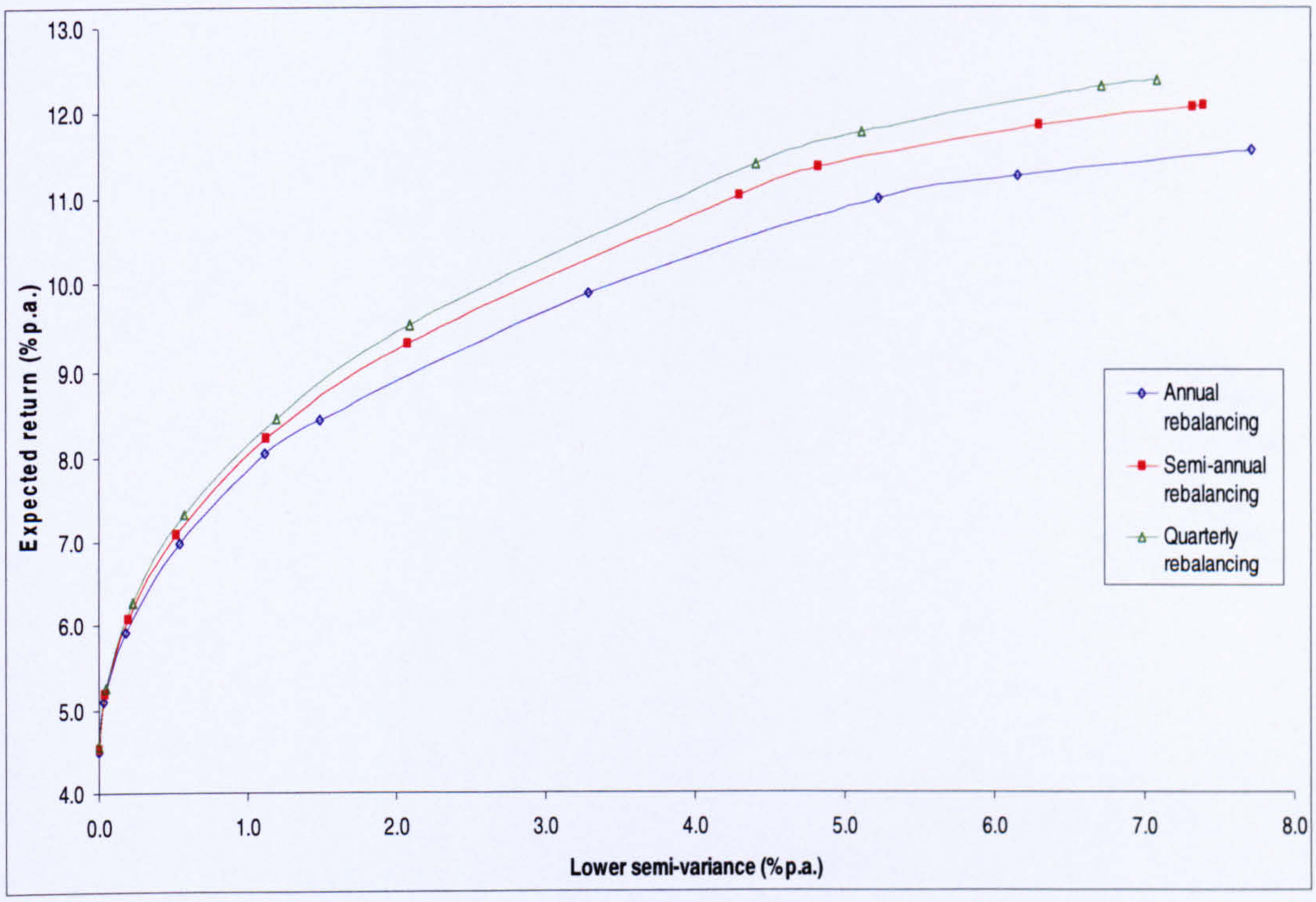


Figure 6.5 shows the efficient frontiers for the CPPI strategy for the three rebalancing frequencies considered when lower semi-variance is the risk measure. It shows that there is an improvement across the entire efficient frontier when increasing the rebalancing frequency both from annual to semi-annual as well as from semi-annual to quarterly. For a lower semi-variance target of above 4% p.a. the higher expected return of the quarterly rebalancing over the annual rebalancing is above 1%p.a.

Figure 6.5: Efficient frontier for a CPPI strategy when lower semi-variance is the risk measure



An even larger improvement in the efficient frontiers form increasing the rebalancing frequency is found when lower partial moment is used as a risk measure as shown in Figure 6.6. For a lower partial moment target of above 0.2%p.a. the extra expected return of the quarterly rebalancing over the annual rebalancing is over 1%p.a.

By comparing the risk measures, I have found that the larger the concentration placed on downside risk the larger the improvement in the efficient frontier from moving to a more frequently rebalancing period. This is intuitive as the greater the concentration placed on downside risk, the more important it is to have frequent rebalancing which reduces the downside risk by allowing the partial sale of equities in a downward market before further losses might occur.

Figure 6.6: Efficient frontier for a CPPI strategy when lower partial moment is the risk measure

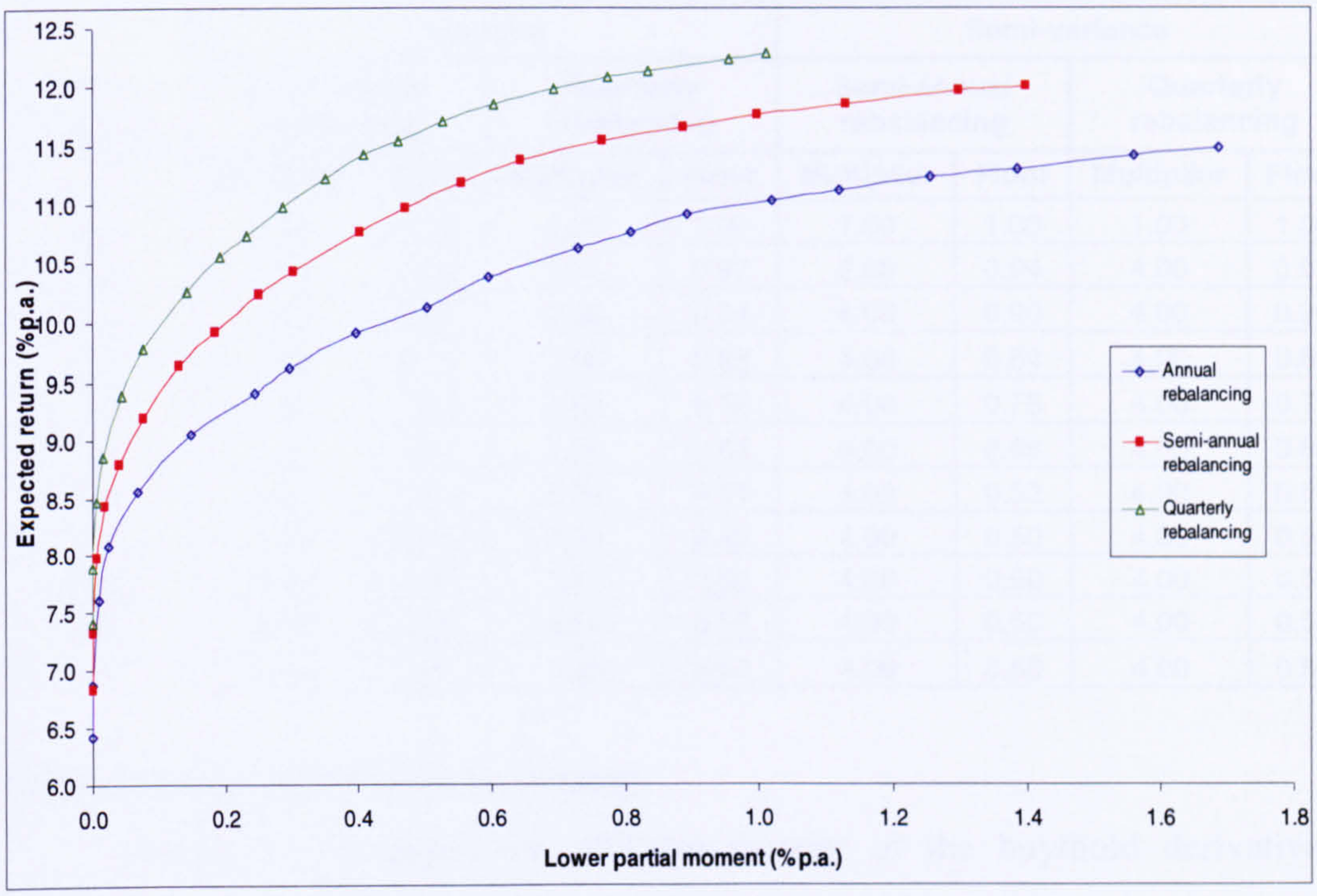


Table 6.2 shows the value of the optimal multiplier and floor for the semi-annual and quarterly rebalancing CPPI strategy at different risk aversion parameters. As the risk aversion parameter increases the optimal value of the multiplier rises from its minimum value of 1 to the maximum allowable value of 4. Correspondingly as the risk aversion parameter increases the optimal value of the floor rises from its maximum value of 1 to the minimum allowable value of 0.5.

Table 6.2: Multiplier and floor values for the CPPI strategy

Risk aversion parameter	Variance				Semi-variance			
	Semi-annual rebalancing		Quarterly rebalancing		Semi-annual rebalancing		Quarterly rebalancing	
	Multiplier	Floor	Multiplier	Floor	Multiplier	Floor	Multiplier	Floor
0.0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
0.1	1.00	0.93	1.71	0.97	2.86	0.94	4.00	0.97
0.2	1.76	0.91	2.02	0.93	4.00	0.90	4.00	0.90
0.3	1.32	0.79	1.66	0.83	4.00	0.84	4.00	0.83
0.4	1.52	0.70	1.66	0.72	4.00	0.75	4.00	0.74
0.5	2.21	0.70	1.88	0.64	4.00	0.66	4.00	0.66
0.6	1.91	0.50	1.88	0.51	4.00	0.52	4.00	0.50
0.7	2.82	0.58	2.30	0.52	4.00	0.50	4.00	0.50
0.8	4.00	0.52	3.52	0.50	4.00	0.50	4.00	0.50
0.9	4.00	0.50	4.00	0.50	4.00	0.50	4.00	0.50
1.0	4.00	0.50	4.00	0.50	4.00	0.50	4.00	0.50

CPPI strategy vs. derivatives strategy

Figure 6.7 compares the efficient frontier of the buy/hold derivative strategy to that with a quarterly rebalancing CPPI strategy, when lower semi-variance is the risk measure.

Figure 6.7: A comparison of the efficient frontiers of a CPPI strategy with a derivative strategy when lower semi-variance is the risk measure

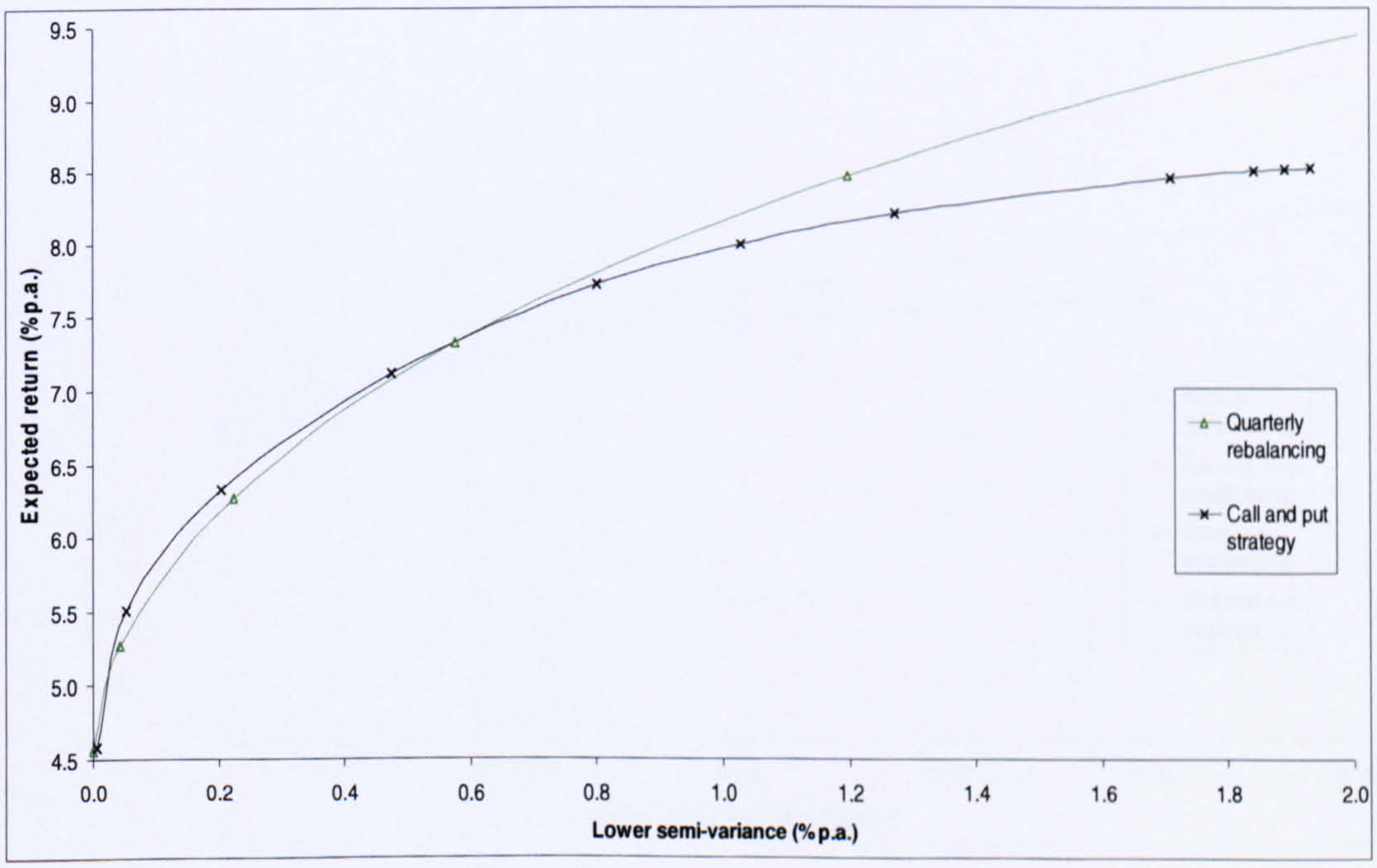
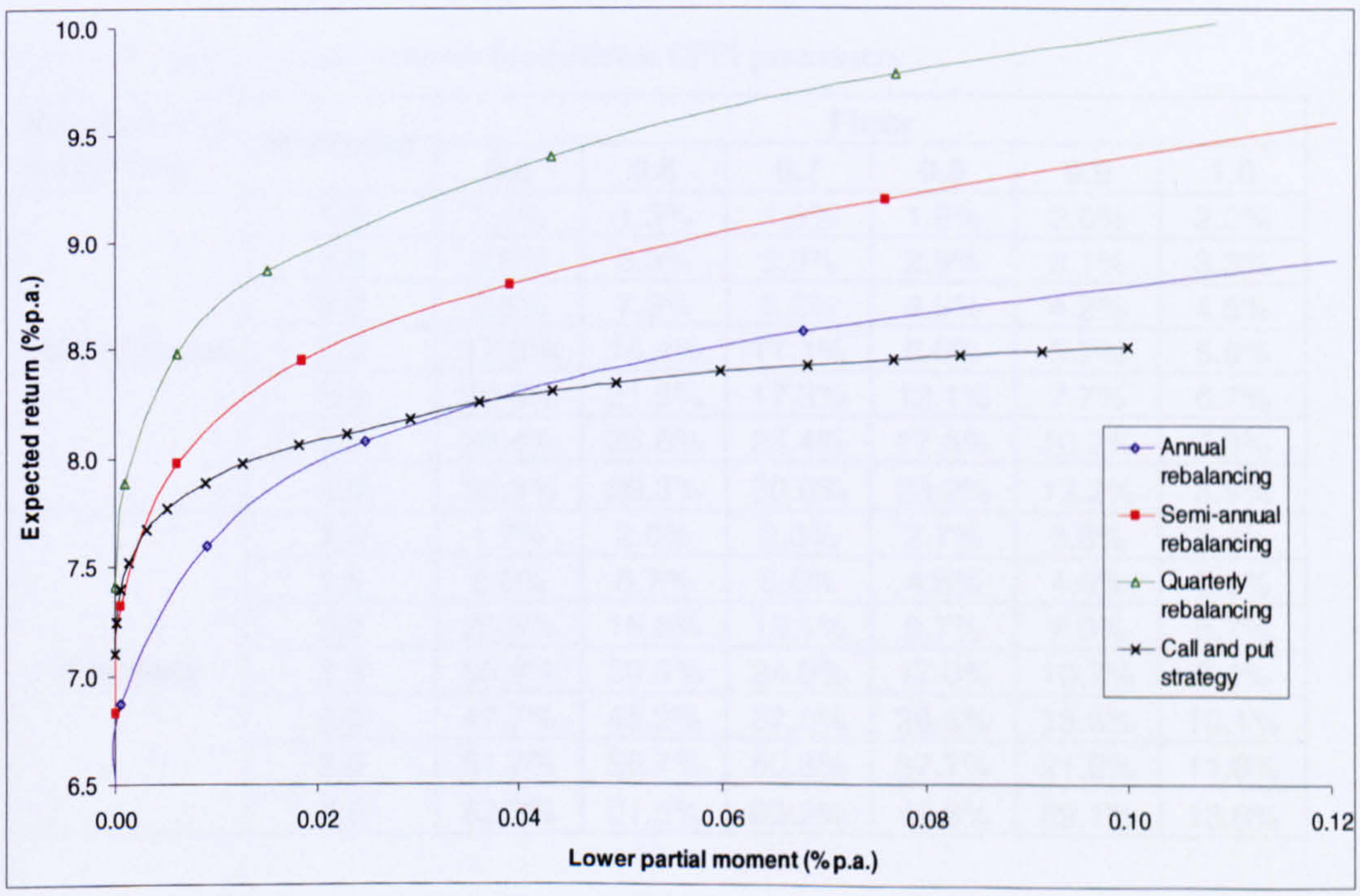


Figure 6.7 shows that for a target lower semi-variance of below 0.6% p.a. the buy/hold derivative strategy produces a slightly higher expected return than the quarterly CPPI strategy is able to achieve. For lower semi-variance targets above 0.6%, the ability to hold above 100% in equities in rising markets results in the efficient frontier of the quarterly CPPI strategy being superior to the buy/hold derivative strategy.

Figure 6.8 compares the efficient frontier of the buy/hold derivative strategy to that with CPPI strategies with different rebalancing periods, when lower partial moment is the risk measure. It shows that the efficient frontiers for the quarterly and semi-annual CPPI strategies are superior to that of the buy/hold derivative strategy. However, when an annual CPPI strategy is considered, the derivative strategy has a superior efficient frontier for target values of the lower partial moment of under 0.03%p.a. with the CPPI strategy performing better for the higher targets of the risk measure.

Figure 6.8: A comparison of the efficient frontiers of a CPPI strategy with a derivative strategy when lower partial moment is the risk measure



By looking at Figures 6.7 and 6.8 together we can conclude that the greater the importance placed on downside risk in the risk measure chosen, the more likely it is for a CPPI strategy to have a superior efficient frontier to that of a buy/hold derivative strategy.

Table 6.3 shows the cumulative equity volume turnover over the rebalancing points, having been averaged across all the simulations, for different CPPI parameter values. These calculations have been based on the returns of the market weighted FTSE All share portfolio and are quoted as a percentage of the initial portfolio wealth. In general the higher the multiplier value, lower the floor value and more frequent the rebalancing frequency the larger will be the equity turnover required to carry out the CPPI strategy. For an institutional investor with say a one way trading cost of 0.05%, the highest equity turnover of 62% would result in a turnover cost of the portfolio of 0.031%. Therefore, the extra expected return achievable from the CPPI strategy should more than offset the additional equity trading costs that the strategy involves.

Table 6.3: Equity volume turnover for different CPPI parameters

Rebalancing frequency	Multiplier	Floor					
		0.5	0.6	0.7	0.8	0.9	1.0
Semi-annual	1.0	1.1%	1.3%	1.6%	1.8%	2.0%	2.2%
	1.5	3.8%	3.3%	3.0%	2.9%	3.1%	3.3%
	2.0	9.6%	7.9%	6.2%	4.9%	4.2%	4.5%
	2.5	17.0%	14.4%	11.1%	8.0%	5.7%	5.6%
	3.0	20.6%	21.5%	17.3%	12.1%	7.7%	6.7%
	3.5	22.4%	25.6%	24.4%	17.3%	10.2%	7.8%
	4.0	36.9%	26.3%	30.0%	23.2%	13.2%	8.9%
Quarterly	1.0	1.7%	2.0%	2.3%	2.7%	3.0%	3.3%
	1.5	8.0%	6.7%	5.6%	4.8%	4.6%	5.0%
	2.0	20.5%	16.8%	13.1%	9.7%	7.0%	6.7%
	2.5	35.9%	30.5%	24.0%	17.0%	10.7%	8.4%
	3.0	47.7%	45.2%	37.0%	26.5%	15.6%	10.1%
	3.5	51.7%	56.7%	50.8%	37.7%	21.8%	11.8%
	4.0	62.3%	61.5%	63.2%	49.8%	29.1%	13.6%

Figure 6.9 shows the improvement in the efficient frontier that is possible when one is able to choose a particular equity allocation between the industrial sectors as compared to the more limited case when the equity allocation must be the same as FTSE All share index (i.e. market portfolio). The improvement is over an extra 1%p.a. expected return for a target variance of above 10%p.a. Similar improvements are found when using the risk measures of lower semi-variance and lower partial moment.

Figure 6.9: A comparison of the efficient frontiers of a CPPI strategy with alternative equity portfolios when variance is the risk measure

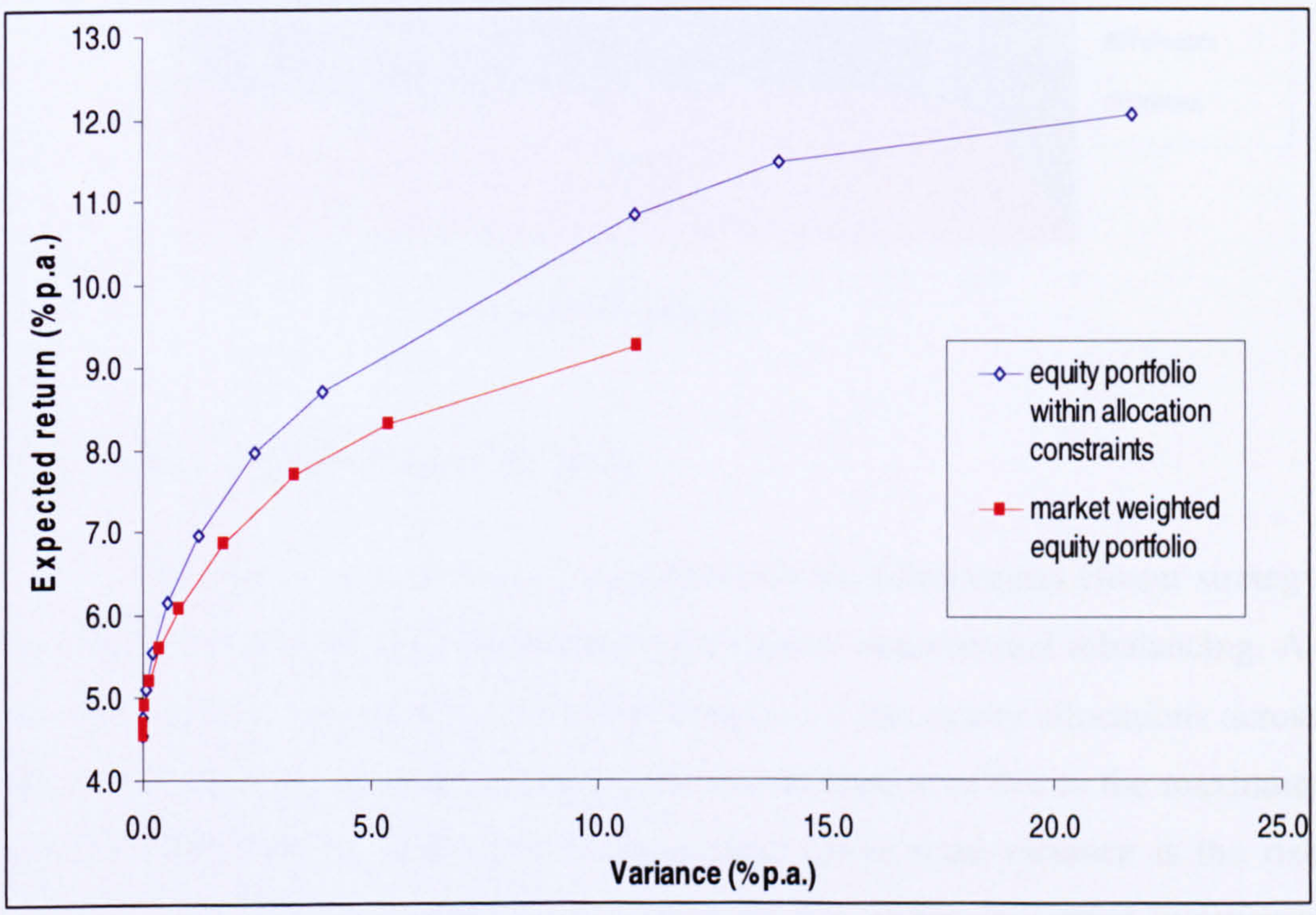
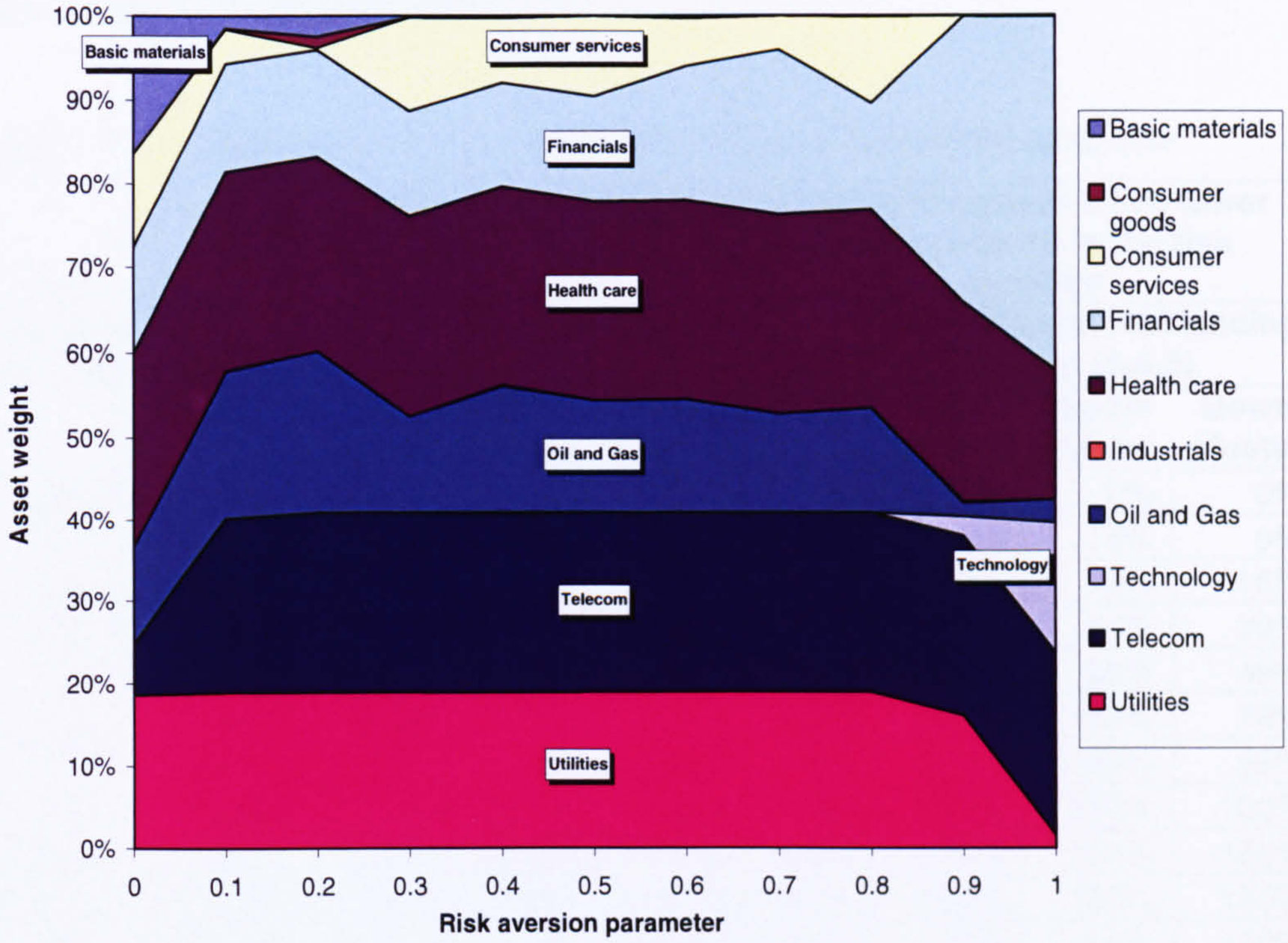


Figure 6.10 shows the equity portfolio split for the CPPI strategy when variance is the risk measure. Compared to the equity market portfolio the equity portfolio found is generally overweight in Heath Care, Technology (only at high risk aversion parameters), Telecom and Utilities while being underweight in the other industrial sectors.

Figure 6.10: Equity portfolio split for the CPPI strategy when variance is the risk measure



6.4.3. Fixed Equity Cluster Strategy

Table 6.4 shows the equity allocations for the fixed equity cluster strategy for different risk aversion parameters in the case of semi-annual rebalancing. As the risk aversion parameter is increased from 0 to 1 the equity allocations across all the clusters rise from the minimum allowed allocation of 0% to the maximum allowed allocation of 150%. In the case when lower semi-variance is the risk measure, the initial equity allocations are larger than in the case when variance is the risk measure. When looking at the change in equity allocations between t = 0 and t = 0.5 you can see that in the lower semi-variance case a more aggressive buy(sale) of equities is made in upward(downward) equity markets. These differences can be explained intuitively, as in the lower semi-variance case, the more aggressive sales in downward equity markets reduces the risk of large downside losses, while the relatively initial high equity allocation and aggressive

increases to equity exposure in upward markets helps to ensure that a high expected return is still possible.

Table 6.4: Fixed equity cluster strategy allocations for the semi-annual rebalancing case

Risk aversion parameter	Equity allocation when variance is the risk measure				Equity allocation when lower semi-variance is the risk measure			
	Initial (t=0)	Semi-annual rebalancing point (t=0.5)			Initial (t=0)	Semi-annual rebalancing point (t=0.5)		
		Up cluster	Middle cluster	Down cluster		Up cluster	Middle cluster	Down cluster
0.0	0%	0%	0%	0%	0%	0%	0%	0%
0.1	7%	10%	8%	8%	19%	42%	15%	9%
0.2	16%	21%	19%	15%	44%	87%	33%	16%
0.3	28%	36%	31%	27%	75%	117%	54%	29%
0.4	45%	58%	49%	44%	98%	150%	96%	46%
0.5	65%	81%	74%	68%	122%	150%	112%	78%
0.6	97%	99%	97%	100%	150%	150%	150%	95%
0.7	114%	141%	117%	105%	150%	150%	150%	100%
0.8	150%	150%	150%	140%	150%	150%	150%	150%
0.9	150%	150%	150%	150%	150%	150%	150%	150%
1.0	150%	150%	150%	150%	150%	150%	150%	150%

Figures 6.11-6.13 show the efficient frontiers for the three rebalancing frequencies for the fixed equity cluster strategy for each of the risk measures considered. The improvement in the efficient frontier achieved when moving from semi-annual to quarterly rebalancing is considerably larger than the improvement seen when moving from annual to semi-annual rebalancing.

The increase in the expected return between annual and quarterly rebalancing is in the order of 0.75%p.a., 1.0%p.a. and 1.5%p.a. in the case of variance, lower semi-variance and lower partial moment respectively. Therefore, the greater the importance placed on downside risk the larger is the benefit found in having more frequent rebalancing.

Figure 6.11: Efficient frontier for the fixed equity cluster strategy when variance is the risk measure

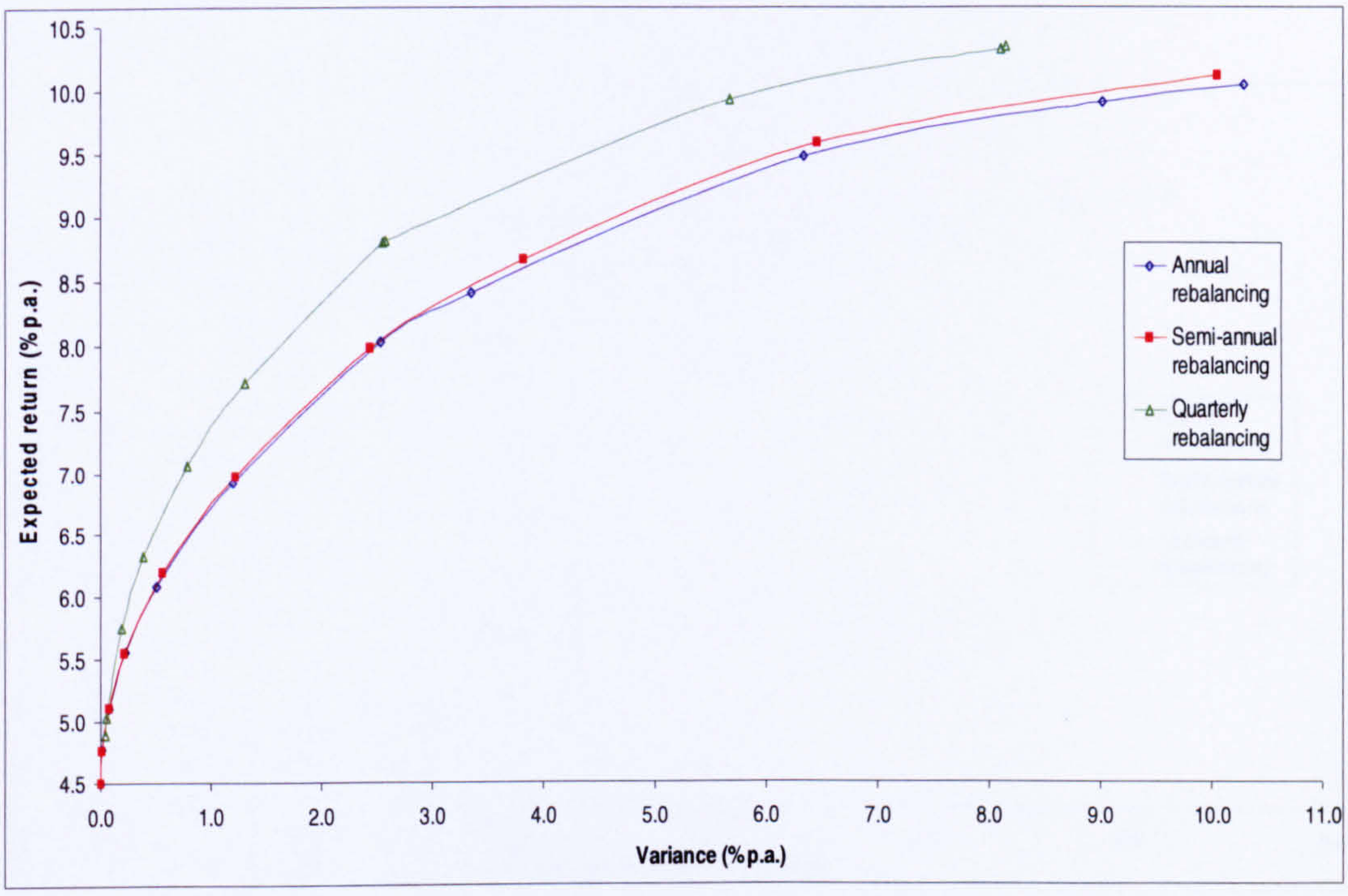


Figure 6.12: Efficient frontier for the fixed equity cluster strategy when lower semi-variance is the risk measure

Figure 6.13: Efficient frontier for the fixed equity cluster strategy when lower partial moment is the risk measure

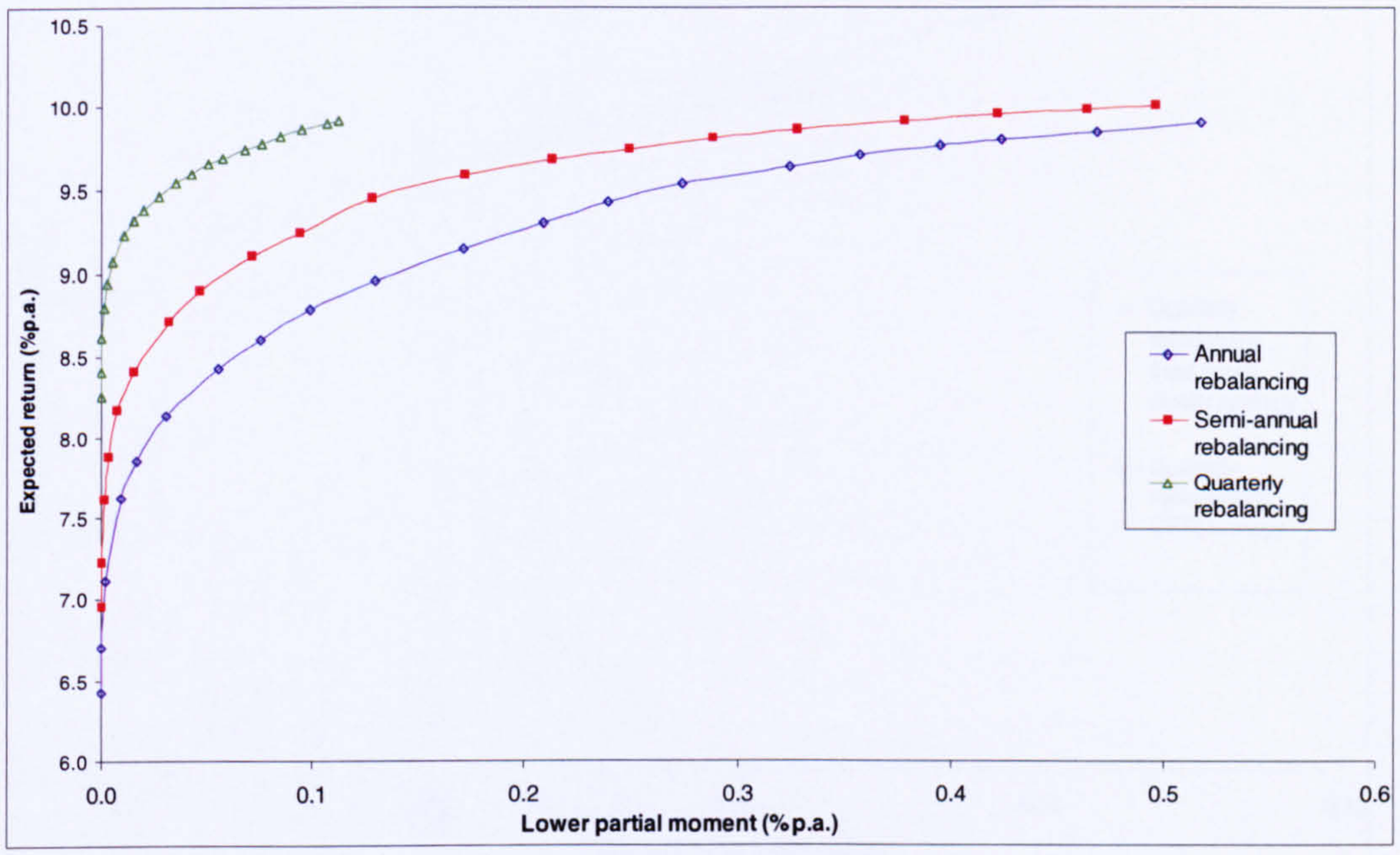
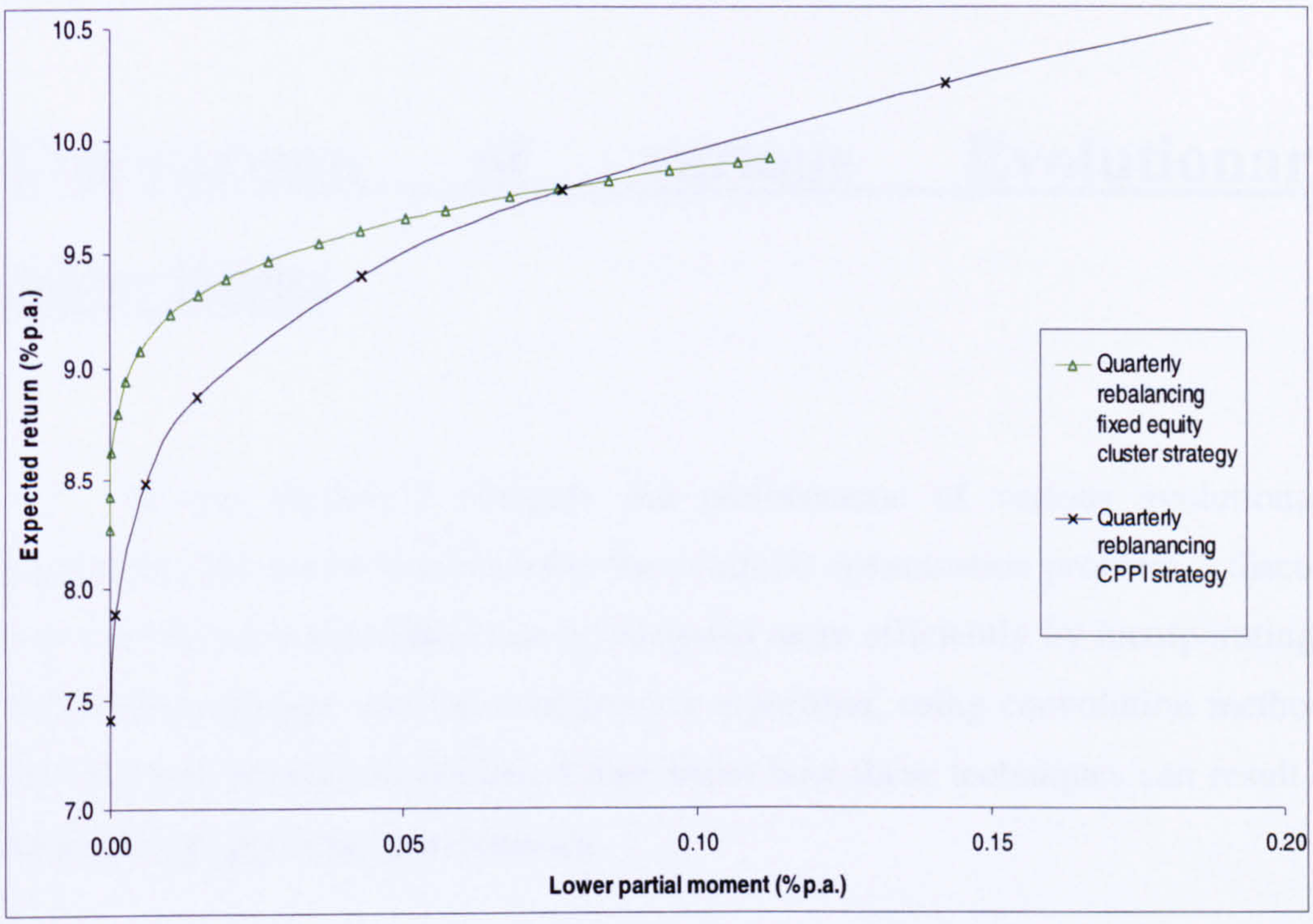


Figure 6.13 shows a comparison of the efficient frontiers of a fixed equity cluster strategy and CPPI strategy for quarterly rebalancing when lower partial moment is used as the risk measure. It shows that the greater freedom allowed in the fixed equity cluster strategy, allows an extra expected return of up to 0.75%p.a. compared to the CPPI strategy when considering a lower partial moment target of under 0.08. However, the ability to invest up to 200% in equity markets in the CPPI strategy, while only 150% in the fixed equity cluster strategy, means that the CPPI strategy is able to achieve a better risk/return trade-off for lower partial moment targets of above 0.08. This same effect was also found when the risk measure of either variance or lower semi-variance was used instead of lower partial moment.

Figure 6.13: A comparison of the efficient frontiers of a fixed equity cluster strategy and CPPI strategy for quarterly rebalancing when lower partial moment is the risk measure



6.5. Conclusions

In this chapter I have demonstrated how the use of multi-period strategies can significantly improve the efficient frontier. I have shown how through the use of a CPPI strategy or fixed equity cluster strategy an investor is able to improve on the achievable risk/return tradeoffs over certain parts of their efficient frontier. I have also shown that when placing a higher importance on downside risk within the risk measure the benefits of more frequent rebalancing is seen to increase. I have also found that a dynamic investment strategy is more likely to outperform a strategy that includes the use of derivatives when a greater importance is placed on downside risk.

Chapter 7

Comparison of various Evolutionary Algorithms

In this chapter I compare the performance of various evolutionary algorithms that can be used to solve the portfolio optimization problem. I discuss how evolutionary algorithms can be designed more efficiently by incorporating a local search process into the evolutionary algorithm, using coevolution methods and multiple population models. I then show how these techniques can result in an improved algorithm performance.

7.1. Memetic Algorithms

Standard genetic algorithms do not incorporate any problem-specific knowledge or local search technique. In contrast, memetic algorithms are evolutionary algorithms, described by Moscato (2002), that use local search or problem specific information. Significant improvements in the efficiency of the algorithms may be possible by using evolutionary algorithms to identify promising areas of search space (exploration) and using local search to aid the evolutionary algorithm in finding the local optima within these promising areas (exploitation). Memetic algorithms differ from standard genetic algorithms in that new solutions are created from highly evolved parents from the previous generation. The independent evolution of the parent solution may be carried out by a local search algorithm.

Use of problem specific information

Memetic algorithms can also benefit from using problem specific information. This may either be using intelligent non random initialization or using intelligent variation operators. Surry & Radcliffe (1996) found that the use of a small number of seeded individuals improved the search process by increasing the average performance and decreasing the variance of performance. However, nonrandom initialization was also less likely to produce an exceptional solution due to the greater limitation on the areas of search space covered. Intelligent crossover and mutation operators may be used to bias the search into areas which are thought to be more promising.

Lamarckian and Baldwinian evolution

A key issue of memetic algorithms is how the global algorithm uses the information learned during the local search process. In Lamarckian evolution, traits that are acquired during an individual's lifetime can be passed from parents to their offspring. In Baldwinian evolution, traits that initially have to be learned by an individual are over time replaced by future generations that genetically have those traits which no longer need to be learned.

When using evolutionary algorithms, knowledge learned through local search can result in solutions with improved genes and fitness values. In Lamarckian evolution both the improved genes and fitness value are passed on to offspring. In Baldwinian evolution, individuals that have learned, thus improving their fitness value, are more likely to have offspring. However, it is the original rather than improved genes that will be passed on to offspring.

Bobo & Goldberg (1997) showed that the combination of a local search method with a global genetic method can lead to an algorithm that is superior to each method in isolation. They also found that the efficiency of the hybrid algorithm depends on the balance between the local and global search. Espinoza, Minsker & Goldberg (2001) showed how an adaptive hybrid algorithm can result in an increased convergence speed.

Hart (1994) investigated the tradeoff between cost and value of the local search. If local search was performed on every individual in the population, it would be very time-consuming, leaving too little time for the global algorithm to explore the search space. The paper found that it was more effective to only use a local search method on a proportion of individuals rather than the entire population.

Whitley, Gordon & Mathias (1994) compared two approaches of hybrid genetic search using Lamarckian and Baldwinian evolution. Both hybrid genetic algorithms were found to give better and faster solutions than a standard algorithm that did not utilize a local search technique. They found that when one needs to obtain fast results the Lamarckian is consistently better. However, they found that Baldwinian evolution may lead to better global optima over an extended period of time as Lamarckian evolution can disrupt the schema processing leading to premature convergence. Elmihoub, Hopgood, Nolle & Battersby (2004) found that using a low probability of local search (i.e. 0.1) the quality of the solution is insignificantly affected by whether a Lamarckian or Baldwinian learning strategy is used. This is because the small probability of local search does not disrupt the schema processing of the global search process. The advantage of the Lamarckian strategy is due to having a faster convergence speed than either the Baldwinian learning strategy or pure global strategy alone.

The two parameters that control the balance between the local and global search are firstly, the probability of local search for each individual and secondly, the number of neighbours searched. An increase to either parameter increases both the computation time and emphasis placed on the local search. Therefore, in order to maintain the balance between the local and global search, an increase in the probability of local search for each individual needs to be offset by a reduction in the number of neighbours searched. As the problem of premature convergence is often increased when using memetic algorithms one also needs to ensure that there is sufficient diversity in the population, e.g. by limiting the frequency and degree to which local search is performed.

Ishibuchi, Yoshida & Murata (2002) describe a hybrid multi-objective optimization algorithm that incorporates local search into the Strength Pareto Evolutionary Algorithm (SPEA). They showed that improved performance was achieved by the hybrid algorithm when only a random number of the best solutions were selected for the local search and if the search was terminated before all neighbours of the current solution were examined. They conclude that the hybrid algorithm outperforms the non-hybrid algorithm when a good balance is achieved between the genetic search and local search due to appropriate parameter specifications.

The pseudocode for a memetic algorithm that has phases of local improvement for the whole population is detailed below.

```
Function MemeticAlgorithm ()
    Initialize Population
    Evaluate each solution
    Repeat Until (termination condition is satisfied)
        Select parents
        Recombine parents to produce offspring
        Mutate the offspring
        Evaluate the new population
        Improve the population via a local search
        Select individuals for the next generation
    End Repeat
    Return Population
End Function
```

7.2. Coevolution

In coevolution the evolution of one species affects the evolution of another species and vice versa. Coevolution can either be cooperative (inspired by symbiosis where species coevolve in a mutually beneficial way) or competitive (inspired by predator/prey relationship where one species has a negative effect on the other, often referred to as an arms race). In standard evolutionary algorithms individuals are evaluated using an absolute fitness measure, based on the overall objective function. In coevolution models an

individual is evaluated using a relative fitness measure that depends on the evolutionary state of the other subpopulations.

Cooperative coevolution works by dividing a problem solution into a number of solution component parts with each component part evolving in a different subpopulation. An individual solution component is then evaluated according to the fitness of whole solutions formed by encounters with selected solution component parts from the other subpopulations. Various methods can be used to select the solution component parts from the other subpopulations e.g. random, stochastic fitness-based or the best individual. The advantage of this divide and conquer approach is that by partitioning the problem into a number of subpopulations, each subpopulation then has a significantly reduced search space to cover. However, it requires that the user is able to find an appropriate partition to the main problem.

Potter & De Jong (1994) use a cooperative generational genetic algorithm approach with each species evolving in a different subpopulation. Each species then takes it in turn to undergo the process of selection, recombination and mutation. Evaluation for each individual is performed twice, firstly with the fittest individual from each other subpopulation and secondly with a randomly selected individual from each other's subpopulation. They found that this approach minimized the risk of premature convergence.

For static optimization problems, standard cooperative coevolutionary algorithms were not always directed towards optimal collaboration, i.e. the composite solution components that result in the optimal problem solution. Cooperative coevolutionary algorithms were found to have a behaviour called relative overgeneralization. This meant that for certain problems instead of finding component parts which make up the optimal problem solution the algorithm tended to find a problem solution that was robust under a change of one of the solution component parts. Relative overgeneralization is desirable

(undesirable) if a robust solution is preferred (not preferred) to a globally optimal solution.

Bucci & Pollack (2005) found that by modifying cooperative coevolutionary algorithms to include Pareto selection they were able to reliably find the global optimal solution. In their example the function they want to maximize has the form f(x,y) with the algorithm having individuals in two subpopulations $x \in X$ and $y \in Y$. They compared the following three methods, which all used the selective pressure parameter p (set at 0.75) and an elitist policy where the fittest individual from the previous generation was always added to the next generation.

1. A stochastic binary tournament (with a probability p that the fitter individual is chosen) between two individuals x_1 and x_2, where the fitness of x_i is given by
   $\max(f(x_i, y_{best}), f(x_i, y_{random}))$.
2. A stochastic binary tournament (with a probability p that the fitter individual is chosen) between two individuals x_1 and x_2. The fitness of x_i is given by $\max(f(x_i, y_1), f(x_i, y_2),\ldots, f(x_i, y_N))$, where there are N individuals in subpopulation Y.
3. Two solutions x_1 and x_2 are compared for Pareto dominance. The solution x_1 is said to dominate x_2 if $f(x_1, y_i) \geq f(x_2, y_i)$ for $i = 1, 2,\ldots, N$ and
   $f(x_i, y_i) > f(x_i, y_i)$ for at least one value of $i = 1, 2,\ldots, N$. If one individual dominates the other than the dominant individual is selected for the next generation with probability p. If neither individual is dominant both are added to the next generation.

The first method always finds individuals at or near the local optima, the second method tends to find individuals near the global optima while the third method always finds individuals at or near the global optimum. Method 3 performed significantly better than method 2 for problems were the spread between the global and local optima is relatively small.

In competitive coevolution individuals compete against each other in order to increase their fitness at the other's expense, where the individuals can either belong to the same or different species. Angeline (2000) says that the main advantage of this method is that it is self-scaling. Weak solutions may survive near the start of the run as their competitors are not strong. As the run progresses, the average fitness level in the population will increase, resulting in an increase in the difficulty of the fitness function.

7.3. Multiobjective Optimization

7.3.1. Cooperative Coevolution Approach

The following papers all use a cooperative algorithm to solve a multiobjective problem where a number of subpopulations, representing different parts of a problem, coevolve in order to find the Pareto frontier.

In Keerativuttitumrong, Chaiyaratana & Varavithya (2002) the evolution of each subpopulation is carried out using Fonseca & Fleming's (1993) MOGA. Each solution is evaluated twice, firstly with the best ranked individuals from each subpopulation and secondly with a randomly selected individual i.e. following Potter & De Jong (1995). Each subpopulation was then evolved according to a dominance rank of solution components within the same subpopulation only, which has the drawback of limiting the assessment of solution components.

Tan, Chew, Lee & Yang (2003) use an external archive to store the nondominated solutions found to date, dynamic sharing and an extending operator. The archive is updated with any new solution that is not dominated by any existing member of the archive, with existing members of the archive that are dominated by this new solution being removed. Once the maximum size of the archive is reached, dynamic sharing is used to maintain diversity of the individuals in the archive. Each individual from each subpopulation is randomly

assigned individuals from the other subpopulations in order to form a complete solution. Each subpopulation is then evolved using Pareto dominance ranking against the archive set. An extending operator is used to guide the search towards the areas of the Pareto frontier which are underrepresented in the archive by copying the archived individual with the lowest niche count into the evolving subpopulations.

Maneeratana, Boonlong & Chaiyaratana (2004) showed that for a variety of test problems and different types of multiobjective evolutionary algorithms the integration of cooperative coevolutionary algorithms into multiobjective evolutionary algorithms outperformed the original multiobjective evolutionary algorithms. They use an archive to store the non-dominated solutions found to date with a crowding distance measure used to maintain diversity and limit the maximum number of archived solutions. For each subpopulation each individual is randomly combined with the remaining solution components from a randomly selected archived member in order to form a complete solution. Once evaluation of each subpopulation is performed a multi-objective algorithm is used to evolve each subpopulation. A number of elite non-duplicated non-dominated individuals from each subpopulation are also passed unaltered to the next generation.

Iorio & Li (2004) introduce a new method for collaborator selection and evolve each subpopulation using Deb, Pratap, Agarwal & Meyarivan's (2002) NSGA2. In the first generation, an individual is evaluated based on a random collaboration with solution components from the other subpopulation. The non-dominated sorting algorithm is then applied over each subpopulation. For each subpopulation, an offspring subpopulation is formed and evaluated by collaborations with randomly selected non-dominated solution components of the other subpopulations. Non-dominating sorting along with a crowding distance method and elitist operator are applied over each combined subpopulation of parents and offspring in order to determine the new parent subpopulation.

7.3.2. Island Model Approach

A detailed overview of parallel multiobjective evolutionary algorithms, including alternative migration, replacement and niching schemes can be found in Van Veldhuizen, Zydallis & Lamont (2003).

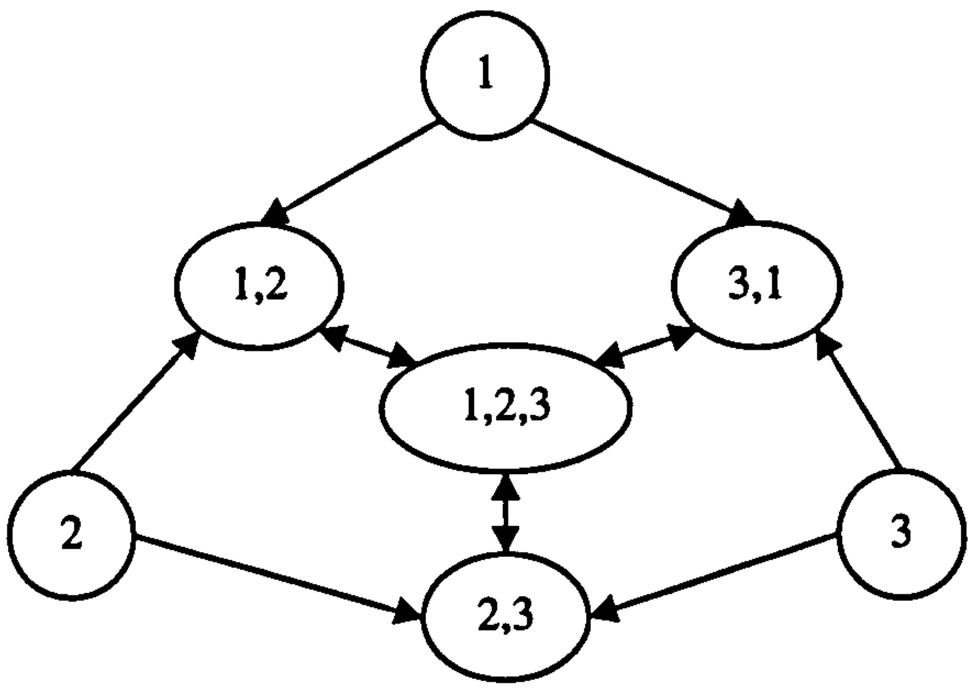
The simplest island model used to solve a single objective optimization problem, involves several homogeneous subpopulations each having the same representation and algorithm parameters. Hiroyasu, Miki & Watanabe (1999) explain that the reason this island model is able to outperform its equivalent single population model at finding the global optima is due to the combination of fast convergence in each subpopulation, due to its reduced size, and diversity between the different subpopulations. However, they explain that a homogeneous island model may not be as effective at solving multi-objective optimization problems due to the inadequate subpopulation size to perform local search effectively and the inefficiency of every island searching the same search space. This is supported by the literature on using island models for multiobjective optimization problems which typically use subpopulations with different search domains.

Two similar island models that use different search domains for each subpopulation are Hiroyasu, Miki & Watanabe's (1999) Divided Range Genetic Algorithm (DRGA) and De Toro, Ortega, Fernandez & Diaz's (2002) Parallel Single Front Genetic Algorithm (PSFGA). An initial population is randomly produced and the non-dominated solutions selected. These non-dominated individuals are sorted according to the values of the objective functions and then partitioned into a number of subpopulations. A sequential multiobjective genetic algorithm is then performed on each subpopulation for a specified number of generations. The subpopulation are then joined into one large population, compared for Pareto dominance and then re-divided according to the values of the objective functions. Both models were found to outperform a single population model.

A similar approach is used by Streichert, Ulmer & Zell (2005) who use a clustering scheme for a parallel island multiobjective optimization problem. Initially, a number of subpopulations simultaneous search for the Pareto frontier using the NSGA2 algorithm. After a specified number of generations all the subpopulations are joined, clustered and redistributed into the subpopulations. Clustering can either be done in the search space or the objective space.

Xiao & Armstrong (2003) use a specialized island model in order to produce a set of diverse non-dominated solutions to a multiobjective optimization problem. Each island is responsible for optimizing a subset of the original problem's objective function. The results produced suggested that this model could effectively find the non-dominated solutions. They regarded this method as a generalized version of Schaffer's (1985) VEGA model, where individuals in each subpopulation are evaluated using only one objective function. The difference between the models is that in the VEGA model the variation operators are performed on the whole population, while in the specialized island model, the variation operators were only performed on each subpopulation. They considered a variety of alternative numbers of subpopulations and communication topologies for solving a three objective optimization problem. Their most complex model structure contained seven subpopulations with a communication topology shown in Figure 7.1.

Figure 7.1: Specialized island model diagram



Each subpopulation is represented by a node, with the number(s) inside each node showing the objective(s) for that subpopulation. A non-dominated sorting approach is used for subpopulations with more than one objective function.

There are two examples in the literature of island models that have been used to solve portfolio optimization problems. Firstly, Loraschi & Tettamanzi (1995) and Loraschi, Tettamanzi, Tomassini & Verda (1995) used an island stepping stone model where each island represents a different risk aversion parameter. Secondly, Fieldsend, Matatko & Peng (2004) used a parallel island-migration model where each island consisted of solutions with a different cardinality level.

7.3.3. Measuring Convergence

For a single objective optimization problem, the level of convergence can be assessed by looking at the improvement in the average fitness level as the generation number increases. Tan, Lee & Khor (1999) give a possible convergence measurement for multiobjective optimization problems that is based on the idea of population domination and a progress ratio. The progress ratio at a generation is defined as the ratio of the number of nondominated individuals at that generation that dominate the nondominated individuals at the previous generation. The progress ratio would typically start close to one (high probability of improving the Pareto frontier) and decrease asymptotically towards a value close to zero (low probability of improving the Pareto frontier). The termination criteria may be based on the smoothed progress ratio, calculated as the average progress ratio over the last specified number of generations, falling below a certain value.

7.3.4. Performance Metrics

The comparison of different multi-objective evolutionary algorithms is more complex than that for single-objective algorithms due to the multiple optimization aims. These aims were identified in Zitzler, Deb & Thiele (2000) as

- the distance of the non-dominated set found to the Pareto-optimal front should be minimized. As my problem specification does not enable me to specify a known Pareto-optimal front I will be unable to test for this.
- the solutions over the non-dominated set should be evenly distributed in order to ensure diversity of the solutions found along the efficient front.
- the extent of the non-dominated front found should be maximized, so that the non-dominated set dominates as many alternative solutions as possible.

Quantitative performance metrics that can be used in order to compare the quality of the obtained non-dominated front are described below.

1. The size of space covered (S metric) measures the area covered by the obtained non-dominated set as detailed in Zitzler & Thiele (1998). A high value of S is more desirable as a larger area under the efficient frontier indicates that more of the alternative solutions will be dominated by the obtained non-dominated set.
2. The spacing metric Δ can be used to measure how evenly the points in the found non dominated set are distributed over the objective space. Deb, Agrawal, Pratab & Meyarivan (2000) specify the spacing metric as

$$\Delta = \frac{\sum_{i=1}^{N-1} \left| d_i - \bar{d} \right|}{(N-1)\bar{d}} \tag{7.1}$$

were d_i represents the Euclidean distance between the consecutive solutions in the found non-dominated set, $\bar{d}$ is the average of these distances and N is the number of solutions in the non-dominated set. A low value of Δ is more desirable as it indicates better diversity along the non-dominated set.

3. The C metric can be used to compare two obtained solution sets by calculating the fraction of each solution set that is covered by the other solution set. It is defined by Zitzler, Deb & Thiele (2000) as

$$C(X', X'') = \frac{\left|\{a'' \in X''; \exists a' \in X' : a' \geq a''\}\right|}{\left|X''\right|}, \tag{7.2}$$

where X' and X'' are two obtained fronts. The function C then maps the ordered pair (X', X'') to the interval [0, 1].

When all points in X'' are dominated by or equal to points in X' then C(X',X'') = 1. When none of the points in X'' are covered by the set X' then C(X',X'') = 0. It is necessary to calculate both C(X',X'') and C(X'',X') as can be seen by the case where X' dominates X'' which gives C(X',X'') = 1 and C(X'',X')=0.

4. Zitzler, Deb & Thiele (2000) also use an extent metric which calculates the extent to which the front spreads out as the Euclidean distance between the two extreme points of the front.

In the case of multiple single objective optimizations the value of the objective function at each risk aversion point can be directly compared between the different evolutionary algorithms in order to determine the quality of the algorithm.

7.4. Alternative Evolutionary Algorithm Designs

In this section I detail the three main types of algorithm I am considering, type SSRAM algorithm, type NSGA2 algorithm and the hybrid SSRAM and NSGA2 algorithm.

SSRAM type algorithms

(i) The fixed migration stepping stone risk aversion model SSRAM(FM) is based on the portfolio optimization model of Loraschi & Tettamanzi (1995) and Loraschi, Tettamanzi, Tomassini & Verda (1995).

(ii) In order to investigate the benefits of allowing migration I have also tested a no migration model SSRAM(NM).

(iii) I also introduce a novel model feature of competitive migration in the SSRAM-CC algorithm which allows the actual migration rate to adaptively change through the algorithm run. In this algorithm, migration only occurs to a neighbouring island if its new fitness value (i.e. according to the objective function of the new island) is above the pre-migration average fitness value of the new subpopulation. The reason for introducing this adaptive migration is due to the reason that as the run progresses and each island begins to reach its optimum point there should be less benefit from migration due to the different objective functions (and therefore different optimum solution) of each island.

(iv) I then extend this new competitive coevolution algorithm to the SSRAM-CCLS algorithm which also includes a local search Lamarckian evolution process.

Table 7.1 details the feature and parameters of type SSRAM algorithms.

Table 7.1: Features and parameters of type SSRAM algorithms

General features and parameters	
Elitist	Uses elitism to ensure that in each island the fittest individual is passed on to the next generation
Number of subpopulations	11 subpopulations having a risk aversion parameter spread uniformly over [0, 1].
Communication topology	Arranged as a stepping stone model i.e. each intermediate island is connected to two subpopulations while the two end subpopulations are only connected to one subpopulation.
Size of each subpopulation	100

Migration features and paramters	
Migration strategies	a) No migration: SSRAM(NM)
	b) Static migration: SSRAM(FM)
	c) Competitive coevolution: SSRAM-CC and SSRAM-CCLS
Epoch length	10 generations
Migrant selection	Individuals are simultaneously randomly selected for copying, without replacement, from each subpopulation. The 2 end subpopulations select 20 individuals each and the remaining 9 subpopulations select 40 individuals each.
Number of individuals copied	For static migration, each subpopulation adds 20 of the copied individuals to each of its adjacent subpopulation, i.e. 20 individuals are added to the 2 end subpopulations and 40 individuals are added to each of the remaining 9 subpopulations.
	The competitive coevolution strategy differs from the static migration in that each individual is only copied to its adjacent subpopulation if its fitness value in the new subpopulation is above the pre-migration average fitness value of the new subpopulation.
Migrant replacement	Random replacement, that excludes deleting the elite individual in each subpopulation, is used to maintain the fixed size of each subpopulation.
Local search Lamarckian features and parameters	
Number of individuals where local search is performed	The elite individual and three more randomly selected individuals
Number of neighbours searched for each individual chosen	3
Move strategy	Each individual chosen is replaced by its best neighbour, using the best admissible move strategy, if it has a higher fitness value (i.e. Lamarckian evolution).
Neighbourhood	Based on the Euclidean distance between the asset allocation weights

NSGA2 type algorithms

(i) Nondominated Sorting Genetic Algorithm (NSGA2) of Deb, Pratap, Agarwal & Meyarivan (2002) which has been used in the portfolio optimization problem of Mukerjee, Biswas, Deb & Mathur (2002).

(ii) A homogeneous multiple population migration model (NSGA2-MP), where each island evolves according to a non-dominated sorting genetic algorithm An example four homogeneous island model is shown in Figure 7.2 and Table 7.2 details the model features and parameters.

Figure 7.2: NSGA2-MP algorithm diagram

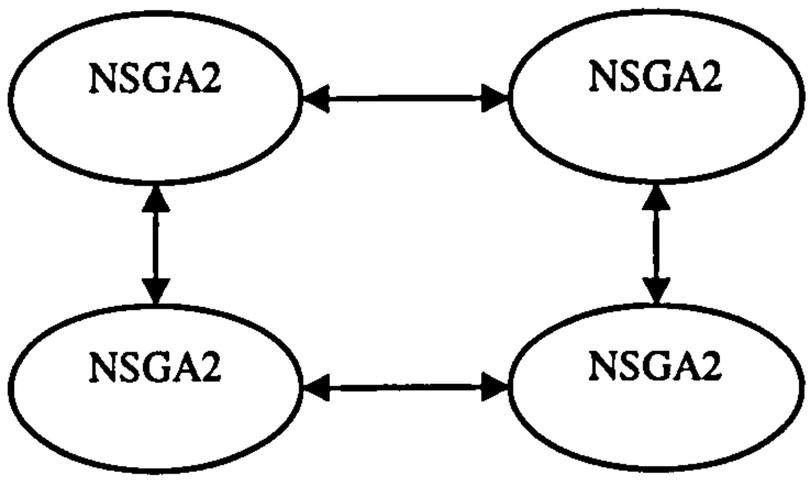


Table 7.2: NSGA2-MP model features and parameters

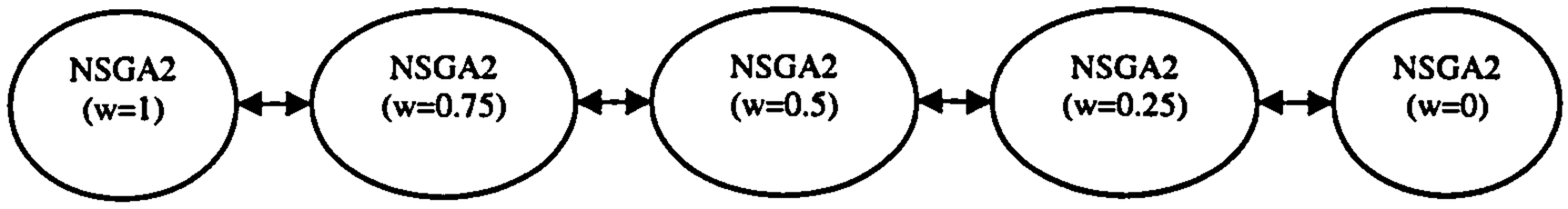
Number and size of subpopulations	Either two or four subpopulations used each of size 100
Communication topology	Arranged as a circular model, i.e. each island is connected to two adjacent islands
Epoch length	10 generations
Migration selection	40 individuals are simultaneously randomly selected for copying, without replacement, from each subpopulation. 20 of the individuals selected are then copied to each of the two adjacent subpopulations.
Migration replacement	Uses Pareto dominance and crowding distance measure to maintain the fixed population size.

(iii) Cooperative coevolution nondominated Sorting Genetic Algorithm (CCNSGA2) which involves dividing the problem into more than one subproblem. This algorithm is based on Iorio & Li (2004) cooperative coevolution multiobjective model that utilises the NSGA2 algorithm. To my knowledge a cooperative coevolution approach has not yet been used to solve a portfolio optimization model. I will divide my problem into two roughly equal subproblems by addressing two key parts of the problem. The first part of the solution will be the asset allocation of the equity asset class,

each individual call option, each individual put option and cash. The second part of the solution is the asset allocation within the equity portfolio, i.e. the relative asset allocation to each of the equity industrial sectors. This division of the overall base case problem means that the first subproblem part has 7 solution components and the second subproblem part has 10 solution components.

(iv) A novel stepping stone island Nondominated Sorting Genetic Algorithm model (SSNSGA2) with each island varying according to the allowed risk measure. This approach may be appropriate if one expects similar asset allocations between the different problem design setups so that parallelism improves the efficiency of the algorithm. This is similar reasoning to Fieldsend, Matatko & Peng (2004) approach who used a parallel island-migration model where each island consisted of solutions with a different cardinality level. Figure 7.3 shows the diagram for the 5-island case with an equally spaced risk measure.

Figure 7.3: SSNSGA2 algorithm diagram



Hybrid SSRAM and NSGA2 algorithm

A novel approach which is a hybrid of the stepping stone risk aversion model and the island nondominated sorting genetic algorithm (SSRAM-CC—NSGA2). The communication between the islands is shown in Figure 7.4 for the case where the SSRAM model has 6 islands. This can be seen as a specific case of the generalised version of Xiao & Armstrong's (2003) specialised island model, where each island is responsible for optimizing a subset of the original problem's objective function. The SSRAM-CC—NSGA2 model can also be seen as a variation of Hiroyasu, Miki & Watanabe's (1999) and De

Toro, Ortega, Fernandez & Diaz's (2002) models. The similarities between these models are that they have both a global part that focuses on the evolution of the whole efficient frontier and a local part that focuses on the evolution of each specific portion of the efficient frontier. However, the difference is that in the SSRAM-CC—NSGA2 model these two evolutionary parts work simultaneous with migration forming communication between the parts, whereas in the other two models the evolution is periodically switched from working on the whole efficient frontier to each part separately.

Figure 7.4: SSRAM-CC—NSGA2 algorithm

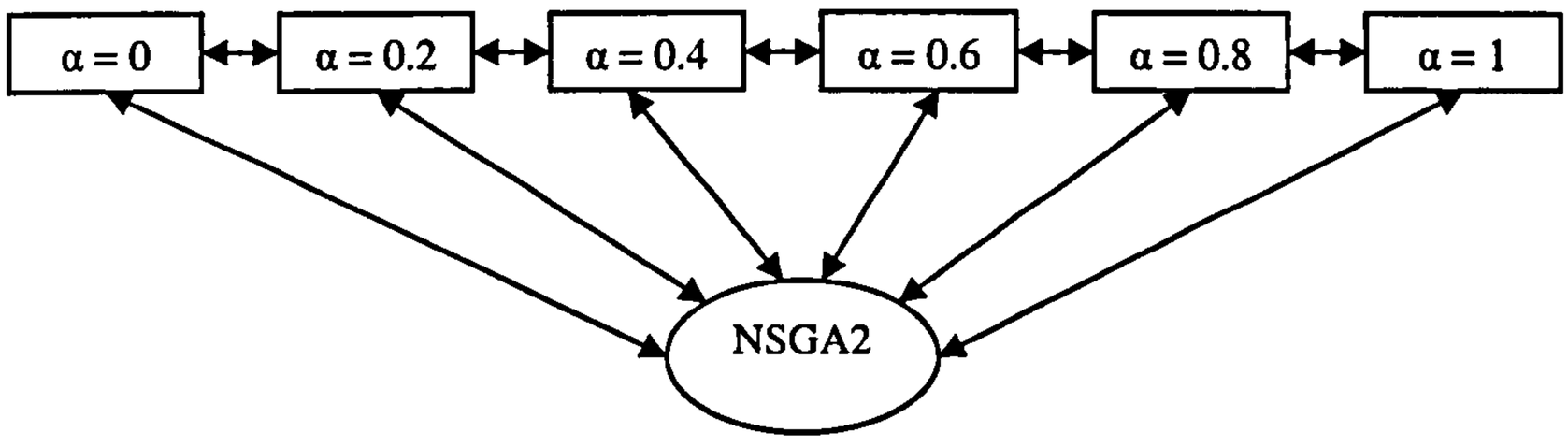


Table 7.3: SSRAM-CC—NSGA2 model features and parameters

Epoch length	5 generations
Migration strategy from NSGA to SSRAM	For each SSRAM subpopulation the individual in the NSGA population with the highest fitness value, according to that subpopulations specific objective measure, is found. This individual is then copied to that SSRAM subpopulation with random replacement, that excludes deleting the elite individual in that subpopulation, used to maintain the fixed subpopulation size.
Migration strategy from SSRAM to NSGA	Each SSRAM subpopulation simultaneously copies its fittest individual to the NSGA population. The NSGA population is then reduced back to its fixed population size through the Pareto dominance and crowding distance criteria.

7.5. Computational Results

The number of solution evaluations for each algorithm run is given by

Num. of islands * Num. of new solutions for each island * Num. of generations

where the number of new solutions for each island is given by population size*(crossover rate + mutation rate), i.e. 100*(0.7+0.05) = 75 for my parameter values.

In order to carry out a performance comparison of the different algorithms the total number of solution evaluations for each run of the algorithm has been fixed at 330,000 evaluations by varying the number of generations. This means that there are 400 generations for the 11 island SSRAM(NM) algorithm and 4,400 generations for the single island NSGA2 algorithm.

For each run, the efficient frontier is outputted at four equally spaced intervals over the run, corresponding to 25%, 50% 75% and 100% of the total number of solution evaluations. The purpose of this is in order to determine the performance of the algorithm for different run lengths. Each algorithm has then been run 5 times, in C++, with the quality of the efficient frontiers produced then compared. In the remaining tables of this Chapter I have shaded in grey the cells in the table that correspond to when an algorithm is found to outperform.

7.5.1. Type SSRAM type algorithms

The type SSRAM algorithms are multiple single objective optimization algorithms, where the aim is to maximize the objective function at each of the risk aversion parameters. In order to compare the performance of these algorithms I calculate the average and maximum objective value, over the 5 algorithm runs, for each risk aversion value. I then compare each algorithm pair by calculating the percentage of the first algorithms average objective values that are higher than the second algorithms values. This is then repeated for the maximum, rather than average, objective values.

Table 7.4 shows the result of these calculations where a value of 100% means that the algorithm in the corresponding column has higher values at all the risk aversions points that the algorithm in the corresponding row.

Table 7.4: Performance comparison of the different type SSRAM algorithms

		Average objective value				Maximum objective value			
Percent through run	**Algorithm**	**SSRAM (NM)**	**SSRAM (FM)**	**SSRAM-CC**	**SSRAM-CCLS**	**SSRAM (NM)**	**SSRAM (FM)**	**SSRAM-CC**	**SSRAM-CCLS**
25%	**SSRAM(NM)**	N/A	64%	91%	100%	N/A	55%	91%	100%
	SSRAM(FM)	36%	N/A	91%	100%	45%	N/A	73%	91%
	SSRAM-CC	9%	9%	N/A	100%	9%	27%	N/A	100%
	SSRAM-CCLS	0%	0%	0%	N/A	0%	9%	0%	N/A
50%	**SSRAM(NM)**	N/A	64%	91%	100%	N/A	55%	91%	91%
	SSRAM(FM)	36%	N/A	91%	91%	45%	N/A	91%	82%
	SSRAM-CC	9%	9%	N/A	55%	9%	9%	N/A	55%
	SSRAM-CCLS	0%	9%	45%	N/A	9%	18%	45%	N/A
75%	**SSRAM(NM)**	N/A	64%	73%	82%	N/A	55%	73%	64%
	SSRAM(FM)	36%	N/A	82%	73%	45%	N/A	82%	73%
	SSRAM-CC	27%	18%	N/A	36%	27%	18%	N/A	55%
	SSRAM-CCLS	18%	27%	64%	N/A	36%	27%	45%	N/A
100%	**SSRAM(NM)**	N/A	64%	82%	82%	N/A	36%	82%	73%
	SSRAM(FM)	36%	N/A	91%	91%	64%	N/A	73%	91%
	SSRAM-CC	18%	9%	N/A	45%	18%	27%	N/A	55%
	SSRAM-CCLS	18%	9%	55%	N/A	27%	9%	45%	N/A

SSRAM(FM) is only found to perform slightly better than SSRAM(NM) on both the average and maximum objective values criteria. SSRAM-CC is found to perform better than both the SSRAM(NM) and SSRAM(FM) algorithms, with both the average objective and maximum objective values being higher at 73-90% of risk aversion points. Given this outperformance, it is no surprise that SSRAM-CCLS also performs better than either the SSRAM(NM) or SSRAM(FM) algorithms. The SSRAM-CCLS and SSRAM-CC algorithms perform comparably, with the exception being that at a quarter of the way through the run, the SSRAM-CCLS outperforms the SSRAM-CC at all the risk aversion points.

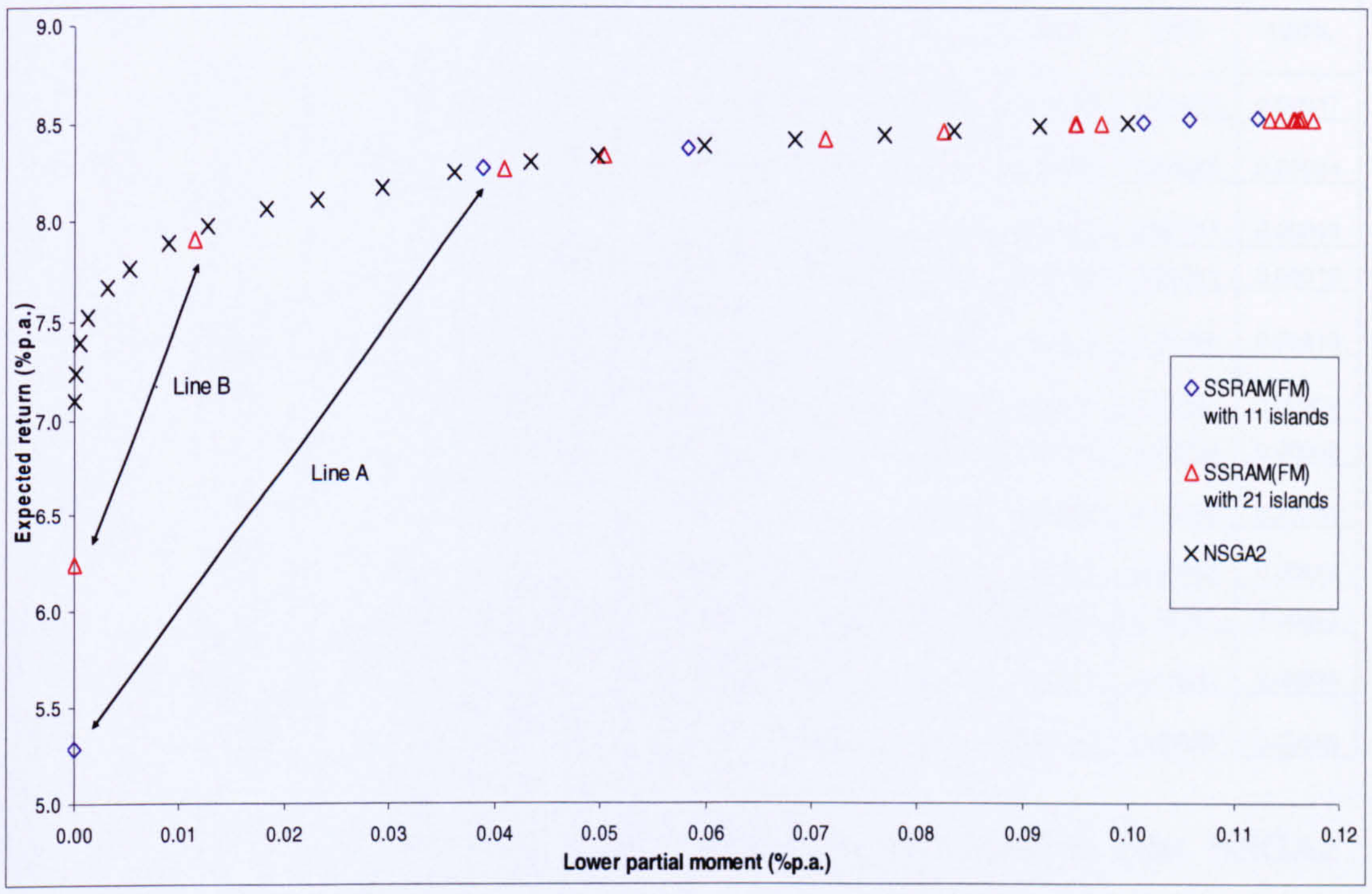
In summary, the addition of competitive coevolution migration to the SSRAM model shows an improvement over a no migration or fixed migration strategy. At shorter run lengths there is also found to be an improvement of the algorithm when a local search Lamarckian process also added.

One of the drawbacks of type SSRAM algorithms is the possibility of an uneven distribution of points over the efficient frontier. This was found to be a significant issue when partial lower moment was used as a risk measure. Figure 7.5 shows the efficient frontiers produced using both the SSRAM(FM) algorithm and NSGA2 algorithm for the call/put strategy when lower partial moment is the risk measure. It shows the heavily uneven distribution of points along the efficient frontier for the SSRAM(FM) model when the number of islands is either 11 or 21. This is in contrast to the more evenly distributed points produced by the NSGA2 algorithm. This difference is quantified by the considerably higher spacing measure of 1.54 for the 21 island SSRAM(FM) frontier compared to the 0.52 value for the 20 point NSGA2 frontier.

In this figure, the large gap between the two least risky points on the 11 point frontier is shown by line A. Although this gap is reduced in the 21 point frontier, as shown by line B, it is still marked. The disadvantage of these uneven frontiers is more severe since it occurs over the convex part of the frontier which has only been produced by the NSGA2 algorithm.

When using type SSRAM algorithms, the distribution of points along the efficient frontier depends on both the distribution of risk aversion parameter values along the [0,1] range and the exact problem being optimized. This is in contrast to the type NSGA2 algorithms which are designed to ensure that points are well distributed over the entire efficient frontier. Therefore, a key drawback of type SSRAM algorithms is that it is not possible to ensure that the chosen distribution of risk aversion parameter values will result in a well distributed frontier for any particular optimization problem considered.

Figure 7.5: Efficient frontier for the call/put derivative strategy where lower partial moment is the risk measure



7.5.2. Type NSGA2 algorithms

Table 7.5 details the performance metrics of the different type NSGA2 algorithms. As one would expect the efficient frontiers improve as the run length increases, which can be seen by the S metric and extent metric values increasing as the run length increases. The highest value of the S metric and extent metric are achieved by the NSGA2-MP (two islands) model. The lowest values of the spacing metric are achieved by the CCNSGA2 algorithm. The results also show that as the number of islands in NSGA2 and NSGA2-MP increases from one to two to four, the values of the spacing metric rise from the 0.31-0.33 range to 0.41-0.45 range. This can be explained by the disruption in the spacing of points along the efficient frontier caused by migration between the different islands. It should be possible to reduce this level of disruption by tailing of the degree of the migration towards the end of the run.

Table 7.5: Performance metrics of the different type NSGA2 algorithms

		Average value				Range			
Algorithm	**Percent through run**	**25%**	**50%**	**75%**	**100%**	**25%**	**50%**	**75%**	**100%**
NSGA2 (single island)	**S metric**	0.00286	0.00311	0.00317	0.00318	0.00032	0.00033	0.00016	0.00017
	Δ spacing metric	0.30964	0.32010	0.32131	0.32783	0.05363	0.03544	0.04089	0.03284
	Extent metric	0.05186	0.05476	0.05532	0.05565	0.00310	0.00362	0.00191	0.00246
NSGA2-MP (two islands)	**S metric**	0.00306	0.00321	0.00325	0.00327	0.00034	0.00023	0.00021	0.00019
	Δ spacing metric	0.35416	0.37084	0.36886	0.34007	0.06036	0.07047	0.04766	0.06819
	Extent metric	0.05425	0.05581	0.05616	0.05636	0.00377	0.00271	0.00242	0.00202
NSGA2-MP (four islands)	**S metric**	0.00289	0.00295	0.00318	0.00321	0.00064	0.00034	0.00058	0.00058
	Δ spacing metric	0.41700	0.42113	0.44560	0.41255	0.07599	0.06986	0.09369	0.23031
	Extent metric	0.05235	0.05328	0.05553	0.05581	0.00659	0.00422	0.00582	0.00583
CCNSGA2	**S metric**	0.00288	0.00294	0.00308	0.00313	0.00094	0.00077	0.00057	0.00045
	Δ spacing metric	0.28600	0.28253	0.29191	0.29621	0.05792	0.04676	0.04601	0.06949
	Extent metric	0.05207	0.05308	0.05449	0.05503	0.01017	0.00764	0.00662	0.00548

Table 7.6 shows the C metric values for the different type NSGA2 algorithms. The C metric values put the four algorithms in the following order of performance with the best first, NSGA2, CCNSGA2, NSGA2-MP(two islands), NSGA2-MP(four islands). The order of magnitude of the difference between the best and worse algorithm over the run is given by C metric values in the range of 0.31-0.37 and 0.13-0.17 for C(NSGA2, NSGA2-MP(four islands)) and C(NSGA2-MP(four islands)), NSGA2) respectively.

Table 7.7 detail the performance metrics of the SSNSGA2 and NSGA2 algorithms. The islands risk coefficient specifies the weighing used in the generalised semi-variance risk measure. Therefore, a value of 0, 0.5 and 1 refers to upper semi-variance, variance and lower semi-variance respectively (see equation 4.5). The table shows that the two algorithms have similar s metric, extent metric and spacing metric values. However, Table 7.8 shows that the SSNSGA2 algorithm has a better C metric value averaging 0.31 for C(SSNSGA2, NSGA2) compared to 0.17 for C(NSGA2, SSNSGA2).

Table 7.6: C metric of the different type NSGA2 algorithms

		C(algorithm in row, algorithm in column)			
Percent through run	**Algorithm**	**NSGA2**	**NSGA2-MP (2 islands)**	**NSGA2-MP (4 islands)**	**CCNSGA2**
25%	**NSGA2**	N/A	0.270	0.312	0.260
	NSGA2-MP (2 islands)	0.234	N/A	0.292	0.218
	NSGA2-MP (4 islands)	0.150	0.202	N/A	0.182
	CCNSGA2	0.252	0.242	0.330	N/A
50%	**NSGA2**	N/A	0.294	0.316	0.286
	NSGA2-MP (2 islands)	0.190	N/A	0.244	0.198
	NSGA2-MP (4 islands)	0.174	0.228	N/A	0.186
	CCNSGA2	0.202	0.238	0.262	N/A
75%	**NSGA2**	N/A	0.342	0.366	0.362
	NSGA2-MP (2 islands)	0.134	N/A	0.246	0.220
	NSGA2-MP (4 islands)	0.162	0.194	N/A	0.212
	CCNSGA2	0.134	0.210	0.236	N/A
100%	**NSGA2**	N/A	0.366	0.312	0.288
	NSGA2-MP (2 islands)	0.126	N/A	0.234	0.170
	NSGA2-MP (4 islands)	0.132	0.238	N/A	0.168
	CCNSGA2	0.162	0.292	0.278	N/A

Table 7.7: Performance metrics of the SSNSGA2 and NSGA2 algorithms

	Algorithm	**NSGA2 (single island)**			**SSNSGA2 (five islands)**		
The islands risk coefficient	**Statistic**	**S metric**	**Δ spacing metric**	**Extent metric**	**S metric**	**Δ spacing metric**	**Extent metric**
0.00	**Average**	0.00193	0.36384	0.04194	0.00194	0.32721	0.04083
	Range	0.00021	0.08319	0.00233	0.00041	0.03304	0.00458
0.25	**Average**	0.00172	0.36451	0.04166	0.00175	0.35288	0.04163
	Range	0.00013	0.04150	0.00156	0.00020	0.10346	0.00167
0.50	**Average**	0.00150	0.35404	0.04131	0.00150	0.36448	0.04124
	Range	0.00009	0.05300	0.00251	0.00026	0.02874	0.00210
0.75	**Average**	0.00134	0.36251	0.04087	0.00135	0.39111	0.04074
	Range	0.00037	0.06799	0.00350	0.00029	0.07811	0.00290
1.00	**Average**	0.00116	0.38294	0.04025	0.00120	0.40012	0.04031
	Range	0.00008	0.08964	0.00104	0.00021	0.07816	0.00199
Average over all the islands	**Average**	**0.00153**	**0.36557**	**0.04121**	**0.00155**	**0.36716**	**0.04095**
	Range	**0.00018**	**0.06707**	**0.00219**	**0.00027**	**0.06430**	**0.00265**

Table 7.8: C metric of the SSNSGA2 and NSGA2 algorithms

The islands risk coefficient	C(NSGA2, SSNSGA2)	C(SSNSGA2, NSGA2)
0.00	0.200	0.320
0.25	0.186	0.296
0.50	0.140	0.254
0.75	0.140	0.380
1.00	0.184	0.294
Average over all the islands	**0.170**	**0.309**

7.5.3. Hybrid SSRAM and NSGA2 algorithm

Table 7.9 shows how the performance of the SSRAM-CC—NSGA2 compared with the different type SSRAM algorithms.

Table 7.9: Performance comparison of the type SSRAM algorithms and hybrid SSRAM-CC—NSGA2 algorithm

		The percent of the SSRAM-CC—NSGA2 average and maximum objective values that are higher than those of the algorithm in the corresponding row	
Percent through run	**Algorithm**	**Average objective value**	**Maximum objective value**
25%	SSRAM(NM)	91%	91%
	SSRAM(FM)	91%	73%
	SSRAM-CC	45%	55%
	SSRAM-CCLS	0%	9%
50%	SSRAM(NM)	100%	100%
	SSRAM(FM)	100%	82%
	SSRAM-CC	45%	45%
	SSRAM-CCLS	36%	45%
75%	SSRAM(NM)	82%	82%
	SSRAM(FM)	82%	82%
	SSRAM-CC	55%	55%
	SSRAM-CCLS	55%	55%
100%	SSRAM(NM)	73%	64%
	SSRAM(FM)	100%	82%
	SSRAM-CC	73%	64%
	SSRAM-CCLS	55%	55%

Table 7.9 shows that the SSRAM-CC—NSGA2 algorithm performs better than the SSRAM(NM) and SSRAM(FM) algorithms at all run lengths. The SSRAM-CC—NSGA2 also performs better than the SSRAM-CC and SSRAM-CCLS in 55-100% of the time for the longest two run lengths.

Table 7.10 shows the performance metrics of the type NSGA2 algorithms and hybrid SSRAM-CC— NSGA2 algorithm. It shows that the hybrid algorithm has a higher S metric and extent metric values than the other algorithms, with the hybrid values shown to be more consistent due to the lower corresponding range values. The hybrid algorithm has the second lowest spacing metric, with only the CCNSGA2 algorithm having a lower value.

Table 7.10: Performance metrics of the type NSGA2 algorithms and hybrid SSRAM-CC—NSGA2 algorithm

Algorithm	Performance statistic at run end	Average value	Range
NSGA2 (single island)	S metric	0.00318	0.00017
	Δ spacing metric	0.32783	0.03284
	Extent metric	0.05565	0.00246
NSGA2-MP (two islands)	S metric	0.00327	0.00019
	Δ spacing metric	0.34007	0.06819
	Extent metric	0.05636	0.00202
NSGA2-MP (four islands)	S metric	0.00321	0.00058
	Δ spacing metric	0.41255	0.23031
	Extent metric	0.05581	0.00583
CCNSGA2	S metric	0.00313	0.00045
	Δ spacing metric	0.29621	0.06949
	Extent metric	0.05503	0.00548
SSRAM-CC—NSGA2	S metric	0.00357	0.00006
	Δ spacing metric	0.30983	0.06002
	Extent metric	0.05999	0.00097

Table 7.11 shows the C metric values for the type NSGA2 algorithm and hybrid SSRAM-CC—NSGA2 algorithm. The hybrid algorithm has better C metric values, i.e. in the range 0.26-0.40, than any of the type NSGA2 algorithms, i.e. in the range 0.09 – 0.17.

Table 7.11: C metric of the type NSGA2 algorithms and hybrid SSRAM-CC— NSGA2 algorithm. R refers to row algorithm

Algorithm	C(R,SSRAM-CC—NSGA2)	C(SSRAM-CC—NSGA2, R)
NSGA2	0.170	0.262
NSGA2-MP (2 islands)	0.092	0.404
NSGA2-MP (4 islands)	0.104	0.380
CCNSGA2	0.138	0.346

7.6. Conclusion

I have shown that the standard SSRAM(FM) algorithm can be improved using a competitive coevolution migration mechanism that adapts the migration rate throughout the run, and that this algorithm can be further improved using a local search Lamarckian process. Comparing the different type NSGA2 algorithms I have also shown that one is also able to solve the portfolio optimization problem using a cooperative coevolution algorithm. I have demonstrated the benefit of using a stepping stone NSGA2 model when solving a portfolio optimization problem simultaneously for different specifications of the risk measure. Finally, I have shown that a hybrid of the type SSRAM and NSGA2 algorithms performs better than either model in isolation.

Chapter 8

Conclusions

In my research I have used a heuristic approach to solve a multiperiod equity portfolio optimization problem. This has enabled me to investigate some important issues regarding the benefits of incorporating derivatives in equity portfolio management and the benefits of allowing a more frequent rebalancing period. I have also investigated the performance of alternative evolutionary algorithms and novel ways to improve on the standard algorithms used.

I have demonstrated how the inclusion of derivatives within equity portfolio optimization can significantly improve the efficient frontier. I have shown how through the sale of call options and/or purchase of put options an investor is able to improve on the achievable risk/return tradeoffs over certain parts of their efficient frontier. To my knowledge the benefits of including call and put options within an equity portfolio under a heuristic optimization method has not yet been demonstrated elsewhere in the current literature. I have also shown that when placing a higher importance on downside risk within the risk measure an investor will increase their allocation in put options within their portfolio.

I then went on to demonstrate how the use of multi-period strategies can significantly improve the efficient frontier. By using a CPPI strategy or fixed equity cluster strategy an investor is able to improve on the achievable risk/return tradeoffs over certain parts of their efficient frontier. I have shown that when placing a higher importance on downside risk within the risk measure the benefits of more frequent rebalancing is seen to increase. I have also found that a dynamic

investment strategy is more likely to outperform a strategy that includes the use of derivatives when a greater importance is placed on downside risk.

I then investigated the performance of a variety of novel evolutionary algorithms, utilizing techniques from island models, coevolution and memetic algorithms. I have demonstrated that the standard SSRAM(FM) algorithm can be improved using a competitive coevolution migration mechanism that adapts the migration rate throughout the run, and that this algorithm can be further improved using a local search Lamarckian process. Comparing the different type NSGA2 algorithms I have also shown that one is also able to solve the portfolio optimization using a cooperative coevolution algorithm, which to my knowledge has not previously been implemented. I have demonstrated the benefit of using a stepping stone NSGA2 model when solving a portfolio optimization problem simultaneously for different specifications of the risk measure. Finally, I have shown that a hybrid of the type SSRAM and NSGA2 algorithms performs better than either model in isolation.

I now briefly discuss the practical implications of my findings. My investigations show that investors should seriously consider whether to incorporate derivative products when deciding on their asset allocations. This decision is likely to depend on factors such as their investment objectives, risk tolerances, investment expertise and portfolio size. At the point of deciding on their initial asset allocation, an investor should also decide on how they intend to rebalance their portfolio in the future following changes in market levels. By taking this dynamic view of portfolio management from the start, rather than either a buy/hold strategy or just reacting to future market moves in an unsystematic way, an investor should be able to achieve a superior investment strategy.

Evolutionary algorithms provide a flexible and robust method for solving practical portfolio optimization problems. The choice of which evolutionary algorithm is most suitability for any particular portfolio optimization problem

will depend on factors such as characteristics of the problem (e.g. problem size), the speed at which an answer is required, the computational power available, the degree to which an exact answer is required and the expertise of the programmers.

8.1. Areas for further research

Below I give details on areas for possible further research.

- An interesting generalization of my problem would be to extend my problem to a surplus optimization problem that includes the modelling of liabilities which are dependent on future interest rates and inflation. On the asset side one could also include bonds and derivatives products dependent on interest rates and inflation rates. One could then investigate how beneficial the inclusion of derivatives and dynamic asset allocation are for financial organizations with liability profiles such as pension schemes and insurance companies.
- A class of evolutionary algorithms where further research would be beneficial is that of coevolution algorithms. Firstly, to the specific application of portfolio optimization, further investigation of the performance of the cooperative coevolution algorithm could be looked at for a problem of a portfolio of say 30 stocks and 30 bonds. The investigation would compare the performance of a standard algorithm that optimizes the entire problem simultaneously to that of a cooperative coevolution model where the problem was subdivided into the two solution parts of finding the equity portfolio and bond portfolio. Secondly, the benefits of the competitive coevolution migration model could be investigated for different types of optimization problems. In particular, one could investigate the relative performance of this algorithm for the case when each island is homogenous to that when each island has a related but unique problem that needs to be optimized.

Abbreviations

CCNSGA2	Competitive coevolution NSGA2 model
CPPI	Constant Proportion Portfolio Insurance
DRGA	Divided Range Genetic Algorithm
MOGA	Multiobjective Genetic Algorithm
NPGA	Niched Pareto Genetic Algorithm
NSGA2	Non-dominated Sorting Genetic Algorithm II
NSGA2-MP	NSGA2 algorithm with multiple populations
OBPI	Option Based Portfolio Insurance
PCA	Principal Component Analysis
PESA	Pareto Envelop-based Selection Algorithm
PI	Portfolio Insurance
PSFGA	Parallel Single Front Genetic Algorithm
SPEA	Strength Pareto Evolutionary Algorithm
SSNSGA2	Stepping stone NSGA2 model
SSRAM	Stepping stone risk aversion model algorithm
SSRAM-CC	SSRAM model with competitive coevolution migration
SSRAM-CCLS	SSRAM-CC algorithm with Lamarckian local search
SSRAM-CC—NSGA2	Hybrid of the SSRAM-CC and NSGA2 algorithms
SSRAM(FM)	Fixed migration SSRAM algorithm
SSRAM(NM)	No migration SSRAM algorithm
TRI	Total Return Index
VEGA	Vector-Evaluated Genetic Algorithm

References

Acerbi, C. & Tasche, D. 2002, "Expected Shortfall: A Natural Coherent Alternative to Value at Risk", *Economic Notes,* vol. 31, no. 2, pp. 379-388.

Adcock, C.J. & Meade, N. 1994, "A Simple Algorithm to Incorporate Transactions Costs in Quadratic Optimisation", *European Journal of Operational Research,* vol. 79, no. 1, pp. 85-94.

Adler, M. 1987, "Global Asset Allocation: Some Uneasy Questions", *Investment Management Review,* Sept/Oct 1987, pp. 13-18.

Alexander, C. 2002, "Principal Component Models for Generating Large GARCH Covariance Matrices", *Economic Notes, Siena.,* vol. 31, no. 2, pp. 337-360.

Alexander, C. 2001, *Market Models: A Guide to Financial Data Analysis,* John Wiley and Sons Ltd.

Angeline, P.J. 2000, "Competitive Fitness Evaluation" in *Evolutionary Computation 2: Advanced Algorithms and Operators*, eds. T. Back, D.B. Fogel & Z. Michalewicz, Institute of Physics Publishing, pp. 12-14.

Armananzas, R. & Lozano, J.A. 2005, "A Multiobjective Approach to the Portfolio Optimization Problem", *IEEE International Conference on Evolutionary Computation Proceedings,* vol. 2, pp. 1388-1395.

Arnone, S., Loraschi, A. & Tettamanzi, A. 1993, "A Genetic Approach to Portfolio Selection", *Neural Network World. International Journal of Neural and Mass-Parallel Computing and Information Systems,* no. 3, pp. 597-604.

Artzner, P., Delbaen, F., Eber, J.M. & Heath, D. 1999, "Coherent Measures of Risk", *Mathematical Finance,* vol. 9, no. 3, pp. 203-228.

Artzner, P., Delbaen, F., Eber, J.M. & Heath, D. 1997, "Thinking Coherently: Generalised Scenarios rather than VAR should be used when Calculating Regulatory Capital", *Risk, London, Risk Magazine Limited,* vol. 10, no. 11, pp. 68-71.

Back, T. 1996, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms.* Oxford University Press Inc, USA.

Back, T., Fogel, D.B. & Michalewicz, Z. (eds) 1997, *Handbook of Evolutionary Computation*, Institute of Physics Publishing and Oxford University Press.

Back, T., Hoffmeister, F. & Schwefel, H.P. 1991, "A Survey of Evolution Strategies", *Proceedings of the Fourth International Conference on Genetic Algorithms,* pp. 2-9.

Baglioni, S., da Costa Pereira, C., Sorbello, D. & Tettamanzi, A.G. 2000, "An Evolutionary Approach to Multiperiod Asset Allocation", *Lecture Notes in Computer Science,* no. 1802, pp. 225-236.

Baker, J.E. 1987, "Reducing Bias and Inefficiency in the Selection Algorithm", *Proceedings of the Second International Conference of Genetic Algorithms,* pp. 14-21.

Barberis, N. 2000, "Investing for the Long Run when Returns are Predictable", *Journal of Finance,* vol. 55, no. 1, pp. 225-264.

Barr, R.S., Golden, B.L., Kelly, J.P., Resende, M.G.C. & Stewart, W.R. 1995, "Designing and Reporting on Computational Experiments with Heuristic Methods", *Journal of Heuristics,* vol. 6, no. 1, pp. 9-32.

Bauer, R.J. 1994, *Genetic Algorithms and Investment Strategies: Alternative Approach to Neural Networks and Chaos Theory,* John Wiley & Sons Inc.

Bean, J.C. & Hadj-Alouane, A.B. 1992, *A Dual Genetic Algorithm for Bounded Integer Programs*, Department of Industrial and Operations Engineering. The University of Michigan Ann Arbor.

Beasley, D., Bull, D.R. & Martin, R.R. 1993b, "An Overview of Genetic Algorithms: Part 2, Research Topics", *University Computing,* vol. 15, no. 4, pp. 170-181.

Beasley, D., Martin, R.R. & Bull, D.R. 1993a, "An Overview of Genetic Algorithms: Part 1, Fundamentals", *University Computing,* vol. 15, no. 2, pp. 58-69.

Beasley, J.E. 2002, "Population Heuristics" in *Handbook of Applied Optimization*, eds. P.M. Pardalos & M.G.C. Resende, Oxford University Press, pp. 138-156.

Beasley, J.E., Meade, N. & Chang, T.J. 2003, "An Evolutionary Heuristic for the Index Tracking Problem", *European Journal of Operational Research,* vol. 148, no. 3, pp. 621-643.

Berger, A.J., Glover, F. & Mulvey, J.M. 1995, *Solving Global Optimization Problems in Long-Term Financial Planning*.

Best, M.J. & Grauer, R.R. 1991, "On the Sensitivity of Mean-Variance-Efficient Portfolios to Changes in Asset Means: Some Analytical and Computational Results", *Review of Financial Studies,* vol. 4, no. 2, pp. 315.

Black, F. & Litterman, R. 1992, "Global Portfolio Optimization", *Financial Analysts Journal,* vol. 48, no. 5, pp. 28.

Black, F. & Scholes, M. 1973, "The Pricing of Options and Corporate Liabilities", *Journal of Political Economy,* vol. 81, no. 3, pp. 637-654.

Blickle, T. & Thiele, L. 1996, "A Comparison of Selection Schemes Used in Evolutionary Algorithms", *Evolutionary Computation,* vol. 4, no. 4, pp. 361-394.

Bobo, F. & Goldberg, D.E. 1997, "Decision Making in a Hybrid Genetic Algorithm", *The 1997 IEEE International Conference on Evolutionary Computation,* pp. 121-125.

Bollerslev, T. 1986, "Generalised Autoregressive Conditional Heteroskedasticy". Journal of Econometrics.

Booker, L. 1987, "Improving Search in Genetic Algorithms" in *Genetic Algorithms and Simulated Annealing*, ed. L. Davis, Morgan Kaufmann.

Brandimarte, P. 2002, *Numerical Methods in Finance,* John Wiley & Sons Inc.

Brennan, M.J., Schwartz, E.S. & Lagnado, R. 1997, "Strategic Asset Allocation", *Journal of Economic Dynamics and Control,* vol. 21, no. 8/9, pp. 1377-1406.

Bucci, A. & Pollack, J.B. 2005, "On Identifying Global Optima in Cooperative Coevolution", *Gecco Conference,* CONF 7; VOL 1, pp. 539-544.

Campbell, J.Y. & Viceira, L.M. 2002, *Strategic Asset Allocation: Portfolio Choice for Long-term Investors,* Oxford University Press.

Cantu-Paz, E. 2001, "Migration Policies, Selection Pressure, and Parallel Evolutionary Algorithms", *Journal of Heuristics,* vol. 7, PART 4, pp. 311-334.

Carino, D.R., Kent, T., Myers, D.H. & Stacy, G. 1995, "The Russell-Yasuda Kasai Model: an Asset/Liability Model for a Japanese Insurance Company using Multistage Stochastic Programming", *Newton Institute Publications,* no. 10, pp. 609-633.

Carino, D.R. & Turner, A.L. 1995, "Multiperiod Asset Allocation with Derivative Assets", *Newton Institute Publications,* no. 10, pp. 182-204.

Chang, T.J., Meade, N., Beasley,J.E. & Sharaiha, Y.M. 2000, "Heuristics for Cardinality Constrained Portfolio Optimisation", *Computers & Operations Research,* vol. 27, no. 13, pp. 1271-1302.

Chen, S.H. (ed) 2002, *Genetic Algorithms and Genetic Programming in Computational Finance*, Kluwer Academic Publishers.

Chen, W., Zhang, R. & Zhang, W. 2006, "Management of Stochastic Portfolio using Improved Genetic Algorithm", *Proceedings of the Sixth World Congress on Intelligent Control and Automation,* vol. 1, pp. 3557-3561.

Chi-cheong, W., Cheung, B., Tang, G. & Man-chung, C. 2002, "Genetic Algorithms in Multi-Stage Asset Allocation System", *IEEE International Conference on Systems Man and Cybernetics,* vol. 3, pp. TA1F5.

Chidambaran, N., Triqueros, J. & Lee, C.W.J. 2002, "Option Pricing Via Genetic Programming", *Studies in Fuzziness and Soft Computing,* vol. 100, pp. 383-398.

Chopra, V.K. 1993, "Improving Optimization", *Journal of Investing,* Fall, pp. 51-59.

Chopra, V.K. & Ziemba, W.T. 1993, "The Effect of Errors in Means, Variances, and Covariances on Optimal Portfolio Choice", *Journal of Portfolio Management,* vol. 19, no. 2, pp. 6-11.

Chopra, V.K., Hensel, C.R. & Turner, A.L. 1993, "Massaging Mean-variance Inputs: Returns from Alternative Global Investment Strategies in the 1980s", *Management Science,* vol. 39, no. 7, pp. 845-855.

Christofides, N. 1994, "The Bionomic Algorithm", *Presented at the AIRO'94 Conference.*

Chu, P.C. & Beasley, J.E. 1998, "Constraint Handling in Genetic Algorithms: The Set Partitioning Problem", *Journal of Heuristics,* vol. 4, no. 4, pp. 323-358.

Coello Coello, C.A. 2000, "An Updated Survey of GA-Based Multiobjective Optimization Techniques", *ACM Computing Surveys,* vol. 32, no. 2, pp. 109-143.

Cohoon, J.P., Hedge, S.U., Martin, W.N. & Richards, D.S. 1991, "Distributed Genetic Algorithms for the Floorplan Design Problem", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 10, no. 4, pp. 483-492.

Cohoon, J.P., Hedge, S.U., Martin, W.N. & Richards, D.S. 1987, "Punctuated Equilibria: A Parallel Genetic Algorithm", *Proceedings of the Second International Conference on Genetic Algorithms,* pp. 148-154.

Collins, R.J. & Jefferson, D.R. 1991, "Selection in Massively Parallel Genetic Algorithms", *Proceedings of the Fourth International Conference on Genetic Algorithms,* pp. 249-256.

Cox, J.C. & Leland, H.E. 2000, "On Dynamic Investment Strategies", *Journal of Economic Dynamics & Control,* vol. 24, no. 11/12, pp. 1859-1880.

Davis, L. 1991, *Handbook of Genetic Algorithms,* Van Nostrand Reinhold.

Davis, L. 1989, "Adapting Operator Probabilities in Genetic Algorithms", *Proceedings of the Third International Conference on Genetic Algorithms,* pp. 61-69.

De Jong, K. 2004, "Two Grand Challenges for EC", *Gena Dordrecht,* vol. 11, pp. 37-52.

De Jong, K. 1975, *The Analysis and Behaviour of a Class of Genetic Adaptive Systems*, University of Michigan.

De Jong, K. & Sarma, J. 1995, "On Decentralizing Selection Algorithms", *Proceedings of the International Conference on Genetic Algorithms,* CONF 6, pp. 17-23.

De Jong, K. & Sarma, J. 1993, "Generation Gaps Revisited" in *Foundations of Genetic Algorithms 2*, ed. D. Whitley, Morgan Kaufmann, pp. 19-28.

De Toro, F., Ortega, J., Fernandez, J. & Diaz, A. 2002, "PSFGA: A Parallel Genetic Algorithm for Multiobjective Optimization", *Euromicro Workshop on Parallel Distributed and Network Based Processing,* no. 10, pp. 384-392.

Deb, K. 2001, *Multi-objective Optimization Using Evolutionary Algorithms,* John Wiley and Sons Ltd.

Deb, K., Agrawal, S., Pratab, A. and Meyarivan, T. 2000, "A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGAII" in Proceedings of the Parallel Problem Solving from Nature VI Conference, ed M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J.J. Merelo and H.P. Schwefel, Springer, pp 849-858.

Deb, K. & Goldberg, D.E. 1989, "An Investigation of Niche and Species Formation in Genetic Function Optimization", *Proceedings of the Third International Conference on Genetic Algorithms,* pp. 42-50.

Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. 2002, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation,* vol. 6, no. 2, pp. 182-197.

Derigs, U. & Nickel, N. 2004, "On a Local-Search Heuristic for a Class of Tracking Error Minimization Problems in Portfolio Management", *Annals of Operations Research,* vol. 131, no. 1-4, pp. 45-77.

Derigs, U. & Nickel, N. 2003, "Meta-heuristic based Decision Support for Portfolio Optimization with a Case Study on Tracking Error Minimization in Passive Portfolio Management", *OR Spectrum,* vol. 25, no. 3, pp. 345-378.

Doerner, K., Gutjahr, W.J., Hartl, R.F., Strauss, C. & Stummer, C. 2004, "Pareto Ant Colony Optimization: A Metaheuristic Approach to Multiobjective Portfolio Selection", *Annals of Operations Research,* vol. 131, no. 1-4, pp. 79-99.

Doisan, L. 2005, "A Multi-Objective Evolutionary Approach to the Portfolio Optimization Problem", *Proceedings of the 2005 International Conference on*

Computational Intelligence for Modeling, Control and Automation, and the International Conference on Intelligent Agents, Web Technologies and Internet Commerce. (CIMCA-IAWTIC'05), vol. 2, pp. 183-187.

Donaldson, L. (ed) 1997, *Equity Derivatives: Applications in Risk Management and Investment*, Risk Books.

Dueck, G. & Scheuer, T. 1990, "Threshold Accepting: A General Purpose Algorithm Appearing Superior to Simulated Annealing", *Journal of Computational Physics,* vol. 90, no. 1, pp. 161-175.

Dunis, C. (ed) 2000, *Advances in Quantitative Asset Management*, Kluwer Academic Publishers.

Dybvig, P.H. & Dybvig, P.H. 1988, "Inefficient Dynamic Portfolio Strategies or How to Throw Away a Million Dollars in the Stock Market", *Review of Financial Studies,* vol. 1, no. 1.

Eby, D., Averill, R.C., Punch, W.F. & Goodman, E.D. 1999, "Optimal Design of Flywheels Using an Injection Island Genetic Algorithm", *Artificial Intelligence for Engineering Design Analysis and Manufacturing,* vol. 13, no. 5, pp. 327-340.

Ehrgott, M., Klamroth, K. & Schwehm, C. 2004, "An MCDM Approach to Portfolio Optimization", *European Journal of Operational Research,* vol. 155, no. 3, pp. 752-770.

Eiben, A.E. & Smith, J.E. 2003, *Introduction to Evolutionary Computing,* Springer-Verlag Berlin.

Eldredge, N. & Gould, J.E. 1972, "Punctuated Equilibria: An Alternative to Phyletic Gradualism" in *Models of Paleobiology*, ed. T.J.M. Schopf, Cooper Freeman, pp. 82-115.

Elmihoub, T., Hopgood, A.A., Nolle, L. & Battersby, A. 2004, "Performance of Hybrid Genetic Algorithms Incorporating Local Search", *Proceedings of the Eighth European Simulation Multiconference*.

Engle, R.F. 1982, "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of UK Inflation". Econometrica.

Engle, R.F., Ng, V. & Rothschild, M. 1990, "Asset Pricing with a Factor ARCH Covariance Structure: Empirical Estimates for Treasury Bills". Journal of Econometrics.

Espinoza, F.P., Minsker, B.S. & Goldberg, D.E. 2001, "A Self Adaptive Hybrid Genetic Algorithm", *Gecco Conference,* no. 10, pp. 759.

Fama, E.F. 1965, "The Behavior of Stock-Market Prices", *Journal of Business,* vol. 38, no. 1, pp. 34-105.

Farrell, C., Kieronska, D. & Schulze, M. 1994, "Genetic Algorithms for Network Division Problem", *IEEE World Congress on Computational Intelligence,* vol. 1, pp. 423-428.

Feinstein, C.D. & Thapa, M.N. 1993, "Notes: A Reformulation of a Mean-absolute Deviation Portfolio Optimization Model", *Management Science,* vol. 39, no. 12, pp. 1552-1553.

Feiring, B.R., Wong, W., Poon, M. & Chan, Y.C. 1994, "Portfolio Selection in Downside Risk Optimization Approach: Application to the Hong Kong Stock Market", *International Journal of Systems Science,* vol. 25, no. 11, pp. 1921.

Fernández, A. & Gómez, S. 2007, "Portfolio Selection using Neural Networks", *Computers & Operations Research,* vol. 34, no. 4, pp. 1177-1191.

Fieldsend, J.E., Matatko, J. & Peng, M. 2004, "Cardinality Constrained Portfolio Optimisation", *Lecture Notes in Computer Science,* no. 3177, pp. 788-793.

Fogarty, T.C. 1989, "Varying the Probability of Mutation in the Genetic Algorithm", *Proceedings of the Third International Conference on Genetic algorithms,* pp. 104-109.

Fogel, D.B. 2006, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence,* 3rd edn, John Wiley & Sons Inc.

Fogel, D.B. 1994, "Asymptotic Convergence Properties of Genetic Algorithms and Evolutionary Programming: Analysis and Experiments", *Cybernetics and Systems, Washington.,* vol. 25, no. 3, pp. 389.

Fogel, D.B. & Stayton, L.C. 1994, "On the Effectiveness of Crossover in Simulated Evolutionary Optimization", *Biosystems, Amsterdam.,* vol. 32, no. 3, pp. 171.

Fogel, G.B. & Fogel, D.B. 1995, "Continuous Evolutionary Programming: Analysis and Experiments", *Cybernetics and Systems, Washington.,* vol. 26, no. 1, pp. 79.

Fonseca, C.M. & Fleming, P.J. 1995, "Multiobjective Genetic Algorithms Made Easy: Selection, Sharing and Mating Restriction", *IEE Conference Publication,* no. 414, pp. 45.

Fonseca, C.M. & Fleming, P.J. 1993, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization", *Genetic algorithms*, ed. S. Forrest, Morgan Kaufmann, pp. 416.

Freedman, R.S. & DiGiorgio, R. 1993, "A Comparison of Stochastic Search Heuristics for Portfolio Optimization", *Second International Conference on Artificial Intelligence Applications on Wall Street,* pp. 149-151.

Frost, P.A. & Savarino, J.E. 1988, "For Better Performance: Constrain Portfolio Weights", *Journal of Portfolio Management,* vol. 15, no. 1, pp. 29-34.

Gilli, M. & Kellezi, E. 2002b, "The Threshold Accepting Heuristic for Index Tracking" in *Financial Engineering, E-Commerce and Supply Chain.*, eds. P.M. Pardalos & V.K. Tsitsiringos, Kluwer Academic Publishers, pp. 1-18.

Gilli, M. & Kellezi, E. 2002a, "Portfolio Optimization with VaR and Expected Shortfall", *Applied Optimization,* vol. 74, pp. 167-182.

Glover, F. 1986, "Future Paths for Integer Programming and Links to Artificial Intelligence", *Computers and Operations Research,* vol. 13, no. 5, pp. 533-549.

Glover, F. & Laguna, M. 1997, *Tabu Search,* Kluwer Academic Publishers.

Glover, F., Laguna, M. & Martí, R. 2003, "Scatter Search" in *Advances in Evolutionary Computation: Theory and Applications*, eds. A. Ghosh & S. Tsutsui, Springer-Verlag, pp. 519-538.

Glover, F., Mulvey, J.M. & Hoyland, K. 1995, "Solving Dynamic Stochastic Control Problems in Finance Using Tabu Search with Variable Scaling", pp. 429-448.

Goldberg, D.E. 1989, *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison Wesley.

Goldberg, D.E. & Deb, K. 1991, "A Comparative Analysis of Selection Schemes used in Genetic Algorithms" in *Foundations of Genetic Algorithms*, ed. G.J.E. Rawlins, Morgan Kaufmann, pp. 69-93.

Goldberg, D.E., Korb, B. & Deb, K. 1989, "Messy Genetic Algorithms: Motivation, Analysis, and First Results", *Complex Systems,* no. 3, pp. 493-530.

Goldberg, D.E. & Richardson, J. 1987, "Genetic Algorithms with Sharing for Multimodal Function Optimization", *Proceedings of the Second International Conference on Genetic Algorithms and their Application,* pp. 41-49.

Golden, B.L. & Wasil, E.A. 2002, "Metaheuristics: Introduction" in *Handbook of Applied Optimization*, eds. P.M. Pardalos & M.G.C. Resende, Oxford University Press, pp. 123-130.

Gorges-Schleuter, M. 1989, "ASPARAGOS: An Asynchronous Parallel Genetic Optimization Strategy", *Proceedings of the Third International Conference on Genetic Algorithms,* pp. 422-427.

Grefenstette, J. 2000b, "Efficient Implementation of Algorithms" in *Evolutionary Computation 2: Advanced Algorithms and Operators*, eds. T. Back, D.B. Fogel & Z. Michalewicz, Institute of Physics Publishing.

Grefenstette, J. 2000a, "Proportional Selection and Sampling Algorithms" in *Evolutionary Computation 1: Basic Algorithms and Operators*, eds. T. Back, D.B. Fogel & Z. Michalewicz, Institute of Physics Publishing, pp. 172-180.

Grefenstette, J. 1986, "Optimization of control Parameters for Genetic Algorithms", *IEEE Transactions on Systems, Man and Cybernetics.*, vol. 16, no. 1, pp. 122-128.

Grinold, R.C. 1999, "Mean-Variance and Scenario-Based Approaches to Portfolio Selection", *Journal of Portfolio Management,* vol. 25, no. 2, pp. 10-22.

Gulpinar, N., Rustem, B. & Settergren, R. 2004, "Simulation and Optimization Approaches to Scenario Tree Generation", *Journal of Economic Dynamics and Control,* vol. 28, no. 7, pp. 1291-1315.

Hamilton, J.D. 1994, *Time Series Analysis,* Princeton University Press.

Hamza, F. & Janssen, J. 1998, "The Mean Semivariances Approach to Realistic Portfolio Optimization Subject to Transaction Costs", *Applied Stochastic Models and Data Analysis,* vol. 14, no. 4, pp. 275-284.

Hamza, F. & Janssen, J. 1996, "Linear Approach for Solving Large-Scale Portfolio Optimization Problems in a Lognormal Market", *AFIR Colloquium*.

Hansen, P. 1986, "The Steepest Ascent Mildest Descent Heuristic for Combinatorial Programming", *Congress on Numerical Methods in Combinatorial Optimization*.

Harlow, W.V. 1991, "Asset Allocation in a Downside-Risk Framework", *Financial Analysts Journal,* vol. 47, no. 5, pp. 28-40.

Hart, W.E. 1994, "Adaptive Global Optimization with Local Search", *University of California*.

Haugh, M.B. & Lo, A.W. 2001, "Asset Allocation and Derivatives", *Quantitative Finance,* vol. 1, no. 1, pp. 45-72.

Haupt, R.L. & Haupt, S.E. 2004, *Practical Genetic Algorithms,* 2nd edn, John Wiley & Sons Inc.

Hensel, C.R. & Turner, A.L. 1995, "Making Superior Asset Allocation Decisions: a Practitioner's Guide", *Newton Institute Publications,* no. 10, pp. 62-86.

Hiemstra, Y. 1996, "Applying Neural Networks and Genetic Algorithms to Tactical Asset Allocation", *Finance & Technology, Neurove$t Journal.,* vol. 4, no. 3, pp. 8-15.

Hiroyasu, T., Miki, M. & Watanabe, S. 1999, "Divided Range Genetic Algorithms in Multiobjective Optimization problems", *International Workshop on Emergent Synthesis (IWES 99).*

Hodges, S.D. 1995, "Dynamic Asset Allocation: Insights from Theory" in *Mathematical Models in Finance*, eds. S.D. Howison, F.P. Kelly & P. Wilmott, pp. 137-148.

Holland, J.H. 1975, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence.* University of Michigan Press.

Horn, J. 1997, "Multicriterion Decision Making" in *Handbook of Evolutionary Computation*, eds. T. Back, D.B. Fogel & Z. Michalewicz, Institute of Physics Publishing and Oxford University Press, .

Horn, J. & Nafpliotis, N. 1993, *Multiobjective Optimization using the Niched Pareto Genetic Algorithm*, University of Illinois.

Hull, J.C. 2005, *Options, Futures and Other Derivatives,* 6th edn, Prentice Hall.

Iorio, A.W. & Li, X. 2004, "A Cooperative Coevolutionary Multiobjective Algorithm Using Non-dominated Sorting", *Lecture Notes in Computer Science,* no. 3102, pp. 537-548.

Ishibuchi, H., Yoshida, T. & Murata, T. 2002, "Balance between Genetic Search and Local Search in Hybrid Evolutionary Multi-Criterion Optimization Algorithms", *Gecco Conference,* no. 4, pp. 1301-1308.

Janikow, C.Z. & Michalewicz, Z. 1991, "An Experimental Comparison of Binary and Floating Point Representations in Genetic Algorithm", *Proceedings of the Fourth International Conference on Genetic Algorithms,* pp. 31-36.

Jobson, J.D. & Korkie, B. 1981, "Putting Markowitz Theory to Work", *Journal of Portfolio Management,* vol. 7, no. 4, pp. 70-74.

Joines, J. & Houck, C. 1994, "On the Use of Non-stationary Penalty Functions to Solve Nonlinear Constrained Optimization Problems with Genetic Algorithms", *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence,* vol. 2, pp. 579-584.

Jorian, P. 1992, "Portfolio Optimization in Practice", *Financial Analysts Journal,* no. Jan-Feb, pp. 68-74.

Kallberg, J.G., White, R.W. & Ziemba, W.T. 1982, "Short Term Financial Planning Under Uncertainty", *Management Science,* vol. 28, no. 6, pp. 670-682.

Keerativuttitumrong, N., Chaiyaratana, N. & Varavithya, V. 2002, "Multi-objective Co-operative Co-evolutionary Genetic Algorithm", *Lecture Notes in Computer Science,* no. 2439, pp. 288-297.

Kellerer, H., Mansini, R. & Grazia Speranza, M. 2000, "Selecting Portfolios with Fixed Costs and Minimum Transaction Lots", *Annals of Operations Research,* vol. 99, no. 1-4, pp. 287-304.

Kellerer, H. & Maringer, D. 2001, "Optimization of Cardinality Constrained Portfolios with a Hybrid Local Search Algorithm", *In the Fourth Metaheuristics International Conference.*

Kingdon, J. & Feldman, K. 1995, "Genetic Algorithms and Applications to Finance", *Applied Mathematical Finance,* vol. 2, no. 2, pp. 89.

Kirkpatrick, S., Gelatt, C. & Vecchi, M. 1983, "Optimization by Simulated Annealing", *Science,* vol. 220, no. 4598, pp. 671-680.

Konno, H. 1990, "Piecewise Linear Risk-Function and Portfolio Optimization", *Journal of the Operators Research Society of Japan,* no. 33, pp. 139-156.

Konno, H. & Yamazaki, H. 1991, "Mean-Absolute Deviation Portfolio Optimization Model and its Applications to Tokyo Stock Market", *Management Science,* vol. 37, no. 5, pp. 519-531.

Kusy, M.I. & Ziemba, W.T. 1986, "A Bank Asset and Liability Management Model", *Operations research,* vol. 34, no. 3, pp. 356.

Laguna, M. 2002, "Scatter Search" in *Handbook of Applied Optimization*, eds. P.M. Pardalos & M.G.C. Resende, Oxford University Press, pp. 183-193.

Le Riche, R.G., Knopf-Lenoir, C. & Haftka, R.T. 1995, "A Segregated Genetic Algorithm for Constrained Structural Optimization", *Proceedings of the International Conference on Genetic Algorithms,* no. 6, pp. 558-565.

Lee, M.A. & Takagi, H. 1993, "Dynamic Control of Genetic Algorithms Using Fuzzy Logic Techniques", *Proceedings of the International Conference on Genetic Algorithms,* no. 5, pp. 76-83.

Leibowitz, M.L., Kogelman, S. & Bader, L.N. 1992, "Asset Performance and Surplus Control: A Dual-Shortfall Approach", *Journal of Portfolio Management,* vol. 18, no. 2, pp. 28-37.

Leibowitz, M.L. & Kogelman, S. 1991, "Asset Allocation under Shortfall Constraints", *Journal of Portfolio Management,* vol. 17, no. 2, pp. 18-23.

Leland, H. 1988, "Who Should Buy Portfolio Insurance" in *Portfolio Insurance: a Guide to Dynamic Hedging*, ed. D.L. Luskin, Wiley.

Lin, D., Wang, S. & Yan, H. 2001, "A Multiobjective Genetic Algorithm for Portfolio Selection Problem", *Proceedings of ICOTA 2001*.

Litterman, B. 2003, *Modern Investment Management: An Equilibrium Approach,* John Wiley & Sons Inc.

Liu, J. & Pan, J. 2003, "Dynamic Derivative Strategies", *Journal of Financial Economics,* vol. 69, no. 3, pp. 401-430.

Loraschi, A. & Tettamanzi, A. 1995, "An Evolutionary Algorithm for Portfolio Selection in a Downside Risk Framework" in *Forecasting Financial Markets*, ed. C. Dunis, Wiley, pp. 275-286.

Loraschi, A., Tettamanzi, A., Tomassini, M. & Verda, P. 1995, "Distributed Genetic Algorithms with an Application to Portfolio Selection Problems", pp. 384-387.

Luskin, D.L. 1988, *Portfolio Insurance: A Guide to Dynamic Hedging,* John Wiley & Sons Inc.

Mandelbrot, B. 1963, "The Variation of Certain Speculative Prices", *Journal of Business,* no. 36, pp. 394-419.

Manderick, B. & Spiessens, P. 1989, "Fine-Grained Parallel Genetic Algorithms", *Proceeding of the Third International Conference on Genetic Algorithms,* pp. 428-433.

Maneeratana, K., Boonlong, K. & Chaiyaratana, N. 2004, "Multi-objective Optimisation by Co-operative Co-evolution", *Lecture Notes in Computer Science,* no. 3242, pp. 772-781.

Mansini, R. & Speranza, M.G. 2005, "An Exact Approach for Portfolio Selection with Transaction Costs and Rounds", *IIE Transactions,* vol. 37, no. 10, pp. 919-929.

Mansini, R. & Speranza, M.G. 1999, "Heuristic Algorithms for the Portfolio Selection Problem with Minimum Transaction Lots", *European Journal of Operational Research,* vol. 114, no. 2, pp. 219-233.

Margrabe, W. 1997, "Selecting and Customizing Equity Derivatives for Investment Funds" in *Equity Derivatives Applications in Risk Management and Investment*, ed. L. Donaldson, Risk Books.

Maringer, D. 2005, *Portfolio Management with Heuristic Optimization,* Springer-Verlag New York Inc.

Maringer, D. & Kellerer, H. 2003, "Optimization of Cardinality Constrained Portfolios with a Hybrid Local Search Algorithm", *OR Spectrum,* vol. 25, no. 4, pp. 481-495.

Markowitz, H. 1991, *Portfolio Selection: Efficient Diversification of Investments,* 2nd edn, Blackwell Publishers.

Markowitz, H. 1987, *Mean-Variance Analysis in Portfolio Choice and Capital Markets,* Blackwell Publishers.

Markowitz, H. 1952, "Portfolio Selection", *Journal of Finance,* vol. 7, no. 1, pp. 77-91.

Martí, R., Laguna, M. & Glover, F. 2006, "Principles of Scatter Search", *European Journal of Operational Research,* vol. 169, no. 2, pp. 359-372.

Martin, W.N., Lienig, J. & Cohoon, J.P. 2000, "Island (Migration) Models: Evolutionary Algorithms based on Punctuated Equilibria" in *Evolutionary Computation 2: Advanced Algorithms and Operators*, eds. T. Back, D.B. Fogel & Z. Michalewicz, Institute of Physics Publishing, pp. 101-124.

Meade, N. 1993, "Forecasting the Return and Risk on a Portfolio of Assets", *International Journal of Forecasting,* vol. 9, no. 3, pp. 373-386.

Meade, N. & Salkin, G.R. 2000, "The Selection of Multinational Equity Portfolios: Forecasting Models and Estimation Risk", *European Journal of Finance,* vol. 6, no. 3, pp. 259-279.

Menon, A. 2004, *Frontiers of Evolutionary Computation,* Kluwer Academic Publishers.

Merton, R.C. 1992, *Continuous-time Finance,* Blackwell Publishers.

Merton, R.C. 1971, "Optimum Consumption and Portfolio Rules in a Continuous-Time Model", *Journal of Economic Theory,* vol. 3, no. 4, pp. 373.

Merton, R.C. 1969, "Lifetime Portfolio Selection Under Uncertainty: the Continuous-Time Case", *Review of Economics & Statistics,* vol. 51, no. 3, pp. 247.

Michalewicz, Z. 1995, "Genetic Algorithms, Numerical Optimization and Constraints", *Proceedings of the International Conference on Genetic Algorithms,* no. 6, pp. 151-158.

Michalewicz, Z. 1992, *Genetic Algorithms + Data Structures = Evolution Programs,* Springer-Verlag.

Michalewicz, Z. & Fogel, D.B. 2004, *How to Solve It: Modern Heuristics,* 2nd edn, Springer-Verlag Berlin.

Michalewicz, Z. & Janikow, C.Z. 1996, "GENOCOP: A Genetic Algorithm for Numerical Optimization Problems", *Communications of the ACM,* vol. 39, no. 12, Article No. 175.

Michalewicz, Z. & Janikow, C.Z. 1991, "Handling Constraints in Genetic Algorithms", *Proceedings of the Fourth International Conference on Genetic Algorithms,* pp. 151-157.

Michaud, R.O. 1998, *Efficient Asset Management: A Practical Guide to Stock Portfolio Optimization and Asset Allocation,* Oxford University Press Inc, USA.

Michaud, R.O. 1989, "The Markowitz Optimization Enigma: Is 'Optimized' Optimal?", *Financial Analysts Journal,* vol. 45, no. 1, pp. 31.

Mori, N., Imanishi, S., Kita, H. & Nishikawa, Y. 1997, "Adaptation to Changing Environments by Means of the Memory Based Thermodynamical Genetic Algorithm", *Proceedings of the International Conference on Genetic Algorithms,* no. 7, pp. 299-306.

Moscato, P. 2002, "Memetic Algorithms" in *Handbook of Applied Optimization*, eds. P.M. Pardalos & M.G.C. Resende, Oxford University Press, pp. 157-167.

Moscato, P. & Cotta, C. 2003, "A Gentle Introduction to Memetic Algorithms", *International Series in Operations Research and Management Science,* no. 57, pp. 105-144.

Mossin, J. 1968, "Optimal Multiperiod Portfolio Policies", *Journal of Business,* vol. 41, no. 2, pp. 215-229.

Muhlenbein, H. & Schlierkamp-Voosen, D. 1993, "Predictive Models for the Breeder Genetic Algorithm: Continuous Parameter Optimization", *Evolutionary Computation,* no. 1, pp. 25-49.

Mukerjee, A., Biswas, R., Deb, K. & Mathur, A.P. 2002, "Multi–objective Evolutionary Algorithms for the Risk–return Trade–off in Bank Loan Management", *International Transactions in Operational Research,* vol. 9, no. 5, pp. 583-597.

Mulvey, J.M. & Chen, Z. 1996, *An Empirical Evaluation of the Fixed-Mix Investment Strategy*, Princeton University Report.

Mulvey, J.M. & Thorlacius, A.E. 1995, "The Towers Perrin Global Capital Market Scenario Generation System", *Newton Institute Publications,* no. 10, pp. 286-314.

Mulvey, J.M. & Ziemba, W.T. 1995, "Asset and Liability Management Systems for Long-Term Investors: Discussion of the Issues", *Newton Institute Publications,* no. 10, pp. 3-40.

Mulvey, J.M. 2001, "Introduction to Financial Optimization: Mathematical Programming Special Issue", *Mathematical Programming,* vol. 89, no. 2, pp. 205-216.

Mulvey, J.M., Pauling, W.R. & Madey, R.E. 2003, "Advantages of Multiperiod Portfolio Models", *Journal of Portfolio Management,* vol. 29, no. 2, pp. 35-45.

Mulvey, J.M. & Shetty, B. 2004, "Financial Planning via Multi-Stage Stochastic Optimization", *Computers & Operations Research,* vol. 31, no. 1, pp. 1-20.

Mulvey, J.M. & Vladimirou, H. 1992, "Stochastic Network Programming for Financial Planning Problems", *Management Science,* vol. 38, no. 11, pp. 1642-1664.

Murata, T. & Ishibuchi, H. 1995, "MOGA: Multi-Objective Genetic Algorithms", *Proceedings of the Second International Conference on Evolutionary Computation,* pp. 289-294.

Novkovic, S. 1998, "A Genetic Algorithm Simulation of a Transition Economy: An Application to Insider-Privatisation in Croatia", *Computational Economics,* vol. 11, no. 3, pp. 221-243.

Ong, C.S., Jih-Jeng, H. & Tzeng, G.H. 2005, "A Novel Hybrid Model for Portfolio Selection", *Applied Mathematics and Computation,* vol. 169, no. 2, pp. 1195-1210.

Pardalos, P.M. & Resende, M.G.C. (eds) 2002, *Handbook of Applied Optimization,* Oxford University Press Inc, USA.

Paredis, J. 2000, "Coevolutionary Algorithms" in *Evolutionary Computation 2: Advanced Algorithms and Operators,* eds. T. Back, D.B. Fogel & Z. Michalewicz, Institute of Physics Publishing.

Pareto, V. 1896, *Cours D'Economie Politique,* Lausanne.

Perold, A.F. & Sharpe, W.F. 1995, "Dynamic Strategies for Asset Allocation", *Financial Analysts Journal,* vol. 51, no. 1, pp. 149.

Pettev, C.C. 2000, "Diffusion (Cellular) Models" in *Evolutionary Computation 2: Advanced Algorithms and Operators,* eds. T. Back, D.B. Fogel & Z. Michalewicz, Institute of Physics Publishing, pp. 125-133.

Potter, M.A. & De Jong, K.A. 1994, "A Cooperative Coevolutionary Approach to Function Optimization", *Lecture Notes in Computer Science,* no. 866, pp. 249-259.

Rachev, S.T. (ed) 2004, *Handbook of Computational and Numerical Methods in Finance*, Birkhauser Boston.

Rasmussen, M. 2003, *Quantitative Portfolio Optimisation, Asset Allocation and Risk Management: A Practical Guide to Implementing Quantitative Investment Theory.* Palgrave Macmillan.

Rechenberg, I. 1973, *Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution,* Fromman-Hozlboog Verlag, Stuttgart.

Rechenberg, I. 1965, *Cybernetic Solution Path of an Experimental Problem*, Royal Aircraft Establishment.

Richardson, J.T., Palmer, M.R., Liepins, G.E. & Hilliard, M.R. 1989, "Some Guidelines for Genetic Algorithms with Penalty Functions", *Proceedings of the Third International Conference on Genetic Algorithms,* pp. 191-197.

Rockafellar, R.T. & Uryasev, S. 2000, "Optimization of Conditional Value-at-Risk", *Journal of Risk,* vol. 2, no. 3, pp. 21-42.

Ross, S.A. 1976, "The Arbitrage Theory of Capital Asset Pricing", *Journal of Economic Theory,* vol. 13, no. 3, pp. 341-360.

Roy, A.D. 1952, "Safety-First and the Holding Assets", *Econometrica,* vol. 20, no. 3, pp. 431-449.

Samuelson, P.A. 1969, "Lifetime Portfolio Selection by Dynamic Stochastic Programming", *Review of Economics & Statistics,* vol. 51, no. 3, pp. 239.

Satchell, S.E. & Scowcroft, A. (eds) 2003, *Advances in Portfolio Construction and Implementation*, Butterworth-Heinemann Ltd.

Schaffer, J.D. 1985, "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms", *Proceedings of the First International Conference on Genetic Algorithms,* pp. 93-100.

Schaffer, J.D., Caruna, R.A., Eshelman, L.J. & Das, R. 1989, "A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization", *Proceedings of the Third International Conference on Genetic Algorithms,* pp. 51-60.

Schaffer, J.D. & Eshelman, L.J. 1991, "On Crossover as an Evolutionarily Viable Strategy", *Proceedings of the Fourth International Conference on Genetic Algorithms,* pp. 61-68.

Scherer, B. 2004, *Portfolio Construction and Risk Budgeting,* 2nd edn, Risk Books.

Scherer, B. (ed) 2003, *Asset and Liability Management Tools*, Risk Books.

Scherer, B. & Martin, R.D. 2005, *Modern Portfolio Optimization,* Springer-Verlag New York Inc.

Schlottmann, F. & Seese, D. 2004, "A Hybrid Heuristic Approach to Discrete Multi-Objective Optimization of Credit Portfolios", *Computational Statistics and Data Analysis,* vol. 47, no. 2, pp. 373-399.

Schwefel, H.P. 1981, *Numerical Optimization of Computer Models,* John Wiley and Sons Ltd.

Schwefel, H.P. 1977, *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie,* Birkhaeuser.

Schwefel, H.P. 1965, *Kybernetische Evolution als Strategie der Experimentellen Forschung in der Stiiimungstechnik*, Technical University of Berlin.

Sharpe, W.F. 1981, "Decentralized Investment Management", *Journal of Finance,* vol. 36, no. 2, pp. 217.

Sharpe, W.F. 1964, "Capital Asset Prices: a Theory of Market Equilibrium Under Conditions of Risk", *Journal of Finance,* vol. 19, no. 3, pp. 425-442.

Sharpe, W.F. & Tint, L.G. 1990, "Liabilities - A new approach", *Journal of Portfolio Management,* vol. 16, no. 2, pp. 5-10.

Skolicki, Z. & De Jong, K. 2005, "The Influence of Migration Sizes and Intervals on Island Models", *Gecco Conference,* vol. 7, no. 2, pp. 1295-1302.

Skolicki, Z. & De Jong, K. 2004, "Improving Evolutionary Algorithms with Multi-representation Island Models", *Lecture Notes in Computer Science,* no. 3242, pp. 420-429.

Sortino, F.A. 2001, *Managing Downside Risk in Financial Markets,* Butterworth-Heinemann Ltd.

Sortino, F.A. & Forsey, H.J. 1996, "On the Use and Misuse of Downside Risk", *Journal of Portfolio Management,* vol. 22, no. 2, pp. 35-43.

Speranza, M.G. 1996, "A Heuristic Algorithm for a Portfolio Optimization Model Applied to the Milan Stock Market", *Computers & Operations Research,* vol. 23, no. 5, pp. 433-442.

Srinivas, N. & Deb, K. 1995, "Multi-Objective Function Optimization using Non-Dominated Sorting Genetic Algorithms", *Evolutionary Computation,* vol. 2, no. 3, pp. 221-248.

Streichert, F. & Tanaka-Yamawaki, M. 2006, "The Effect of Local Search on the Constrained Portfolio Selection Problem", *2006 IEEE Congress on Evolutionary Computation,* pp. 2368-2374.

Streichert, F., Ulmer, H. & Zell, A. 2005, "Parallelization of Multi-objective Evolutionary Algorithms Using Clustering Algorithms", *Lecture Notes in Computer Science,* no. 3410, pp. 92-107.

Streichert, F., Ulmer, H. & Zell, A. 2004, "Evaluating a Hybrid Encoding and Three Crossover Operators on the Constrained Portfolio Selection Problem", *Proceedings of the Congress on Evolutionary Computation,* vol. 1, pp. 932-939.

Streichert, F., Ulmer, H. & Zell, A. 2003, "Evolutionary Algorithms and the Cardinality Constrained Portfolio Optimization Problem", *Operations Research Proceedings,* pp. 253-260.

Subbu, R., Bonissone, P.P., Eklund, N., Bollapragada, S. & Chalermkraivuth, K. 2005, "Multiobjective Financial Portfolio Design: A Hybrid Evolutionary Approach", *IEEE International Conference on Evolutionary Computation Proceedings,* vol. 2, pp. 1722-1729.

Surry, P.D. & Radcliffe, N.J. 1996, "Inoculation to Initialise Evolutionary Search", *Lecture Notes in Computer Science,* no. 1143, pp. 269-285.

Tan, K.C., Chew, Y.H., Lee, T.H. & Yang, Y. 2003, "A Cooperative Coevolutionary Algorithm for Multiobjective Optimization", *IEEE International Conference on Systems Man and Cybernetics,* vol. 1, pp. 390-395.

Tan, K.C., Khor, E.F. & Lee, T.H. 2005, *Multiobjective Evolutionary Algorithms and Applications,* Springer-Verlag London Ltd.

Tan, K.C., Lee, T.H. & Khor, E.F. 1999, "Evolutionary Algorithms with Goal and Priority Information for Multi-Objective Optimization", *Proceedings of the Congress on Evolutionary Computation,* vol. 1, pp. 106-116.

Tang, W. & Wang, Y. 2004, "Intelligent Method for Dynamic Portfolio Selection with Probability Criterion", *IEEE International Conference on Systems Man and Cybernetics,* vol. 4, pp. 3323-3327.

Thaler, R.H., Tversky, A., Kahneman, D. & Schwartz, A. 1997, "The Effect of Myopia and Loss Aversion on Risk Taking: an Experimental Test", *Quarterly Journal of Economics,* vol. 112, no. 2, pp. 647-661.

Trippi, R.R. & Harriff, R.B. 1991, "Dynamic Asset Allocation Rules: Survey and synthesis", *Journal of Portfolio Management,* vol. 17, no. 4, pp. 19-26.

Tsoi, E., Wang, K.P. & Fung, C.C. 1995, "Hybrid GA/SA Algorithms for Evaluating Trade-Off Between Economic Cost and Environmental Impact in Generation Dispatch", *IEEE International Conference on Evolutionary Computation,* vol. 1, pp. 132-137.

Van Veldhuizen, D.A., Zydallis, J.B. & Lamont, G.B. 2003, "Considerations in Engineering Parallel Multiobjective Evolutionary Algorithms", *IEEE Transactions on Evolutionary Computation,* vol. 7, no. 2, pp. 144-173.

Venugopal, M.S., Subramanian, S. & Rao, U.S. 2004, "Usefulness of Genetic Algorithm Model for Dynamic Portfolio Selection", *Journal of Financial Management & Analysis,* vol. 17, no. 1, pp. 45-53.

Whitley, D. 2001, "An Overview of Evolutionary Algorithms: Practical Issues and Common Pitfalls", *Information and Software Technology,* vol. 43, no. 14, pp. 817-831.

Whitley, D. 1989, "The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best", *Proceedings of the Third International Conference on Genetic Algorithms,* pp. 116-123.

Whitley, D., Gordon, V.S. & Mathias, K. 1994, "Lamarckian Evolution, The Baldwin Effect and Function Optimization", *Lecture Notes in Computer Science,* no. 866, pp. 6-15.

Whitley, D., Rana, S. & Heckendorn, R.B. 1999, "The Island Model Genetic Algorithm: On Separability, Population Size and Convergence", *Journal of Computing and Information Technology,* vol. 7, no. 1, pp. 33-48.

Wilding, T. 2003, "Using Genetic Algorithms to Construct Portfolios" in *Advances in Portfolio Construction and Implementation*, eds. A. Scowcroft & S.E. Satchell, pp. 135-160.

Wright, S. 1964, "Stochastic Processes in Evolution" in *Stochastic Models in Medicine and Biology*, ed. J. Gurland, University of Wisconsin Press, Springer.

Xiao, N. & Armstrong, M.P. 2003, "A Specialized Island Model and Its Application in Multiobjective Optimization", *Lecture Notes in Computer Science,* no. 2724, pp. 1530-1540.

Yagiura, M. & Ibaraki, T. 2002, "Local Search" in *Handbook of Applied Optimization*, eds. P.M. Pardalos & M.G.C. Resende, Oxford University Press, pp. 104-122.

Yoshimoto, A. 1996, "The Mean-Variance Approach to Portfolio Optimization Subject to Trading Costs", *Operations Research Society of Japan Journal,* vol. 39, no. 1, pp. 99-117.

Young, M.R. 1998, "A Minimax Portfolio Selection Rule with Linear Programming Solution", *Management Science,* vol. 44, no. 5, pp. 673-683.

Ziemba, W.T., Mulvey, J.M. & Moffatt, H.K. (eds) 1998, *Worldwide Asset and Liability Modeling*, Cambridge University Press.

Zitzler, E., Deb, K. & Thiele, L. 2000, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results", *Evolutionary computation,* vol. 8, no. 2, pp. 173-196.

Zitzler, E., Laumanns, M. & Thiele, L. 2001, *Improving the Strength Pareto Evolutionary Algorithm*, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, Zurich.

Zitzler, E. & Thiele, L. 1998, "Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study", *Lecture Notes in Computer Science,* vol. 1498, pp. 292-304.

LONDON UNIV.