Imperial College London
Department of Computing

MEng Computing (Software Engineering)

Robust Portfolio Optimization Using Conditional Value At Risk

Final Report

by

Wei Ning Cho (wnc04)

Project Supervisor: Professor Berç Rustem

June 22, 2008

Abstract

In this report, we propose a worst-case robust multi-period portfolio optimization model using conditional value at risk. We use a min-max algorithm and an optimization framework based on scenario trees. The min-max formulation gives the investor a portfolio that is optimal for the worst-case scenario and performance is guaranteed to improve if the worst-case does not happen. This feature protects the investor from errors that arise from uncertainties in the expected returns values for the assets. Numerical experiments backtesting the optimization strategies at different risk levels are reported. We also investigate the performance differences between interior point solvers and simplex solvers.

Acknowledgements

I would like to thank my project supervisor Professor Berç Rustem for all the help and motivation he has given during my project. I would also like to thank my second marker, Professor Daniel Kuhn for the useful suggestions and tips.

Contents

Chapter 1

Introduction

Asset allocation is a problem faced by every investor. When making investment decisions, an investor has to seek a balance between risk and returns. In 1952, Harry Markowitz published his seminal work on portfolio selection[17], in which he established a framework for investment decisions. In the single-period Markowitz model, the investor maximizes the expected return of the portfolio and minimizes the risk, measured by the variance of portfolio returns.

Using the portfolio variance as the risk measure has its limitations. The variance is a symmetrical measure that does not take into consideration the direction of movement. An asset experiencing better than expected returns is deemed to be as risky as an asset that is suffering from lower than expected returns. To address this issue, alternative risk measures such as Value at Risk and Conditional Value at Risk have been introduced to replace the variance.

In the original Markowitz model, the investment problem is taken to be a single-period problem. In reality, investments are multi-period, with adjustments made to the portfolio allocation periodically to optimize the performance of the portfolio. Furthermore, an investor needs to pay a fee when buying or selling a stock, so we need to incorporate transaction costs in the optimization model.

Multi-period portfolio optimization problems entail the construction of a scenario tree to forecast expected future prices. There is an inherent uncertainty in this forecast which would affect the optimization results. To overcome this uncertainty, we want a robust strategy that will take into account rival scenarios, and produce a strategy that guarantees performance in view of all rival scenarios.

In this report, we propose and implement a multi-period worst-case robust portfolio optimization model based on conditional value at risk.

1.1 Key Contributions

- We suggest a multi-period portfolio optimization framework that maximizes returns and minimizes the portfolio conditional value at risk.
- We extend the formulation to provide a worst-case robust optimal strategy given rival forecast scenarios. (See Section 3)
- We implement the worst-case robust mean-conditional value at risk portfolio optimization model, and analyze the properties of the robust portfolio. (See Section 4)
- We evaluate the performance of the model using historical data, and present the results of the backtesting. We compare the performance of the mean-conditional value at risk model against the classical mean-variance model. We also compare the performance of the worst-case robust portfolio optimization model against the non-robust model. (See Section 5)
- We compare the performance of interior point solvers against simplex solvers.

Chapter 2

Background

In this chapter, we will highlight the current state of the art, and introduce key concepts used throughout this report. Section 2.1 introduces the portfolio optimization problem. Section 2.2 and 2.3 introduces the value at risk and conditional value at risk measures respectively. Finally, in Section 2.4 we look at multi-period extensions to the portfolio optimization problem.

2.1 Portfolio Optimization

Before introducing the portfolio optimization model, we first introduce some of the terminology that will be used.

2.1.1 Terminology

Suppose that there are n assets with random rates of returns $\xi_1, \xi_2, ..., \xi_n$. The expected rates of returns are $E[\xi_1], E[\xi_2], ..., E[\xi_n]$.

Assuming we have a portfolio that consists of n assets, and w_i is the weight of asset i in the portfolio, such that $\sum_{i=1}^{n} w_i = 1$.

The portfolio returns is thus $\sum_{i=1}^{n} w_i \xi_i$ and the expected portfolio return is $\sum_{i=1}^{n} w_i E[\xi_i]$.

Risk

Risk is usually identified as the variance of the portfolio, σ^2. The variance of a random variable is its second central moment, and its mathematical definition is

$$
\begin{aligned}
\sigma^2 &= E[(\xi - \overline{\xi})^2] \\
&= E(\xi^2) - 2E(\xi)\overline{\xi} + \overline{\xi}^2 \\
&= E(\xi^2) - (\overline{\xi})^2
\end{aligned}
$$

Given the variance of individual assets, the variance of the portfolio can be calculated using the covariance between asset i and asset j - σ_{ij}.

Let σ_i^2 and σ_p^2 be the variance of asset i and the portfolio, respectively. The variance of the portfolio can be calculated as shown below:

$$
\begin{aligned}
\sigma_p^2 &= E[(\xi_p - \bar{\xi}_p)^2] \\
&= E\Big[\Big(\sum_{i=1}^{n} w_i\xi_i - \sum_{i=1}^{n} w_i\bar{\xi}_i\Big)^2\Big] \\
&= E\Big[\Big(\sum_{i=1}^{n} w_i(\xi_i - \bar{\xi}_i)\Big)^2\Big] \\
&= E\Big[\Big(\sum_{i=1}^{n} w_i(\xi_i - \bar{\xi}_i)\Big)\Big(\sum_{i=0}^{n} w_i(\xi_i - \bar{\xi}_i)\Big)\Big] \\
&= E\Big[\Big(\sum_{i,j=1}^{n} w_i w_j(\xi_i - \bar{\xi}_i)(\xi_j - \bar{\xi}_j)\Big)^2\Big] \\
&= \sum_{i,j=1}^{n} w_i w_j \sigma_{ij}
\end{aligned}
$$

Alternatively, let $\mathbf{W}$ be the vector representing the weights of each asset, P be the current portfolio position, and Σ be the variance-covariance matrix. The variance of the portfolio can also be written as

$$\sigma_p^2 = \mathbf{W^T \Sigma W} \tag{2.1}$$

2.1.2 Maximum Expected Returns Problem

For an investor that wishes to attain the maximum expected return, the following problem maximizes the expected return of the portfolio.

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} w_i E[\xi_i] \\ \text{subject to} \quad & \sum_{i=1}^{n} w_i = 1 \end{aligned} \tag{2.2}$$

2.1.3 Minimum Variance Portfolio

Maximizing the returns of the portfolio might sound tempting, but the model given in Equation 2.2 does not take into account the risk of the portfolio. Instead of maximizing the returns of the portfolio,risk-averse investors might want to minimize the risk (variance) of the portfolio instead. The following problem gives the portfolio with the minimum variance.

$$\begin{aligned} \text{minimize} \quad & \sum_{i,j=1}^{n} w_i w_j \sigma_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{n} w_i = 1 \end{aligned}$$

2.1.4 Mean-Variance Efficient Portfolio

The maximum returns portfolio and minimum variance portfolio give the two extremes of returns and risk. Most investors would like to strike a balance between the maximum returns and minimum variance portfolio. The classical mean-variance portfolio optimization [17] can be formulated as in Equation 2.3.

$$\begin{aligned} \text{minimize} \quad & \sum_{i,j=1}^{n} w_i w_j \sigma_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{n} w_i = 1 \\ & \sum_{i=1}^{n} w_i E[\xi_i] \geq R \end{aligned} \tag{2.3}$$

The formulation in Equation 2.3 will give the optimum portfolio (i.e. minimum risk) for the specified minimum required wealth, R. That is, no other portfolios can give a similar or higher return with a lower risk. Alternatively, we could also formulate the problem to maximize returns given a specified risk level σ, as shown in Equation 2.4.

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{n} w_i E[\xi_i] \\
\text{subject to} \quad & \sum_{i=1}^{n} w_i = 1 \\
& \sum_{i,j=1}^{n} w_i w_j \sigma_{ij} \leq \sigma
\end{aligned}
\tag{2.4}
$$

2.1.5 Efficient Frontier

By finding all the optimum portfolios and plotting them on a risk-returns graph, we obtain the efficient frontier (also known as the Markowitz Frontier). The portfolios that lie on the efficient frontier are the portfolios that give the maximum returns for a given level of risk.

To obtain the efficient frontier, we first solve the minimum-risk problem ignoring any returns constraints, thus obtaining a minimum expected return W_{min}. We then solve the maximum-returns problem, ignoring any constraints on the level of risk, thus obtaining a maximum expected return $\mathcal{W}_{max}$. We can then solve the minimum-risk problem for a set of points $\mathcal{W} \in [\mathcal{W}_{min}, \mathcal{W}_{max}]$. The resulting set of points thus from the efficient frontier. Figure 2.1 shows an example of an efficient frontier.

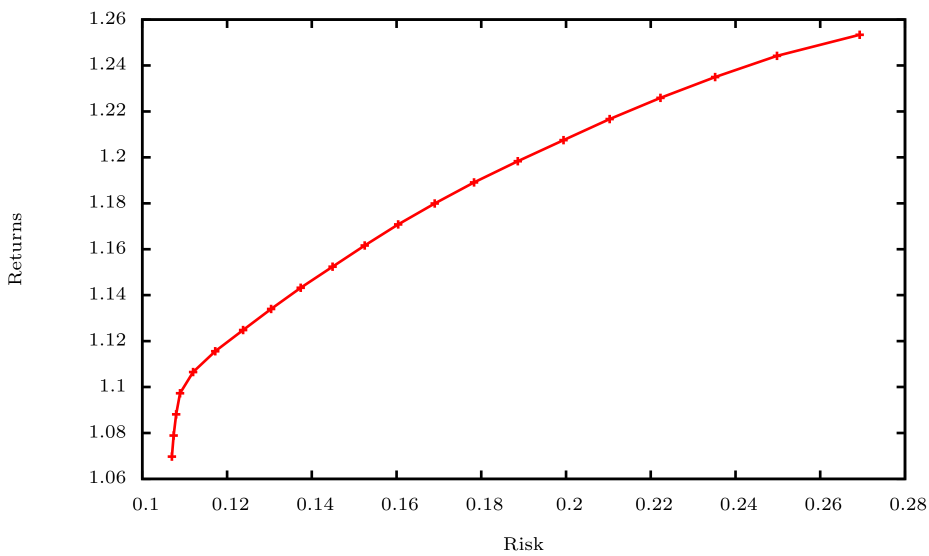

Figure 2.1: Efficient Frontier

2.2 Value at Risk

In the mean-variance portfolio optimization model in [17], Markowitz uses the portfolio variance as the risk measure. The variance is a measure of statistical dispersion that gives us the average of the squared distance of the possible values from the expected value. It treats both upside and downside payoffs symmetrically. However, while most investors will be disturbed by a drop in prices, most investors will not mind an increase in price.

One alternative measure of risk is the Value at Risk. In [21], Value at Risk is defined as the difference between the corresponding percentile of the profit and loss distribution and the current value of the portfolio. That is, it measures the worst expected loss over a given time horizon at a given confidence level. In [24], value at risk is defined mathematically as in Equation 2.5.

$$\begin{aligned} VaR &= \zeta_\alpha(\xi) \\ &= inf\{\zeta | P(\xi \leq \zeta) \geq \alpha\} \end{aligned} \tag{2.5}$$

2.2.1 Calculating the Value at Risk

Common ways of calculating the value at risk[13, 21, 12, 4] include:

- Variance-Covariance
- Historical Simulation
- Monte Carlo Simulation

Variance-Covariance Method

The release of RiskMetrics[13, 21] by J.P.Morgan has made the variance-covariance method of calculating value at risk popular.

In this section, we briefly describe how the variance-covariance method is used to calculate the value at risk. For details and proof of the method, please read [13, 21].

As suggested by its name, the value at risk can be calculated using the variance-covariance matrix, σ.

When using the variance-covariance method, we make the assumption that the returns of the assets are normally distributed. Therefore, the profit and loss distribution is normally distributed with mean 0 and variance $\mathbf{W^T \Sigma W}$ (see equation 2.1). In other words,

$$P\&L \sim N(0, \mathbf{W^T \Sigma W})$$

Since the value at risk is a percentile of the profit and loss distribution, which is a normal distribution, we can thus calculate the $\alpha\%$ value at risk as

$$VaR = -z_{(1-\alpha)}\sqrt{\mathbf{W^T \Sigma W}}$$

where z_k is the corresponding percentile of the standard normal distribution.

Historical Simulation

Instead of trying to specify the distributions of the portfolio return, we can use historical data to simulate the shape of the distribution. Therefore, we do not need to assume normality of the returns distribution when using historical simulation. Given sufficient amount of historical data, this method can give a realistic distribution as it will also take into account major market events such as stock market crashes.

However, the downside to historical simulation is that it assumes the distribution of returns in the future is similar to that in the past, which might not necessarily be true. One way of mitigating this is by updating the historical data to reflect the difference between the historical volatility of the market variable and its current volatility [10].

Once we have obtained a database of historical data, we will be able to calculate the profit and loss. As an example, Table 2.1 shows the prices for an asset, along with the calculated profit and loss.

Having generated the n profit and loss scenarios, we then arrange them in descending order, denoting them by $\Delta V_1, \Delta V_2, ..., \Delta V_n$.

The α-VaR can thus be defined as in Equation 2.6.

$$\zeta_\alpha = -\Delta V_{n\alpha} \tag{2.6}$$

Using the data in Table 2.1, the profit and loss scenarios can thus be arranged as shown in Table 2.2, and the 95%-value at risk is $-\Delta V_9$ (see Equation 2.6), which is £2.

Date	Opening Price (£)	Closing Price (£)	Profit & Loss (£)
1/1/2008	100	95	-2
2/1/2008	95	98	3
3/1/2008	98	102	4
4/1/2008	102	101	-1
5/1/2008	101	98	-3
6/1/2008	98	96	-2
7/1/2008	96	97	1
8/1/2008	97	102	5
9/1/2008	102	103	1
10/1/2008	103	101	-2

Table 2.1: Prices and Calculation of Profit & Loss

ΔV_1	ΔV_2	ΔV_3	ΔV_4	ΔV_5	ΔV_6	ΔV_7	ΔV_8	ΔV_9	ΔV_{10}
5	4	3	1	1	-1	-2	-2	-2	-3

Table 2.2: Profit and Loss Scenarios

Monte Carlo Simulation

The Monte Carlo simulation method involves randomly generating scenarios based on parameters obtained from historical data.

After generating these scenarios, we then proceed to calculate the profit and loss, followed by the value at risk in a similar fashion as described in the previous section about historical simulation.

2.2.2 Properties of Value at Risk

In [1], it has been shown that Value at Risk is not a coherent measure of risk (See [1] for the definition of coherency).

Value as risk was rejected as a measure of risk because

- value at risk does not behave nicely with respect to the addition of risks, even independent ones, thereby creating severe aggregation problems.

- the use of value at risk does not encourage and, indeed, sometimes prohibit diversification.

Furthermore, its non-convexity means the Value at Risk has multiple local minima. This makes optimization hard as we will need to find the global minimum. Therefore, we shall consider another risk measure - conditional value at risk.

2.3 Conditional Value at Risk

An alternative measure of loss is Conditional Value at Risk - the conditional expected value of loss, under the condition that it exceeds the value at risk.

In [22], the Conditional Value at Risk is defined mathematically as

$$CVaR_\alpha(w,\zeta) = \zeta + (1-\alpha)^{-1}\int_{\xi\in\mathbb{R}} [f(w,\xi)-\zeta]^+ p(\xi)d\xi \tag{2.7}$$

Alternatively, it can also be formulated as

$$\begin{aligned} CVaR_\alpha(w,\zeta) &= E[\xi|\xi \geq \zeta_\alpha(\xi)] \\ &= E[\xi|\xi \geq VaR] \end{aligned} \tag{2.8}$$

2.3.1 Calculating the Conditional Value at Risk

In [22], it has been shown that Formula 2.7 can be approximated using scenarios.

$$\begin{aligned} CVaR_\alpha(w,\zeta) &= \zeta + (1-\alpha)^{-1}\int_{\xi\in\mathbb{R}} [f(w,\xi)-\zeta]^+ p(\xi)d\xi \\ &\approx \zeta + (1-\alpha)^{1}\sum_{s=1}^{S} [f(w,\xi_s)-\zeta]^+ p_s \end{aligned}$$

$[f(w,\xi_s)-\zeta]^+$ can be reduced to the following

$$\begin{aligned} &[f(w,\xi_s)-\zeta]^+ = z_s \\ &z_s \geq f(w,\xi_s)-\zeta \\ &z_s \geq 0 \end{aligned}$$

where $f(w,\xi_s)$ is the loss function of the portfolio.

Therefore, the problem of minimizing the Conditional Value at Risk can thus be formulated as the following linear programming problem

$$\begin{aligned} \text{minimize} \quad & \zeta + (1-\alpha)^{-1}\sum_{s=1}^{S} z_s p_s \\ \text{subject to} \quad & z_s \geq f(w,\xi_s)-\zeta \\ & z_s \geq 0 \end{aligned} \tag{2.9}$$

2.3.2 Properties of Conditional Value at Risk

Unlike Value at Risk, the Conditional Value at Risk has been shown to be a coherent measure of risk [1, 20, 22, 23].

By the definitions given in Equation 2.8, it can be seen that CVaR $\geq \zeta$, and thus it is a more conservative risk measure. If we can find a portfolio with a low conditional value at risk, then it will also have a low value at risk.

2.3.3 Mean-CVaR Efficient Portfolio

The advantage of the Conditional Value at Risk minimization technique described in Equation 2.9 is that we can formulate the mean-conditional value at risk portfolio optimization problem as a linear programming problem, as shown in Equation 2.10.

$$
\begin{aligned}
\text{minimize} \quad & \zeta + (1-\alpha)^{1} \sum_{s=1}^{S} z_s p_s \\
\text{subject to} \quad & \sum_{i=1}^{n} w_i = 1 \\
& \sum_{i=0}^{n} w_i E[\xi_i] \geq \mathrm{R} \\
& z_s \geq f(w, \xi_s) - \zeta \\
& z_s \geq 0
\end{aligned} \tag{2.10}
$$

An implementation and discussion of a mean-conditional value at risk optimized portfolio can be found in [18].

2.4 Multi-period Portfolio Optimization

The classical portfolio optimization problem can be extended to multistage programming.

The goal of multi-period portfolio optimization is to determine the optimal portfolio for a given finite investment horizon. After making the initial investment at time t = 0, the portfolio might be restructured at times t =1, 2, ... T-1, and redeemed at the end of the period (t = T).

2.4.1 Generating Future Prices

To perform multi-period portfolio optimization, we need to be able to generate future prices at various time periods for the assets in the portfolio. This can be achieved by using scenario trees.

The future prices are approximated by a discrete set of scenarios, or sequence of events. Given an event history up to a particular time, the uncertainty in the next period is characterized by finitely many possible outcomes for the next observation.

Let $\mathcal{N}$ be the set of nodes in the scenario tree. Each node of the scenario tree is denoted by $\mathbf{e} = (s, t)$ where s is a scenario (path from root to leaf), and time period t specifies a particular node on that path. The root node, $\mathcal{N}_{\mathbf{0}} = (s, 0)$, in the scenario tree represents the current time period, and the prices of the assets at the root node are readily observable. The ancestor of event $\mathbf{e}$ is denoted $a(e) = (s, t-1)$.

The nodes that branch from a node gives realizations of the asset prices in the next time period. A probability $P_{\mathbf{e}}$ is attached to each branch. The leafs of the tree represent the realization of the asset prices after T time periods, where T is the depth of the scenario tree. The set of unique paths from the root node to the leaf nodes represent the set of scenario, and the cumulative probability of each branch in the scenario represents the probability of that scenario. Hence, the set of scenarios correspond to the leaf nodes of the scenario tree, $\mathcal{N}_{\mathcal{T}}$.

Generating Scenario Trees

Various techniques for generating scenario trees have been suggested. In [9], an optimization-based scenario tree generation technique is described. This technique involves generating scenario trees that satisfy specified statistical properties. In [15], a criticism of this technique is given. It was argued that matching statistical properties is not sufficient, and arbitrage constraints need to be excluded to obtain realistic outcomes.

In [8], a simulation and randomized clustering approach to scenario tree generation is described. Each node of the scenario tree contains a cluster of scenarios, one of which is designated as the centroid. The final tree consists of the centroids of each node, and their branching probabilities. Instead of a detailed clustering algorithm, a randomized clustering algorithm is used.

2.4.2 Transaction Costs

In a multi-period model where investors are allowed to modify the composition of their portfolio during intermediate time periods, it is essential that transaction costs are taken into account when developing the model. The effect of transaction costs on the mean-variance model has been investigated in [19, 16, 2].

Without transaction costs, the portfolio balance can be calculated using Equation 2.11.

$$\mathbf{w_t} = \xi_t \mathbf{w_{t-1}} + \mathbf{b_t} - \mathbf{s_t} \tag{2.11}$$

where $\mathbf{w_t}$ is the wealth at time t, $\mathbf{b_t}$ are the assets bought in time t, and $\mathbf{s_t}$ are the assets sold in time t.

Taking transaction costs into account, the portfolio balance can then be calculated using Equation 2.12.

$$\mathbf{w_t} = \xi_t \mathbf{w_{t-1}} + (1 - c_b)\mathbf{b_t} - (1 + c_s)\mathbf{s_t} \tag{2.12}$$

where c_b is the unit transaction cost to buy assets, and c_s is the unit transaction cost to sell assets.

2.4.3 Multi-period Mean-Variance Efficient Portfolio

A multistage extension of the mean-variance optimization problem is considered in [7]. The multi-period Mean-Variance portfolio optimization problem can be formulated as below.

$$\begin{array}{rl}
\text{minimize} & \displaystyle\sum_{i,j=1}^{n} w_i w_j \sigma_{ij} \\
\text{subject to} & \mathbf{P} + (1-c)\mathbf{b_0} - (1+c)\mathbf{s_0} = \mathbf{w_0} \\
 & \mathbf{1'b_0} - \mathbf{1's_0} = \mathbf{1} - \mathbf{1'P} \\
\forall \mathbf{e} \in \mathcal{N}_I & \xi_\mathbf{e} \mathbf{w_{a(e)}} + (1-c)\mathbf{b_e} - (1+c)\mathbf{s_e} = \mathbf{w_e} \\
\forall \mathbf{e} \in \mathcal{N}_I & \mathbf{1'b_e} - \mathbf{1's_e} = \mathbf{0} \\
 & \displaystyle\sum_{\mathbf{e} \in \mathcal{N}_T} p_\mathbf{e} \xi_\mathbf{e} \mathbf{w_{a(e)}} \geq R \\
\forall \mathbf{e} \in \mathcal{N} & \mathbf{w^L} \leq \mathbf{w_e} \leq \mathbf{w^U} \\
\forall \mathbf{e} \in \mathcal{N}_I \cup \mathcal{N}_0 & \mathbf{0} \leq \mathbf{b_e} \leq \mathbf{b_e^U} \\
\forall \mathbf{e} \in \mathcal{N}_I \cup \mathcal{N}_0 & \mathbf{0} \leq \mathbf{s_e} \leq \mathbf{s_e}^U
\end{array}$$

where $\mathbf{w^U}$ and $\mathbf{w^L}$ are the upper and lower bounds for the portfolio balance at time period t, $\mathbf{b_e}^U$, $\mathbf{s_e}^U$ are the maximum assets that the investor is allowed to buy/sell in each time period, and $\mathbf{P}$ is the initial portfolio.

2.4.4 Multi-period Mean-CVaR Efficient Portfolio

The multi-period mean-conditional value at risk model is described and implemented in [14]. The model is as follows:

$$
\begin{array}{rl}
\text{minimize} & \zeta + \dfrac{1}{1-\alpha} \displaystyle\sum_{\mathbf{e} \in \mathcal{N}_T} p_{\mathbf{e}} z_{\mathbf{e}} \\
\text{subject to} & \mathbf{P} + (1-c)\mathbf{b_0} - (1+c)\mathbf{s_0} = \mathbf{w_0} \\
 & \mathbf{1}'\mathbf{b_0} - \mathbf{1}'\mathbf{s_0} = \mathbf{1} - \mathbf{1}'\mathbf{P} \\
\forall \mathbf{e} \in \mathcal{N}_I & \xi_{\mathbf{e}} \mathbf{w_{a(e)}} + (1-c)\mathbf{b_e} - (1+c)\mathbf{s_e} = \mathbf{w_e} \\
\forall \mathbf{e} \in \mathcal{N}_I & \mathbf{1}'\mathbf{b_e} - \mathbf{1}'\mathbf{s_e} = \mathbf{0} \\
\forall \mathbf{e} \in \mathcal{N}_T & z_{\mathbf{e}} \geq W_0 - W_0(1'\mathbf{w_e}) - \zeta \\
\forall \mathbf{e} \in \mathcal{N}_T & z_{\mathbf{e}} \geq 0 \\
 & \displaystyle\sum_{\mathbf{e} \in \mathcal{N}_T} p_{\mathbf{e}} \xi_{\mathbf{e}} \mathbf{w_{a(e)}} \geq R \\
\forall \mathbf{e} \in \mathcal{N} & \mathbf{w^L} \leq \mathbf{w_e} \leq \mathbf{w^U} \\
\forall \mathbf{e} \in \mathcal{N}_I \cup \mathcal{N}_0 & \mathbf{0} \leq \mathbf{b_e} \leq \mathbf{b_e^U} \\
\forall \mathbf{e} \in \mathcal{N}_I \cup \mathcal{N}_0 & \mathbf{0} \leq \mathbf{s_e} \leq \mathbf{s_e}^U
\end{array}
$$

where W_0 is the initial wealth of the portfolio.

Chapter 3

Robust Portfolio Optimization

The expected return and risk of the portfolio plays a key role in the optimization problem. These values are inherently uncertain, and investors are often presented with multiple rival forecasts. Choosing between these forecasts is not a trivial matter. [6] introduces a robust mean-variance formulation that guarantees performance in view of all rival scenarios. A worst-case value at risk and robust portfolio optimization technique using a conic programming approach is proposed in [5]. An estimation-free, robust Conditional Value at Risk portfolio allocation model is described in [11]. However, the techniques described in both [5] and [11] only apply to a single period model. We wish to formulate a model that is robust over a multi-period model, taking transaction costs into account.

We assume that the investor is presented with rival scenario trees, and all these rival scenarios are equally likely and the investor is not able to rule out any of them. Given these assumptions, a rational investment strategy should incorporate the information from all these rival scenarios. We propose a worst-case robust optimal strategy that ensures optimal worst-case performance, and will improve if the worst-case scenarios do not materialize.

3.1 Min-Max Optimization

We can extend the model described in Section 2.4.4 to worst-case design with multiple rival scenario trees. The robustness of the strategy can be guaranteed by using a min-max formulation.

Let $\mathcal{K}$ be the set of scenario trees, and $\mathrm{CVaR}^k_\alpha(w, \zeta)$ be the conditional value at risk for scenario tree $k \in \mathcal{K}$. The worst-case conditional value at risk is thus the maximum of these, and can be formulated as

$$\text{Worst Case CVaR}_\alpha = \max\{\text{CVaR}_\alpha^k(w,\zeta)\} \tag{3.1}$$

The problem of minimizing the worst-case conditional value at risk can thus be formulated as

$$\min \max \{\text{CVaR}_\alpha^k(w,\zeta)\} \tag{3.2}$$

This is equivalent to

$$\begin{aligned} &\text{minimize} && v && (3.3)\\ &\text{subject to} && v \geq \text{CVaR}_\alpha^k(w,\zeta) \quad \forall k \in \mathcal{K} && (3.4) \end{aligned}$$

3.2 Problem Formulation

Using the min-max optimization technique described above, we can now formulate the worst-case robust multi-period mean-conditional value at risk portfolio optimization problem.

$$\begin{aligned}
\text{minimize} \quad & \text{WCVaR}_\alpha \\
\text{subject to} \quad & \mathbf{P} + (1-c_b)\mathbf{b_0} - (1+c_s)\mathbf{s_0} = \mathbf{w_0} \\
& \mathbf{1'b_0} - \mathbf{1's_0} = \mathbf{1} - \mathbf{1'P} \\
\forall k \in \mathcal{K} \quad & \text{WCVaR}_\alpha \geq \zeta + \frac{1}{1-\alpha} \sum_{\mathbf{e}\in\mathcal{N}_T^k} p_\mathbf{e} z_\mathbf{e} \\
\forall k \in \mathcal{K}, \forall \mathbf{e} \in \mathcal{N}_I^k \quad & \mathbf{1'b_e} - \mathbf{1's_e} = 0 \\
\forall k \in \mathcal{K}, \forall \mathbf{e} \in \mathcal{N}_I^k \quad & \xi_\mathbf{e} \mathbf{w_{a(e)}} + (1-c_b)\mathbf{b_e} - (1+c_s)\mathbf{s_e} = \mathbf{w_e} \\
\forall k \in \mathcal{K}, \forall \mathbf{e} \in \mathcal{N}_T^k \quad & z_\mathbf{e} \geq W_0 - W_0(\xi_e \mathbf{w_{a(e)}}) - \zeta \\
\forall k \in \mathcal{K}, \forall \mathbf{e} \in \mathcal{N}_T^k \quad & z_\mathbf{e} \geq 0 \\
\forall k \in \mathcal{K} \quad & \sum_{\mathbf{e}\in\mathcal{N}_T^k} p_\mathbf{e} \xi_\mathbf{e} \mathbf{w_{a(e)}} \geq R \\
\forall k \in \mathcal{K}, \forall \mathbf{e} \in \mathcal{N}^k \quad & \mathbf{w^L} \leq \mathbf{w_e} \leq \mathbf{w^U} \\
\forall k \in \mathcal{K}, \forall \mathbf{e} \in \mathcal{N}_I^k \cup \mathcal{N}_0 \quad & \mathbf{0} \leq \mathbf{b_e} \leq \mathbf{b_e^U} \\
\forall k \in \mathcal{K}, \forall \mathbf{e} \in \mathcal{N}_I^k \cup \mathcal{N}_0 \quad & \mathbf{0} \leq \mathbf{s_e} \leq \mathbf{s_e}^U
\end{aligned}$$

Scenarios	Branching	Rows		Variables	
		Model 1	Model 2	Model 1	Model 2
16	4.4	273	1089	769	3169
32	4.8	493	2125	1385	6185
64	8.8	525	3789	1417	11017
128	2.8.8	1099	7627	2981	22181
256	4.8.8	2145	15201	5809	44209

Table 3.1: Comparison of Model Size

3.3 Alternate Formulation

Some readers might wonder why buy sell decisions are only made at interior nodes, and not terminal nodes. It is indeed possible to formulate the problem to include buy sell decision at terminal nodes, as shown in Equation 3.5. We shall call this alternate formulation Model 2, and the original formulation Model 1.

The reason we do not adopt this alternative formulation in this report is due to the adverse effect it has on the performance of the model. This minor modification to the formulation results in a huge increase in the size of the model. Table 3.1 shows the number of rows and variables in the model for the two different formulations. Figure and show the huge increase in the number of rows and variables of the model if Model 2 is used. Section 5.1 shows how this impacts the time taken to solve the model.

$$
\begin{array}{rl}
\text{minimize} & \mathrm{WCVaR}_\alpha \\
\text{subject to} & \mathbf{P} + (1-c_b)\mathbf{b_0} - (1+c_s)\mathbf{s_0} = \mathbf{w_0} \\
& \mathbf{1}'\mathbf{b_0} - \mathbf{1}'\mathbf{s_0} = \mathbf{1} - \mathbf{1}'\mathbf{P} \\
\forall k \in \mathcal{K} & \mathrm{WCVaR}_\alpha \geq \zeta + \dfrac{1}{1-\alpha} \displaystyle\sum_{\mathbf{e} \in \mathcal{N}_T^k} p_\mathbf{e} z_\mathbf{e} \\
\forall k \in \mathcal{K}, \forall \mathbf{e} \in \mathcal{N}_I^k \cup \mathcal{N}_k^k & \mathbf{1}'\mathbf{b_e} - \mathbf{1}'\mathbf{s_e} = 0 \\
\forall k \in \mathcal{K}, \forall \mathbf{e} \in \mathcal{N}_I^k \cup \mathcal{N}_k^k & \xi_\mathbf{e}\mathbf{w_{a(e)}} + (1-c_b)\mathbf{b_e} - (1+c_s)\mathbf{s_e} = \mathbf{w_e} \\
\forall k \in \mathcal{K}, \forall \mathbf{e} \in \mathcal{N}_k^k & z_\mathbf{e} \geq W_0 - W_0(\mathbf{w_e}) - \zeta \\
\forall k \in \mathcal{K}, \forall \mathbf{e} \in \mathcal{N}_k^k & z_\mathbf{e} \geq 0 \\
\forall k \in \mathcal{K} & \displaystyle\sum_{\mathbf{e} \in \mathcal{N}_k^k} p_\mathbf{e} \xi_\mathbf{e} \mathbf{w_{a(e)}} \geq R \\
\forall k \in \mathcal{K}, \forall \mathbf{e} \in \mathcal{N}^k & \mathbf{w^L} \leq \mathbf{w_e} \leq \mathbf{w^U} \\
\forall k \in \mathcal{K}, \forall \mathbf{e} \in \mathcal{N}^k \cup \mathcal{N}_0 & \mathbf{0} \leq \mathbf{b_e} \leq \mathbf{b_e^U} \\
\forall k \in \mathcal{K}, \forall \mathbf{e} \in \mathcal{N}^k \cup \mathcal{N}_0 & \mathbf{0} \leq \mathbf{s_e} \leq \mathbf{s_e}^U
\end{array}
\tag{3.5}
$$

3.4 Numerical Examples

Let w^*, ζ^* be the solution to Equation 3.2. Then by optimality, we have

$$\begin{aligned} \min\max\ \{\mathrm{CVaR}_\alpha^k(w,\zeta)\} &= \max\ \{\mathrm{CVaR}_\alpha^k(w^*,\zeta^*)\} && (3.6)\\ &\geq \mathrm{CVaR}_\alpha^k(w^*,\zeta^*),\ \forall k \in \mathcal{K} && (3.7) \end{aligned}$$

Therefore, if the worst-case scenario does not materialize, the conditional value at risk is guaranteed to improve. This non-inferiority of the min-max optimization strategy can be seen in Figure 3.1, which shows the portfolio returns when we implement the min-max strategy on each rival scenario tree. As expected, the performance improves if the worst-case scenario does not materialize. This can be seen more clearly in Figure 3.2, which shows the returns relative to the efficient frontier. Figures 3.3 and 3.4 shows the equivalent figures when the optimization is performed without any bounds on the asset weights.

As shown in the figures, in many cases the min-max strategy is able to optimize the portfolios in such a way that the performance of the portfolio is identical regardless of the scenario tree that is realised. This is especially true when the bounds on the asset weights are removed, as the min-max strategy has more freedom when optimizing the portfolio. This phenomenon of multiple-maxima is good as it gives us a guaranteed performance regardless of the scenario tree that is realized.

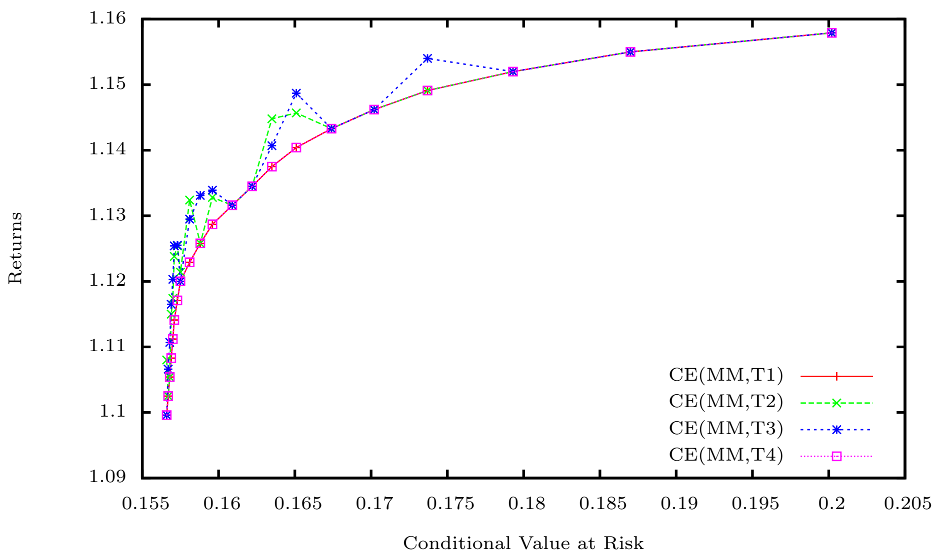

Figure 3.1: Non-Inferiority of Min-Max (With Bounds)

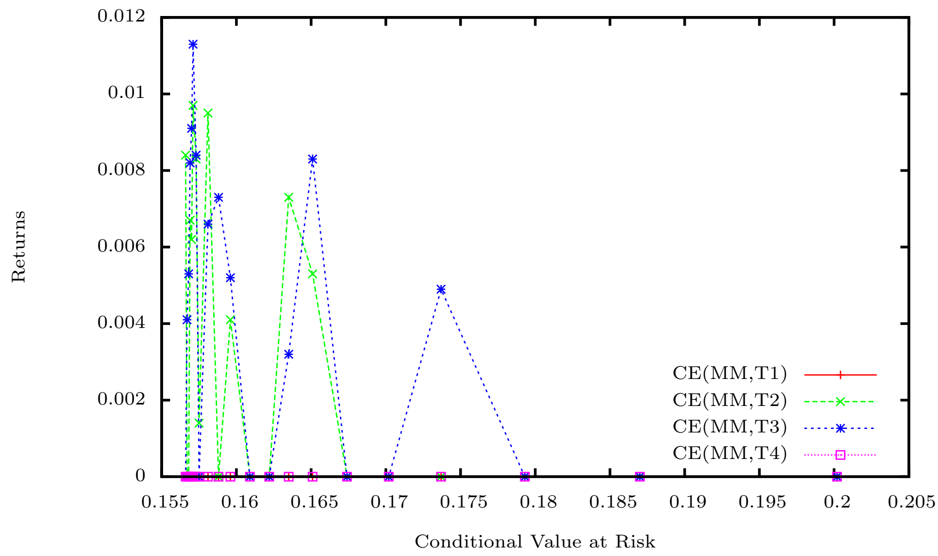

Figure 3.2: Non-Inferiority of Min-Max (Returns Relative to Efficient Frontier, With Bounds)

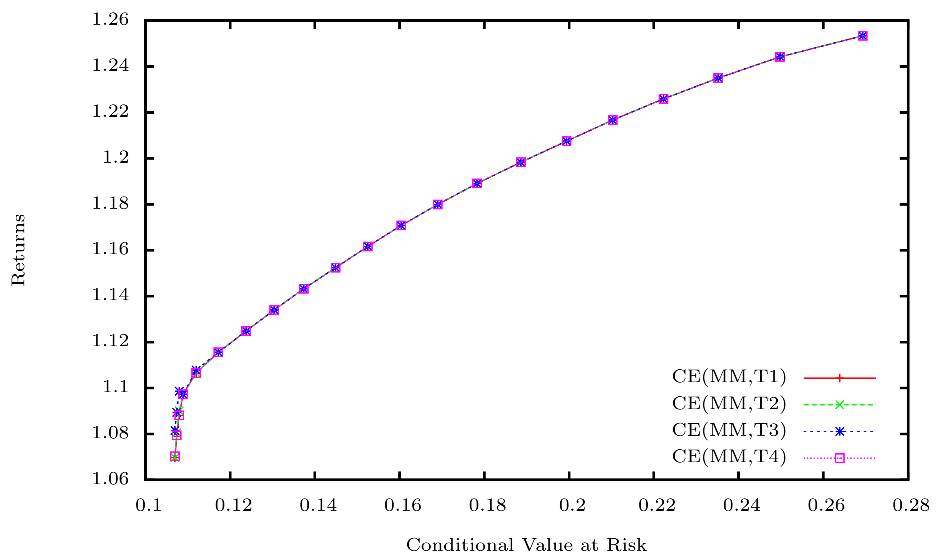

Figure 3.3: Non-Inferiority of Min-Max (Without Bounds)

Efficient Frontiers

Figure 3.5 shows the efficient frontiers of various strategies. The bottom curve shows the efficient frontier obtained by min-max optimization over four scenario trees. The other four curves shows the efficient frontier obtained when optimizing using each individual tree.

Cross Evaluation

When presented with rival scenario trees, implementing the strategy obtained by optimizing over a rival scenario tree can lead to poor results. Figure 3.6 shows what would happen if Tree 1 is realized. At lower risk levels, implementing the strategies obtained by optimizing over trees 2, 3, and 4 gives results that are clearly below the efficient frontier obtained by optimizing over Tree 1. However, we also notice that at higher risk levels, those strategies actually give performance that is above the efficient frontier.

By definition, no portfolios should exist above the efficient frontier. The reason these strategies were able to out-perform the efficient frontier is due to the inclusion of bounds on the asset weights when performing the optimization. When we implement the alternative strategies, these bounds might be violated, which could lead to better performance. Figure 3.7 shows the results when we remove the bounds when performing the optimization. The results now show the efficient frontier clearly dominating all the alternative strategies. Furthermore, we also observe that the min-max optimization gives performance that is superior to all strategies apart from the strategy obtained by optimizing over Tree 1. Figures 3.8, 3.9, and 3.10 show similar graphs assuming Tree 2, 3, and 4 are realized respectively. Further cross evaluation results can be found in Appendix B.

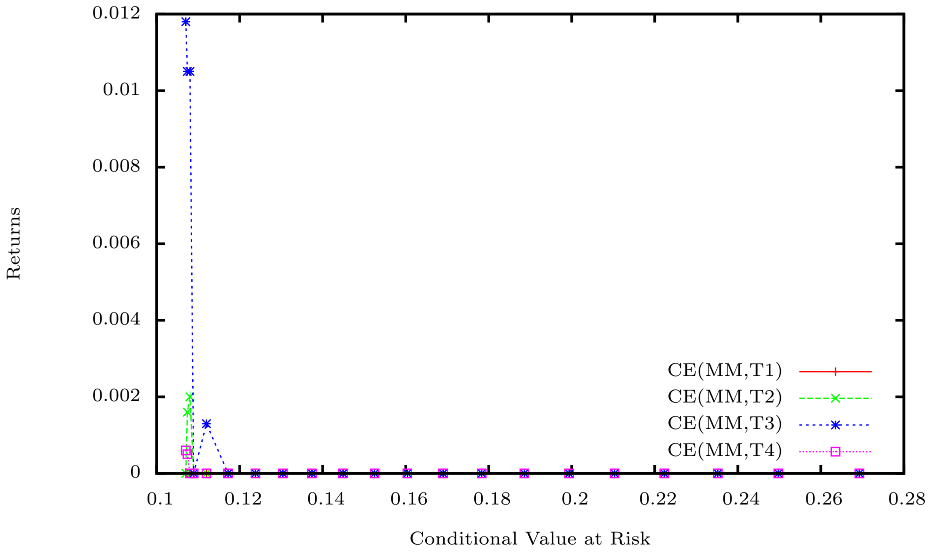

Figure 3.4: Non-Inferiority of Min-Max (Returns Relative to Efficient Frontier, Without Bounds)

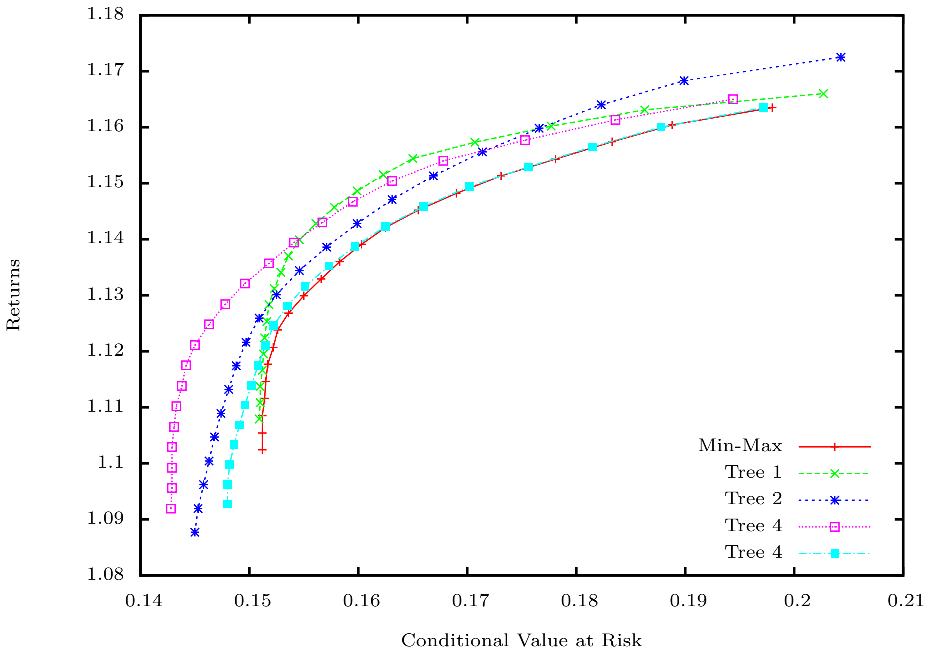

Figure 3.5: Efficient Frontiers of various strategies

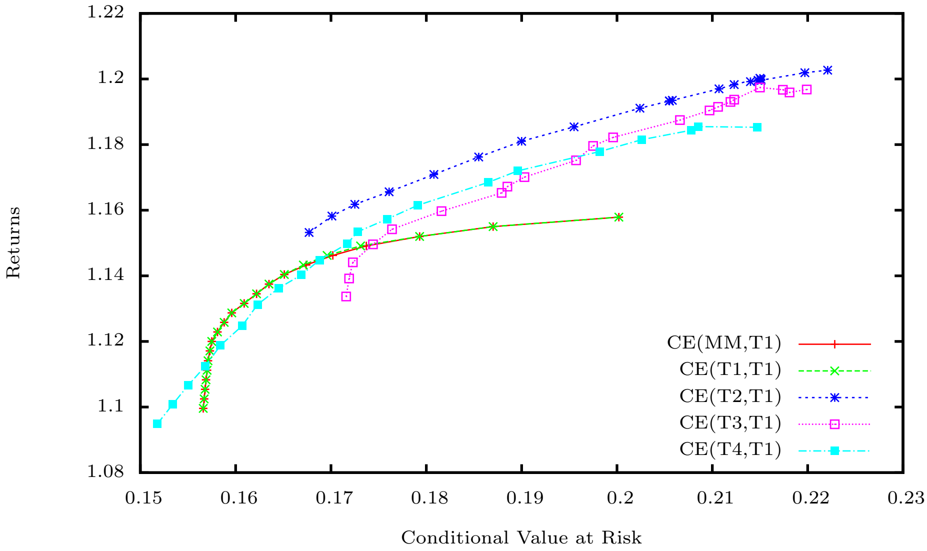

Figure 3.6: Cross Evaluation of Implementing Various Strategies on Tree 1 (With bounds)

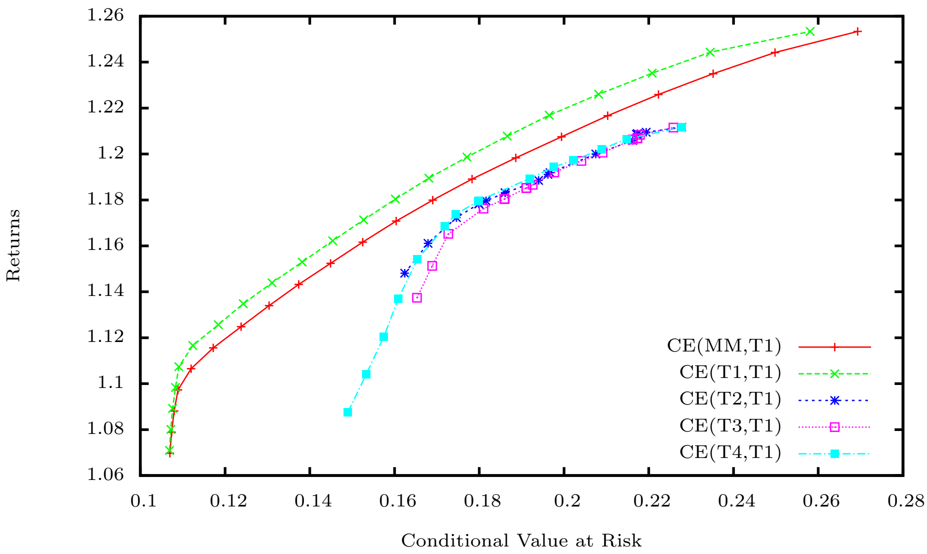

Figure 3.7: Cross Evaluation of Implementing Various Strategies on Tree 1 (Without bounds)

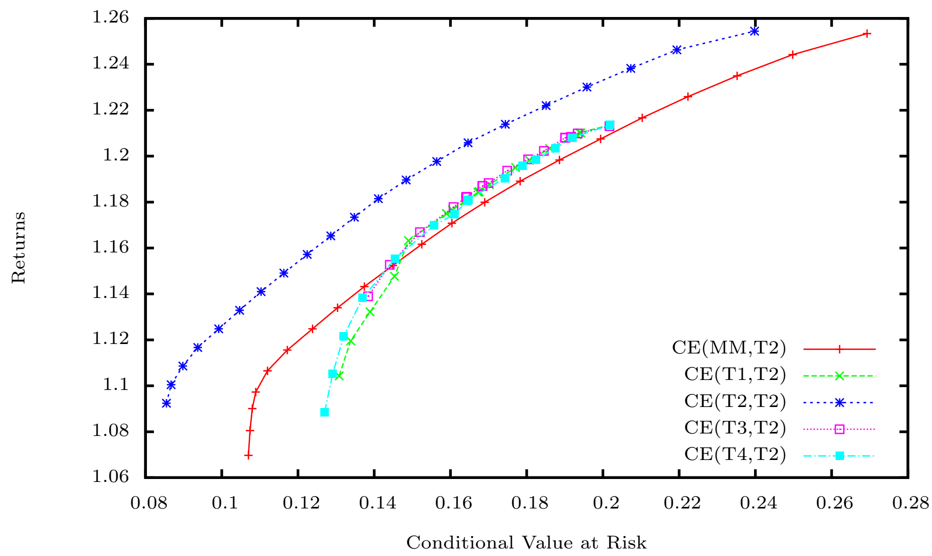

Figure 3.8: Cross Evaluation of Implementing Various Strategies on Tree 2 (Without bounds)

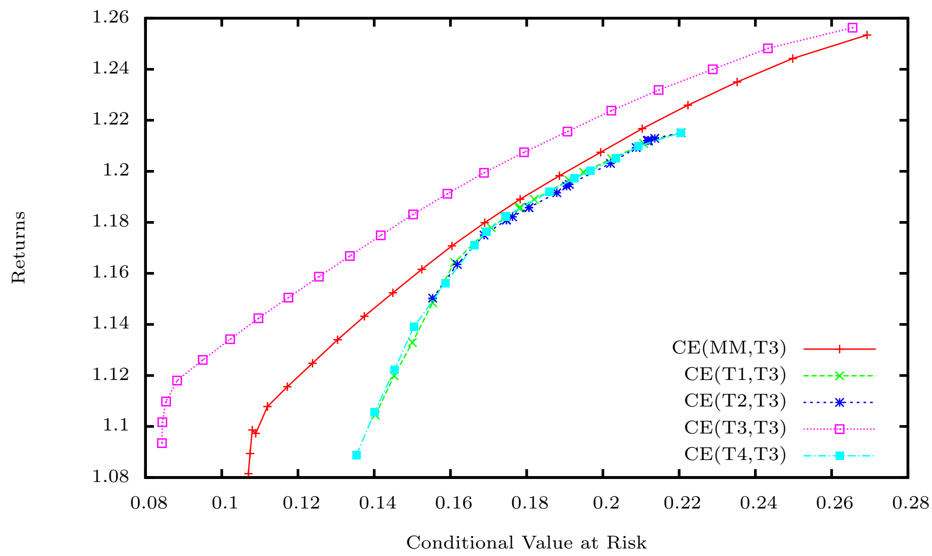

Figure 3.9: Cross Evaluation of Implementing Various Strategies on Tree 3 (Without bounds)

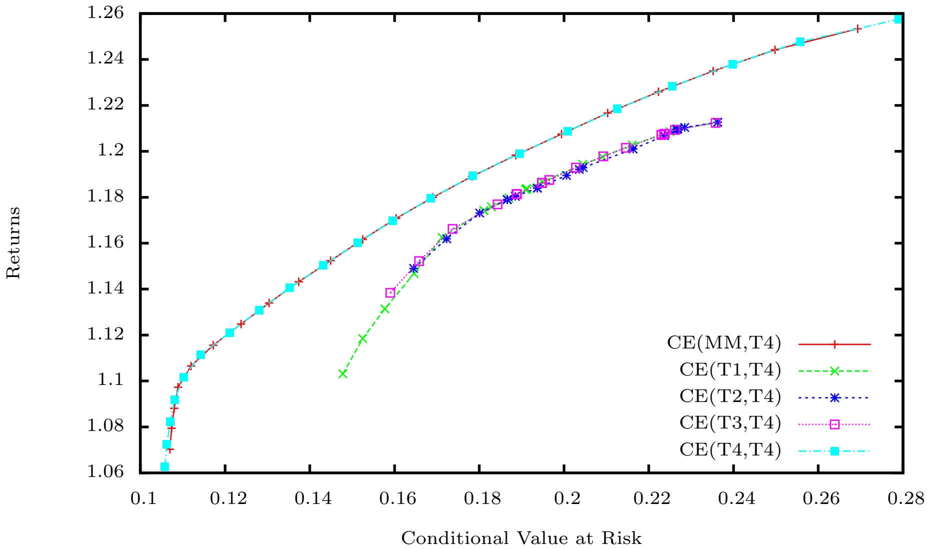

Figure 3.10: Cross Evaluation of Implementing Various Strategies on Tree 4 (Without bounds)

Stability of Results

The optimization results will depend on the scenario trees that are used. Since we are using a randomized procedure to scenario tree generation, repetitive runs of the experiments will yield different scenario trees, which will in turn lead to different optimization results.

To analyze the stability of the optimization results, we repeat the optimization 100 times, using different scenario trees for each optimization. This would give us 100 different efficient frontiers. Figures 3.11 and 3.12 show the box and whiskers plots of the min-max and single tree efficient frontiers respectively. Figures 3.13 and 3.14 show the variance of the returns on the min-max and single tree efficient frontiers respectively. The results show the efficient frontier obtained by min-max optimization has a much lower standard deviation compared to the efficient frontier obtained by single tree optimization. This shows that min-max optimization produces results that are more stable compared to optimization using a single tree.

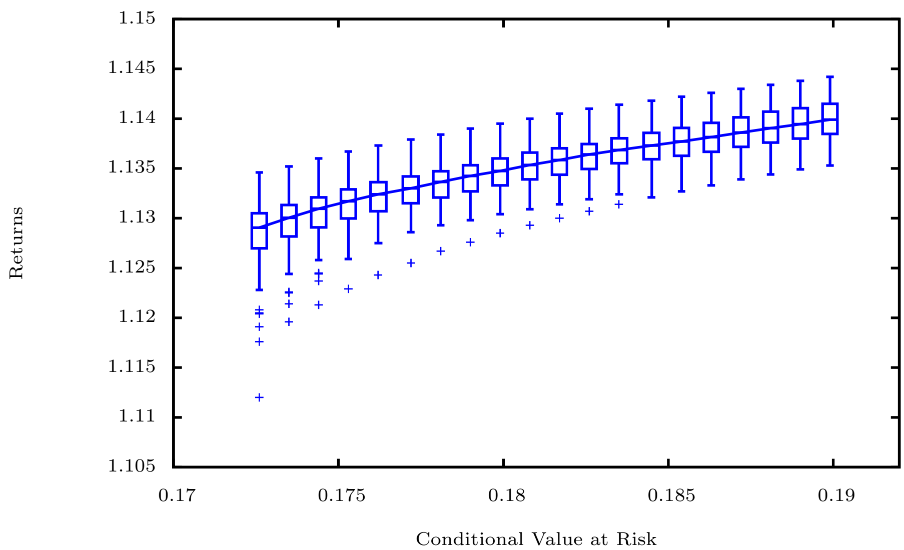

Figure 3.11: Box and Whiskers Plot of Min-Max Efficient Frontier

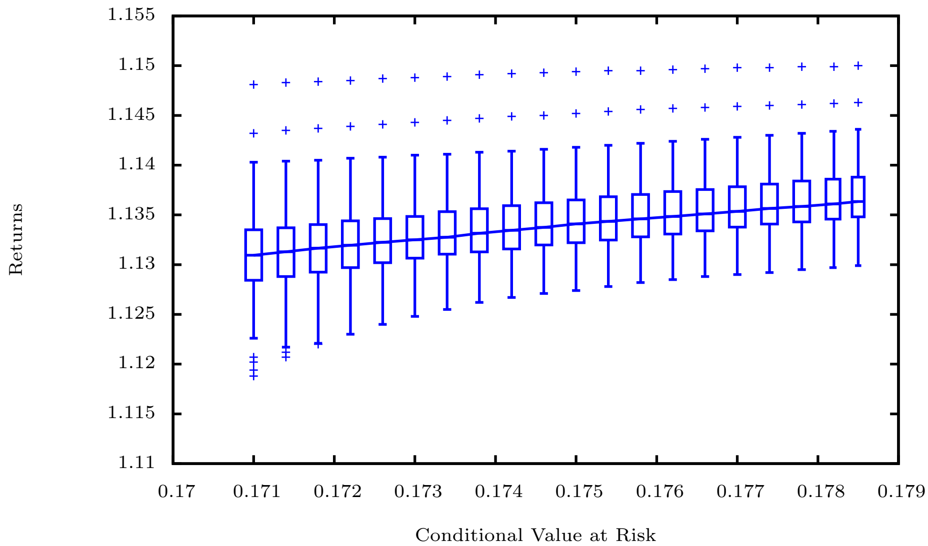

Figure 3.12: Box and Whiskers Plot of Single Tree Efficient Frontier

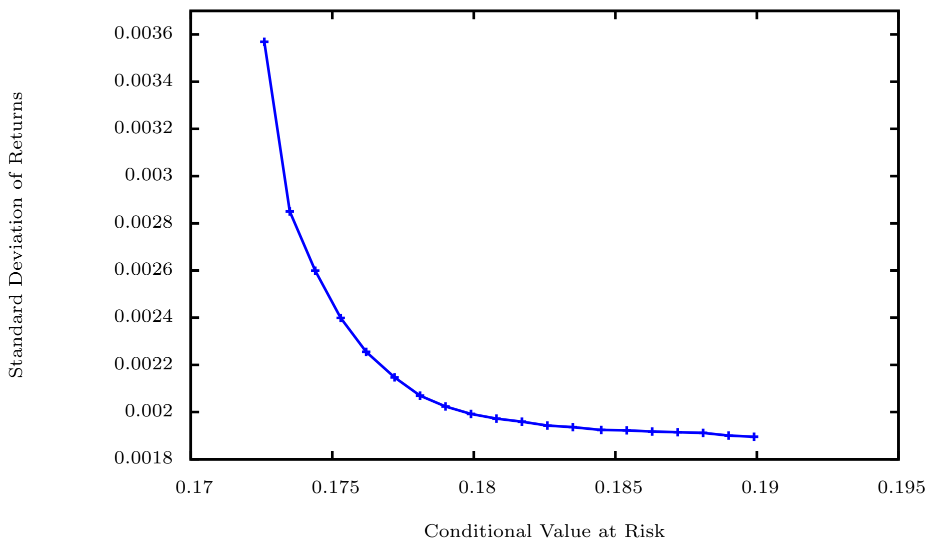

Figure 3.13: Standard Deviation of Returns on Min-Max Efficient Frontier

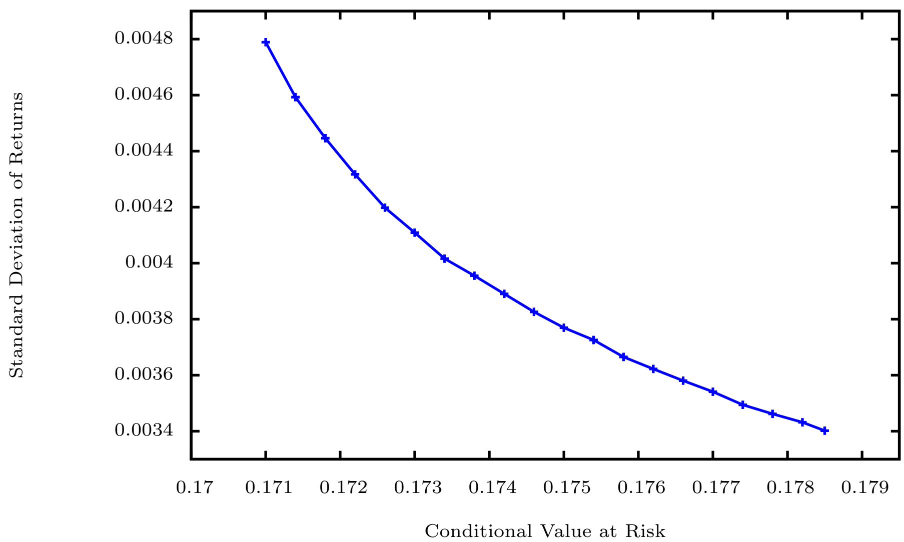

Figure 3.14: Standard Deviation of Returns on Single Tree Efficient Frontier

Chapter 4

Design and Implementation

The goal of the project is to implement the model described in Section 3.2, and to evaluate the performance of the mean-conditional value at risk portfolio optimization technique through backtesting. The following sections describe the various components that make up the overall system.

4.1 Historical Data

Historical prices of the assets are needed to calculate the assets' exponential growth rates and covariance matrix. Furthermore, the historical prices are also used during backtesting.

4.1.1 Data Source

Historical prices of assets from the FTSE 100 can be found online on sites such as Yahoo! Finance and Google Finance. However, I also required the market capitilization values for the assets, and this information was not available on either websites. Furthermore, I wanted data spanning the past 20 years so that I could perform more comprehensive backtesting, and neither websites were able to provide that amount of data. Therefore, I decided to obtain the historical data from the Thomson Datastream service provided by the Imperial College Library.

Thomsom Datastream is a widely large financial statistical database. I was able to download the data I wanted into a Microsoft Excel file. I then wrote a Ruby script to copy the data into the SQLite database, which is described below.

4.1.2 Data Storage

I chose to store the historical prices in a database. Using a database allows me to manage the data much more efficiently than using a flat file. Instead of using the departmental PostgreSQL database, I decided to use SQLite as the database management system. SQLite is an embedded SQL database engine that can be linked in to the program instead of running as a standalone process. This leads to lower system complexity and also higher performance since a database call is simply a function call, and does not rely on network calls to a separate database server.

4.2 Generating Scenario Trees

To generate the scenario trees, we use the simulation and clustering approach described in [8]. The Department of Computing at Imperial College London has an implementation of the simulation and randomized clustering approach to scenario tree generation - the C++ application *Cluster*. *Cluster* performs a large number of static simulations on assets, and then "summarises" those simulation by clustering the results around a centroid, and outputting these centroids in the form of a multi-stage scenario tree.

The simulation is guided by statistical information, which is provided as input to Cluster. The information used to generate the scenarios are:

- Exponential Growth Rate
- Covariance matrix describing the deviation of the assets

The exponential growth rate and covariance matrix can both be calculated from historical asset prices. As described in Appendix A, a linear least squares fitting is used to calculate the exponential growth rate. Instead of manually writing functions to calculate the covariance matrix and perform the least squares fitting, I decided to use the functions provided by the GNU Scientific Library (GSL), which is a free numerical library that provides a wide range of mathematical routines.

Apart from the above information, the following information can also be provided to Cluster to configure the shape and size of the scenario tree:

- Depth of the Scenario Tree
- Branching Rate
- Number of Scenarios

- Initial Prices

4.3 Model Implementation

4.3.1 Mean-Variance Model

The multi period mean-variance optimisation problem described in [7] has been implemented as a C++ software package known as *DFoliage*. *DFoliage* uses the BPMPD linear programming/quadratic programming solver developed at the Deparment of Computing, Imperial College London. BPMPD is based on an interior point alogrithm.

Instead of re-implementing the mean-variance model, I decided to use *DFoliage*. However, I had to write a wrapper around *DFoliage* as it currently only exists as a standalone binary with no API that allows me to call functions in *DFoliage*. Writing a wrapper also allowed me to ensure both the mean-variance model and the mean-conditional value at risk model that I will be writing will have the same interface, thus simplifying the development process

4.3.2 Mean-Conditional at Risk Model

Modelling Language

The constraints of the linear programming problem can be modelled as a matrix, while the objective function can be modelled as a vector. Therefore, constructing the model involves creating and manipulating matrices and vectors. For this purpose, I chose to use the matrix and vector classes provided by COIN-Utils.

There are various benefits gained from using COIN-Utils. First of all, COIN-Utils is provided by the same organization that provides COIN LP (CLP) - the simplex solver that I will be using. Therefore, the integration between the two means that I do not need to do any additional conversions when inputting my model to the simplex solver. Furthermore, COIN-Utils also allows me to output the model I created in MPS format, which allows me to use alternate solvers if needed.

Linear Programming Solver

The portfolio optimization problem described is a linear programming problem that can be solved with an optimization library.

When implementing the software, I discovered that the choice of solver has a substantial impact on the time taken to solve the linear programming problem.

Therefore, it was important that the user can easily change the solver that is used.

Instead of calling the solver directly, I chose to use the COIN-OR Open Solver Interface (OSI), which is a uniform API for interacting with callable solver libraries. OSI supports many popular linear programming solvers such as COIN-OR Linear Programming Solver (CLP), the GNU Linear Programming Kit (GLPK), etc. In addition to the solvers supported by OSI, I also added the ability to use BPMPD, which is the solver used by $DFoliage$.

4.4 Backtester

The backtester is used to evaluate the performance of the models using historical data. To accomplish this, it needs to use all the components that were described in the previous sections. The steps taken to perform the back testing are as follows:

Analyze Calculate the exponential growth rates and covariance matrix using the historical data from the database.

Forecast Generate a scenario tree using $Cluster$, passing in the exponential growth rates and covariance matrix.

Optimize Obtain the optimal investment decisions by solving the mean-conditional value at risk or mean-variance model.

Invest Invest among the various assets according to the initial allocation obtained from the optimal solution.

Accrue Move from current time t to $t + 1$, and update the portfolio value according to the change in price.

Iterate Go to Step "Analyze" until we run out of historical data.

Chapter 5

Evaluation

5.1 Performance

As mentioned in Section 3.3, the alternative formulation of the problem leads to a substantial increase in the problem size. In this section, we show how this increase in problem size affects the time taken to solve the optimization problem. Model 1 is the model described in Section 3.2, and Model 2 is the alternate model described in 3.3.

Furthermore, we also show how the choice of linear programming solver can have a huge impact on the time taken. The linear programming solvers tested include an interior point method solver (BPMPD) and two simplex solvers (CLP and GLPK).

For each optimization problem, 50 assets were used. The tests were performed on an Intel Core 2 Duo 2.13GHz machine with 2GB of RAM.

Scenarios	Branching	Time (s)					
		Model 1			Model 2		
		BPMPD	CLP	GLPK	BPMPD	CLP	GLPK
16	4.4	1.71	0.51	0.74	3.63	3.18	7.04
32	4.8	3.49	1.10	1.89	7.09	9.16	24.57
64	8.8	5.03	1.31	1.95	17.17	25.36	89.22
128	2.8.8	16.88	2.90	7.90	49.39	82.18	357.06
256	4.8.8	51.78	7.98	32.96	127.83	291.01	1630.94
512	2.4.8.8	271.29	32.55	126.86	-	-	-
1024	4.4.8.8	945.85	126.45	611.09	-	-	-

Table 5.1: Performance of various LP solvers

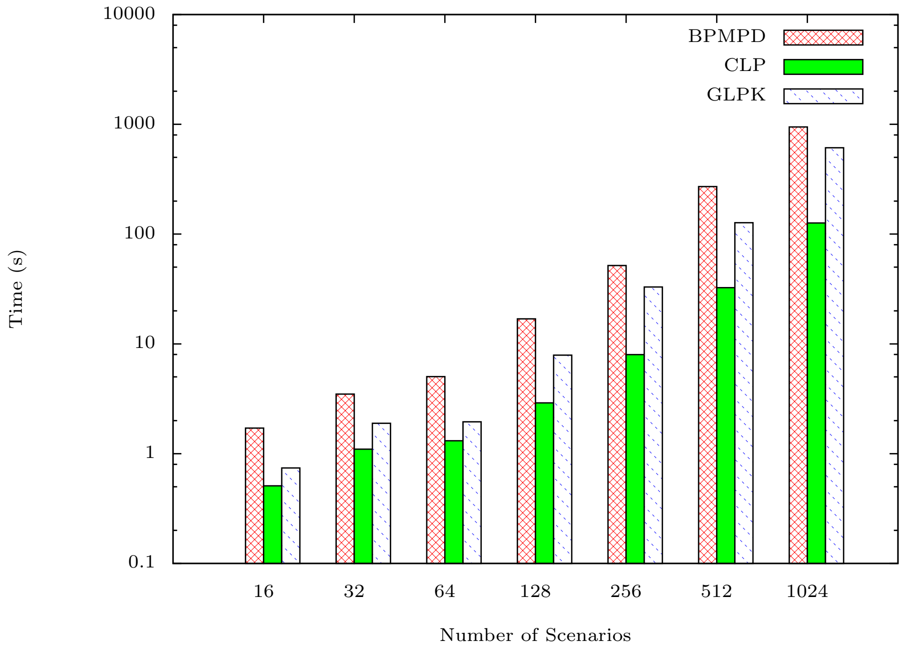

Figure 5.1: Comparison of time taken to solve Model 1

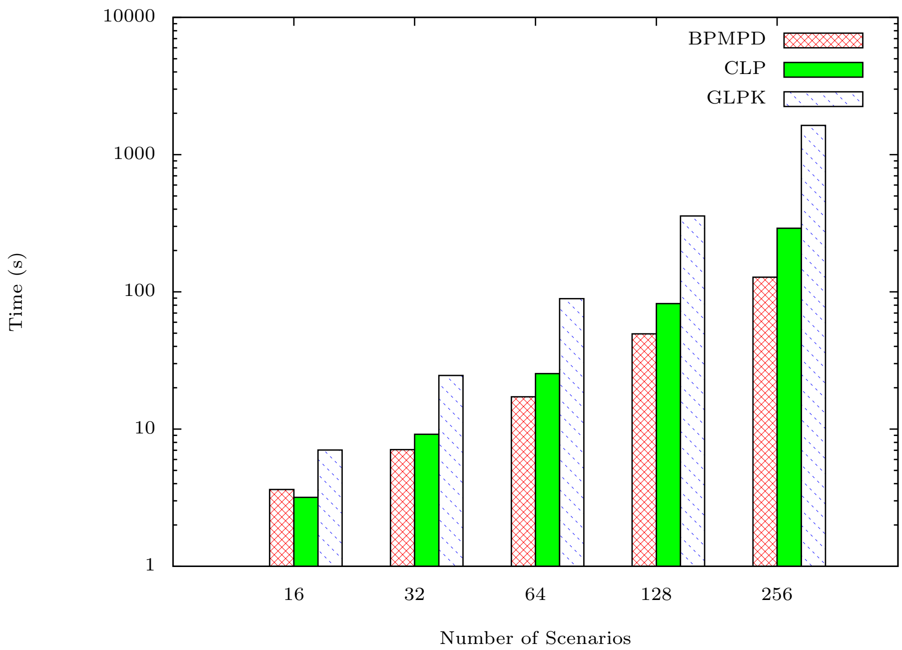

Figure 5.2: Comparison of time taken to solve Model 2

5.1.1 Results

Table 5.1 shows the time taken by each optimizer to find 20 equally spaced points on the efficient frontier. Figure 5.1 and 5.2 show the data in Table 5.1 in graphical form.

As expected, the time taken by the alternative formulation is much greater due to the increase in the problem size. We can also see that the time taken increases quickly as we increase size of the scenario tree. The reader should also note that the graphs are plotted on a logarithmic scale, so the increase in time is close to exponential.

An interesting observation that can be made is that with the original formulation, the simplex solvers perform better than the interior point solver. However, we observe the opposite when we use the alternative formulation. This is likely due to the much larger problem size encountered using the alternative formulation. Further investigation can be made to uncover the reasoning behind this observation.

5.2 Backtesting

To demonstrate the viability of the techniques that we have developed, we evaluate the performance of the models that we have developed through backtesting. We choose to perform the backtesting using 20 assets from the FTSE 100 Index. A complete list of the stocks used can be found in Appendix C.

The value for the expected returns constraint would depend on the risk-level chosen for the optimization strategy. For a risk level r, the expected return $\mathcal{W}_r = r\mathcal{W}_{max} + (1-r)\mathcal{W}_{min}$.

We have chosen two benchmarks to assess the performance of the various optimization strategies. The first benchmark is the FTSE 100 Index, which is a share index of the 100 most highly capitalized companies listed on the London Stock Exchange. The FTSE 100 Index gives a broad representation of the performance of UK companies, and thus serves as a good benchmark.

The second benchmark chosen is the naive equally weighted portfolio, where a fraction of $1/n$ of wealth is allocated to each of the n assets available at the start, and no further rebalancing is done after that. It is argued in [3] that the allocation mistakes caused by the equal weights in the equally weighted portfolio can turn out to be smaller than the weights from the optimized model that has inputs that has been estimated with errors.

5.2.1 Mean-CVaR vs. Mean-Variance

We first compare the performance of the mean-conditional value at risk model against the classical mean-variance model. For a fair comparison, we use a single scenario tree for the mean-conditional value at risk model. We ran the backtesting with the following parameters:

Initial Weights Equal weighting of the twenty assets
Confidence Level 99%
Number of Time Periods in Past Horizon 5
Number of Days Per Time Period 20
Number of Stages 4
Branching Rate 2-2-8-8
Transaction Cost 0.5%
Upper/Lower Bound for Transactions 5%/0%
Upper/Lower Bound for Holding 10%/0.5%
Start/End Date 19th July 1988/27th February 2008

Results

Table 5.2 shows the benchmark relative gains of single scenario tree optimization strategies at various risk levels. Figure 5.3 shows the backtesting results for the mean-conditional value at risk optimized portfolio at risk levels of 10%, 25%, 50%, 75% and 99%. Figure 5.4 shows a similar graph for the mean-variance optimized portfolio. Figures 5.5, 5.6, 5.7, 5.8, and 5.9 show how the mean-conditional value at risk optimized portfolio compares against the mean-variance optimized portfolio at risk levels of 10%, 25%, 50%, 75% and 99% respectively.

Table 5.3 shows the conditional value at risk of single tree optimization strategies at various risk levels. At the 10% and 25% risk levels, the mean-conditional value at risk optimized portfolios has a conditional value at risk similar to that of the FTSE 100 Index, while giving far-superior returns. At higher risk levels, the optimized portfolios clearly outperform the FTSE 100 Index. Although the equally weighted 1/n portfolio gave higher returns compared to the lower risk strategies, it had a much higher conditional value at risk compared to every other strategy.

Figure 5.11 compares the risk-return relationship of every optimization strategy. Both the mean-conditional value at risk and mean-variance optimization strategies are clearly superior compared to the equally weighted 1/n portfolio and the FTSE 100 Index.

Risk Level	Mean-CVaR	Mean-Variance	Equally Weighted	Index
10%	421.2%	450.7%	491.1%	282.8%
25%	458.7%	455.4%		
50%	533.4%	441.1%		
75%	563.2%	587.9%		
99%	527.6%	582.9%		

Table 5.2: Comparison of Returns of Single Scenario Tree Optimization Strategies

Risk Level	Mean-CVaR	Mean-Variance	Equally Weighted	Index
10%	13.7%	14.3%	16.3%	13.8%
25%	13.8%	14.3%		
50%	14.2%	14.6%		
75%	14.6%	14.7%		
99%	15.0%	15.0%		

Table 5.3: Comparison of Conditional Value at Risk of Single Scenario Tree Optimization Strategies

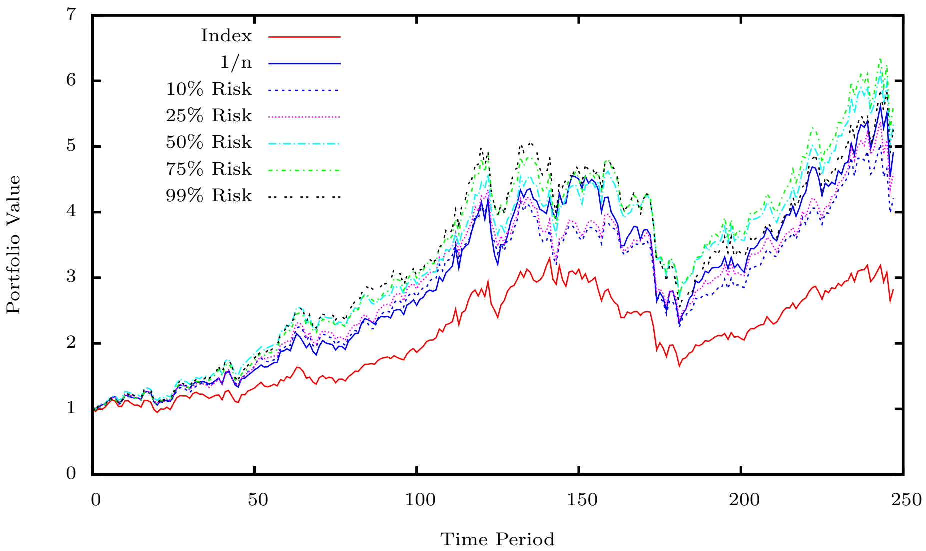

Figure 5.3: Backtesting of Mean-Conditional Value at Risk Model at different risk levels

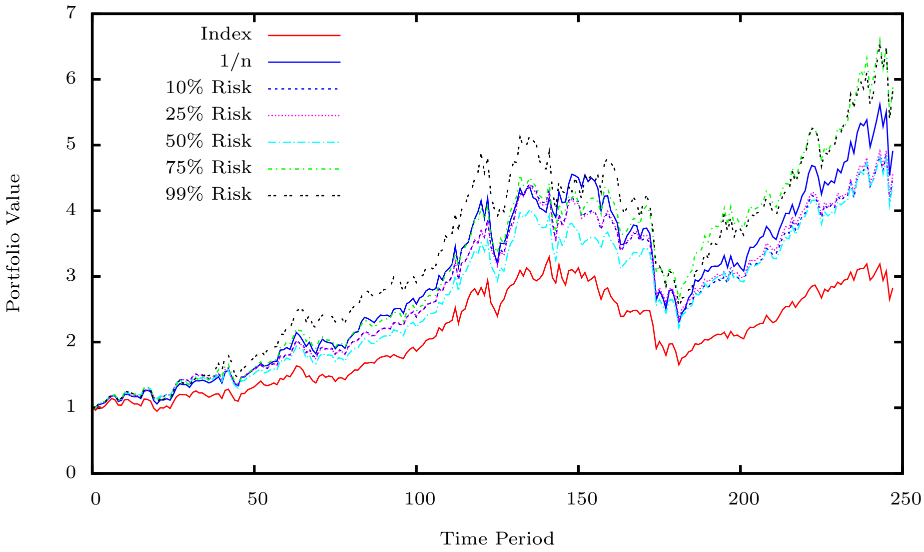

Figure 5.4: Backtesting of Mean-Variance Model at different risk levels

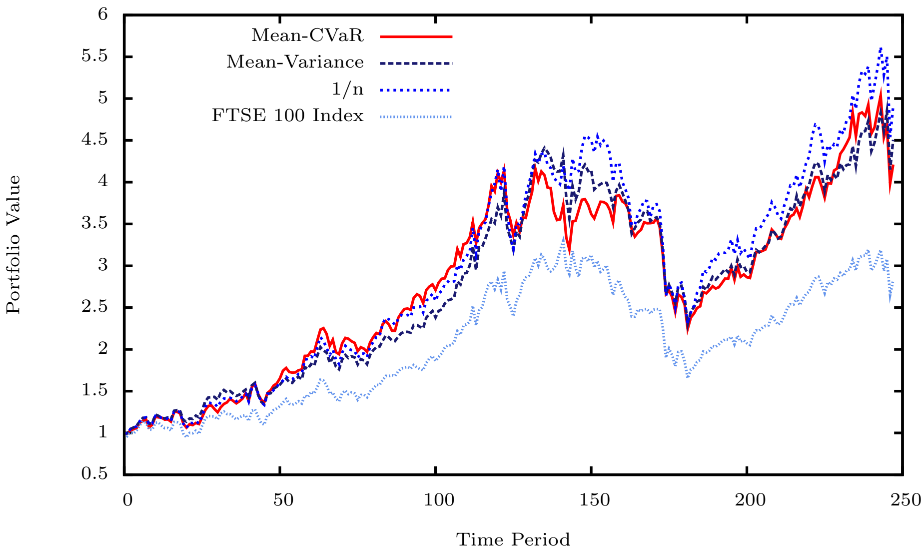

Figure 5.5: Comparison of Mean-CVaR and Mean-Variance optimization strategies at 10% risk level.

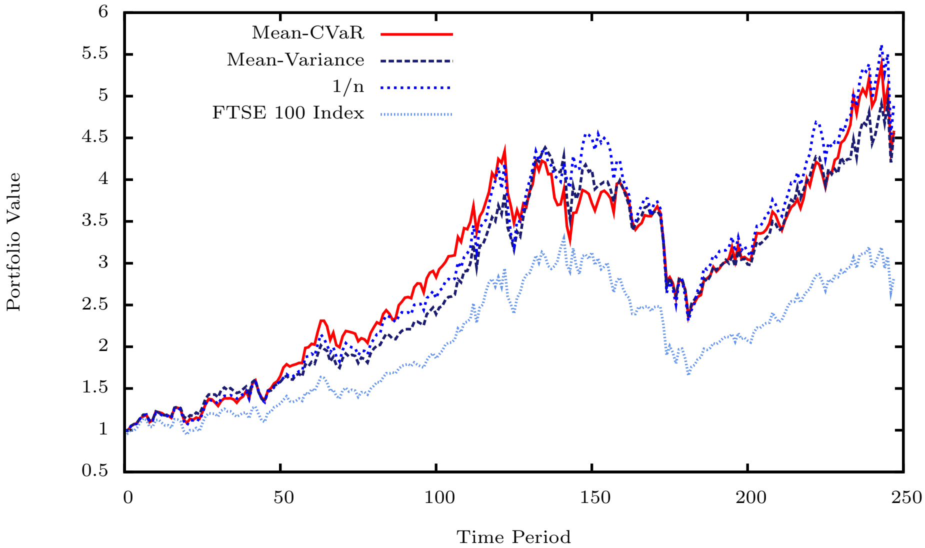

Figure 5.6: Comparison of Mean-CVaR and Mean-Variance optimization strategies at 25% risk level.

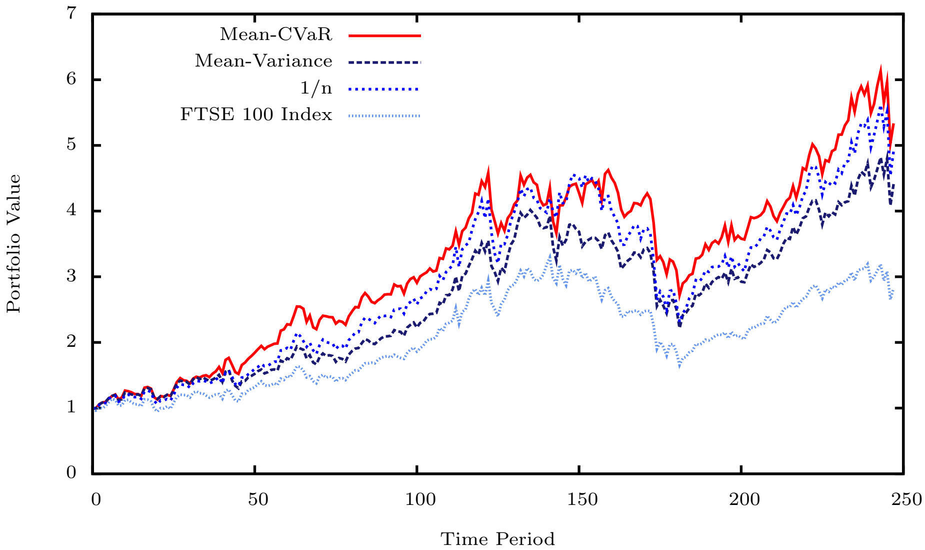

Figure 5.7: Comparison of Mean-CVaR and Mean-Variance optimization strategies at 50% risk level.

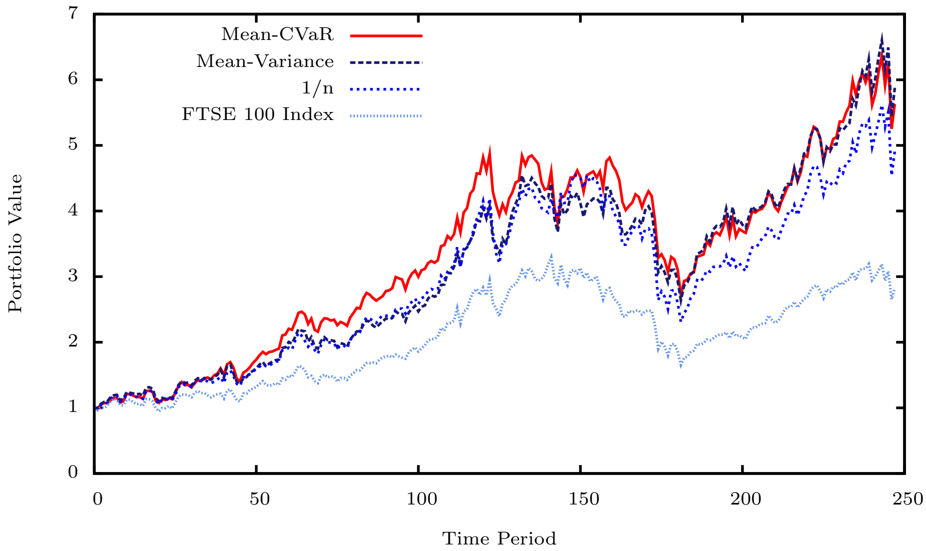

Figure 5.8: Comparison of Mean-CVaR and Mean-Variance optimization strategies at 75% risk level.

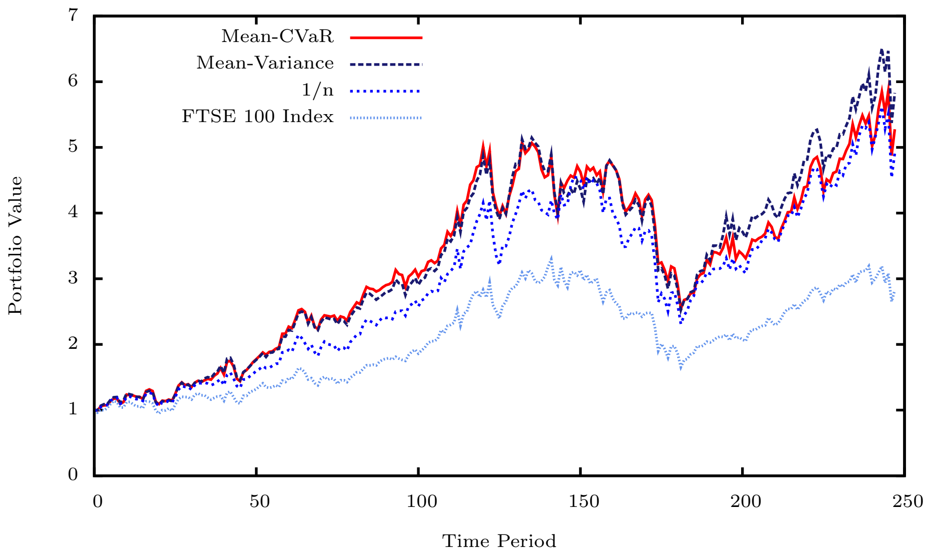

Figure 5.9: Comparison of Mean-CVaR and Mean-Variance optimization strategies at 99% risk level.

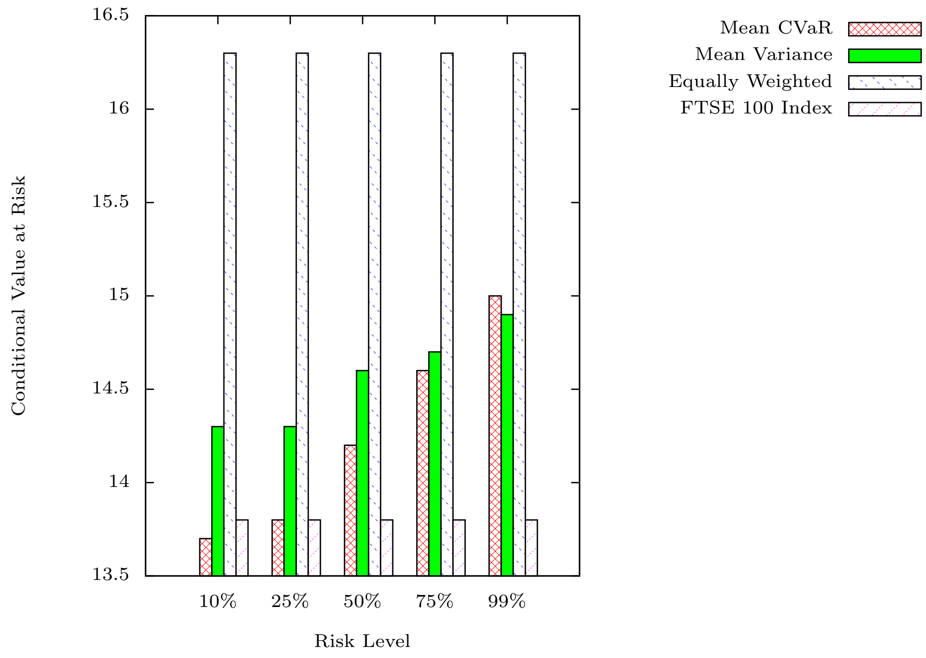

Figure 5.10: Comparison of Conditional Value at Risk at different risk levels

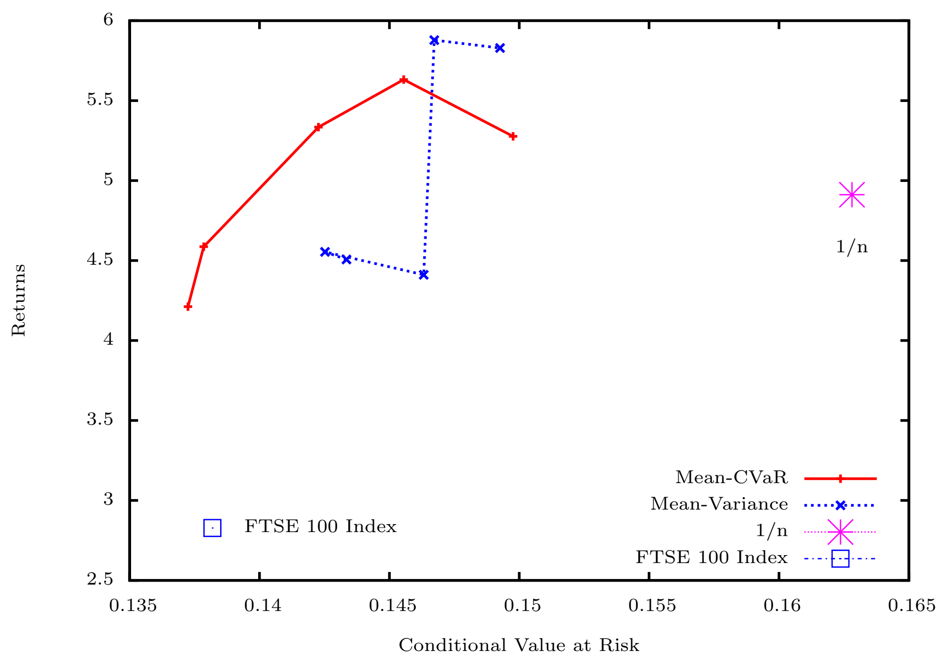

Figure 5.11: Risk-Returns comparison of various strategies

Risk Level	Single Tree	Min-Max	Equally Weighted	Index
10%	311.9%	337.6%	491.1%	282.8%
25%	363.1%	381.5%		
50%	458.3%	454.3%		
75%	535.2%	577.5%		
99%	510.0%	514.5%		

Table 5.4: Comparison of Returns of Single Tree vs. Min-Max Strategies

Risk Level	Single Tree	Min-Max	Equally Weighted	Index
10%	14.1%	13.7%	16.3%	13.8%
25%	14.2%	14.1%		
50%	14.2%	14.1%		
75%	14.8%	14.2%		
99%	14.9%	14.9%		

Table 5.5: Comparison of Conditional Value at Risk of Single Tree vs. Min-Max Strategy

5.2.2 Min-Max vs. Single Scenario

The robust min-max formulation of the model is tested against the single scenario tree model. We tested using four rival scenario trees. We used the same exponential growth rates and covariance matrix for the four scenario trees. To ensure we obtain four different trees, we provided different seeds to the random number generator for each tree.

We ran the backtesting with the same parameters as the previous section.

Results

Table 5.4 shows the benchmark relative gains of the single tree and min-max strategies. Figure 5.12 shows the backtesting results for the worst-case robust mean-conditional value at risk optimized portfolio at risk levels of 10%, 25%, 50%, 75% and 99%. Figures 5.13, 5.14, 5.15, 5.16, and 5.17 show how the single tree mean-conditional value at risk optimized portfolio compares against the worst-case robust mean-conditional value at risk optimized portfolio at risk levels of 10%, 25%, 50%, 75% and 99% respectively.

Table 5.5 shows the conditional value at risk of the various strategies. Figure 5.19 compares the risk-return relationship of every optimization strategy. As expected, we see that the robust min-max strategy is able to give better risk/returns performance when compared against alternative strategies.

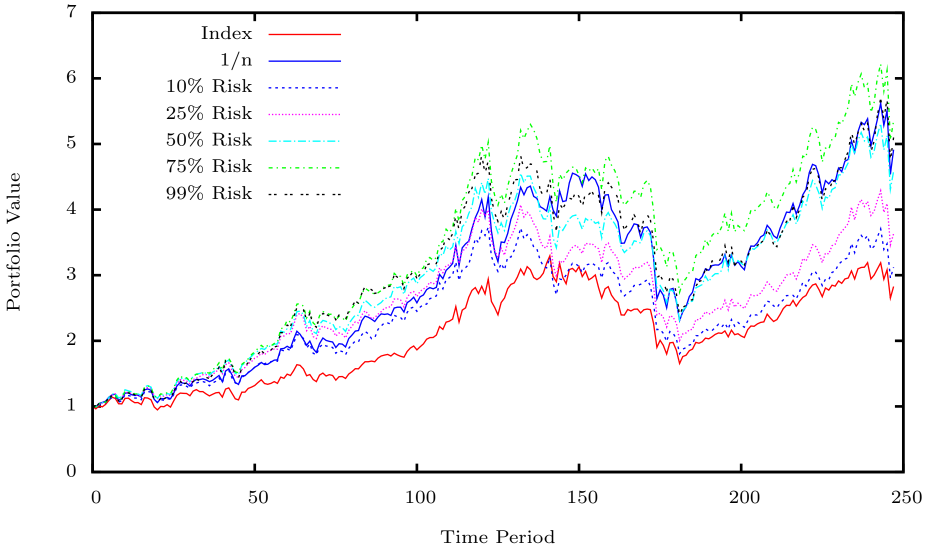

Figure 5.12: Backtesting of min-max optimization at different risk levels

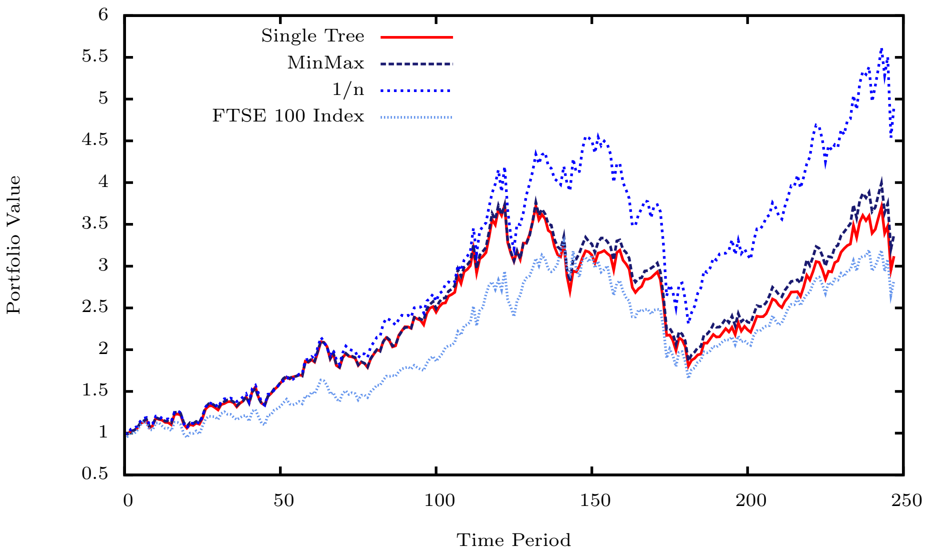

Figure 5.13: Comparison of single tree vs. min-max strategies at 10% risk level.

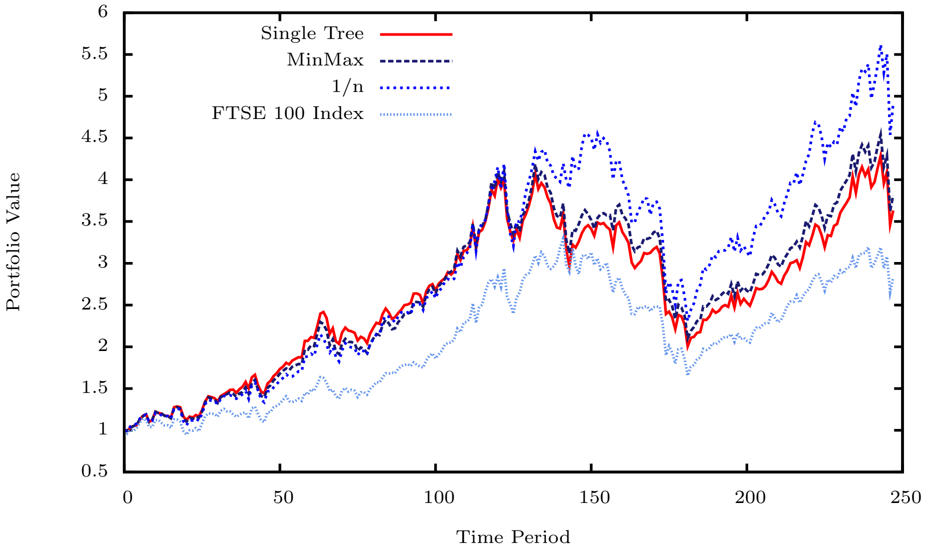

Figure 5.14: Comparison of single tree vs. min-max strategies at 25% risk level.

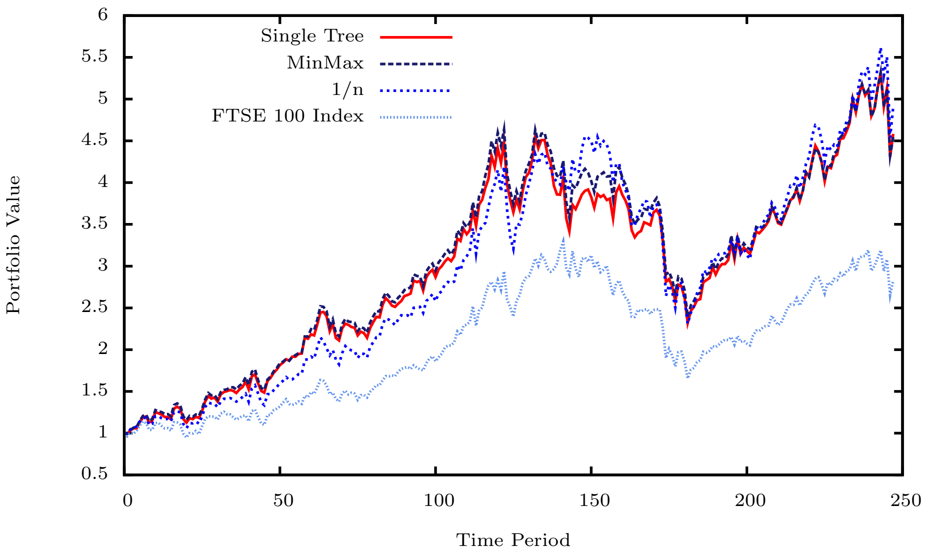

Figure 5.15: Comparison of single tree vs. min-max strategies at 50% risk level.

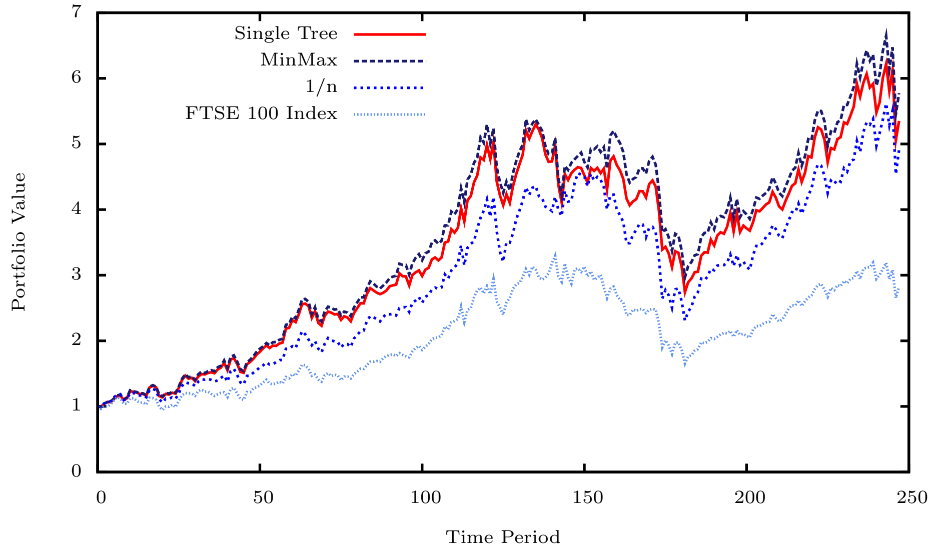

Figure 5.16: Comparison of single tree vs. min-max strategies at 75% risk level.

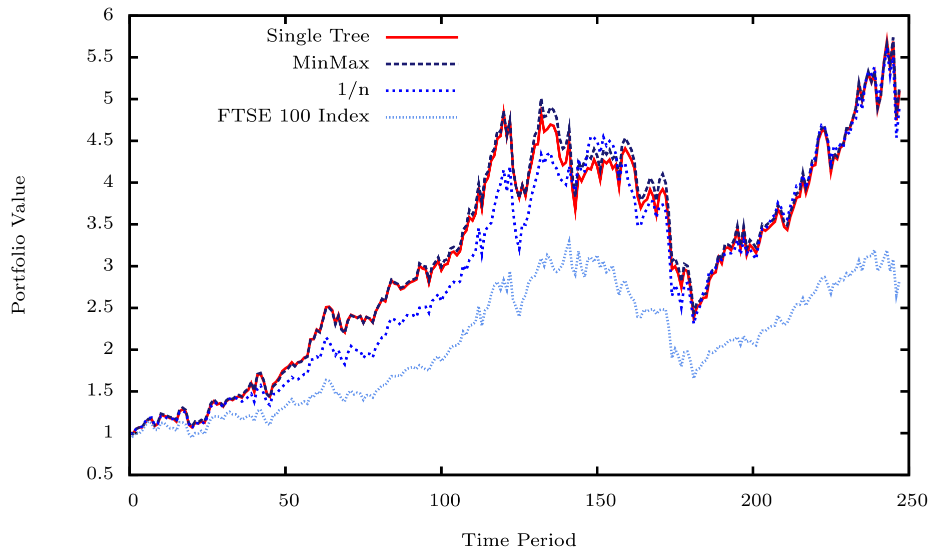

Figure 5.17: Comparison of single tree vs. min-max strategies at 99% risk level.

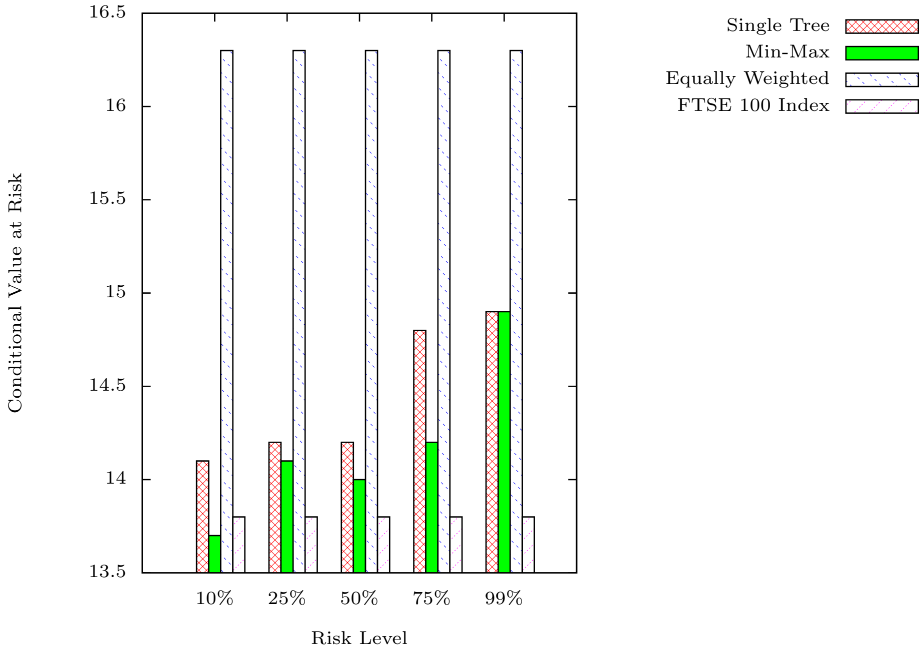

Figure 5.18: Comparison of min-max Conditional Value at Risk at different risk levels

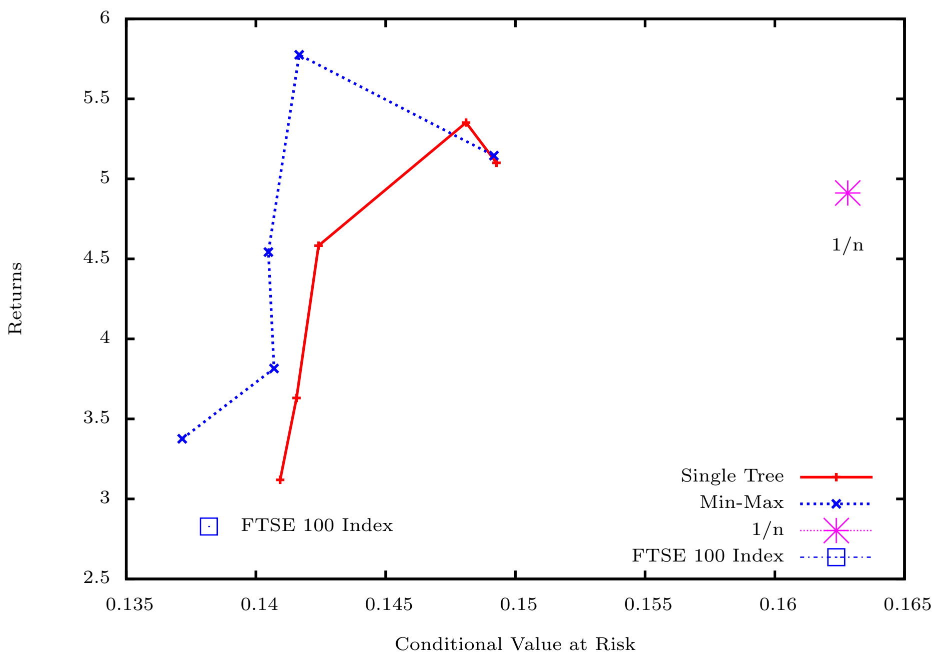

Figure 5.19: Risk-Returns comparison of various min-max strategies

Chapter 6

Conclusion

We have designed, implemented and tested a new robust portfolio optimization model that guarantees optimal worst-case performance. We optimize the mean and conditional value at risk of the portfolio returns of a multiperiod investment problems with transaction costs. We also look at the differences observed when using different linear programming solvers.

Backtesting results show that the robust mean-conditional value at risk optimization model is able to produce high returns while minimizing the conditional value at risk of the portfolio. When compared to the FTSE 100 Index and the Equally Weighted 1/n portfolio, the optimized portfolios give superior risk/return performance.

6.1 Future Work

We have shown in Section 3.4 that applying bounds on the asset weights restricts the min-max strategy. However, these restrictions are needed to ensure diversifcation and over-allocation in a single asset. Further backtesting could be performed to investigate how varying the level of the bounds could affect the backtesting results.

We have used the simulation and clustering approach to scenario tree generation. It would be interesting to look into other means of scenario tree generation and comparing the performance of the various strategies.

Bibliography

[1] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, 9:203–228, 1999. [cited at p. 9, 11]

[2] Chopra, Vijay K., Hensel, Chris R., and Turner, Andrew L. Massaging mean-variance inputs: Returns from alternative global investment strategies in the 1980s. *Management Science*, 39(7):845–855, jul 1993. [cited at p. 13]

[3] Victor DeMiguel, Lorenzo Garlappi, and Raman Uppal. Optimal Versus Naive Diversification: How Inefficient is the 1/N Portfolio Strategy? *Rev. Financ. Stud.*, page hhm075, 2007. [cited at p. 35]

[4] Kevin Dowd. *Beyond Value at Risk: The New Science of Risk Management*. John Wiley & Sons, 1999. [cited at p. 7]

[5] Laurent El Ghaoui, Maksim Oks, and Francois Oustry. Worst-Case Value-At-Risk and Robust Portfolio Optimization: A Conic Programming Approach. *Oper. Res.*, 51(4):543–556, 2003. [cited at p. 15]

[6] Nalan Gülpinar and Berç Rustem. Worst-case robust decisions for multi-period mean-variance portfolio optimization. *European Journal of Operational Research*, 127(3):981–1000, December 2007. [cited at p. 15]

[7] Nalan Gülpinar, Berç Rustem, and Reuben Settergren. Multistage stochastic mean-variance portfolio analysis with transaction costs. *Innovations in Financial and Economic Networks*, 3:46–63, 2003. [cited at p. 13, 31]

[8] Nalan Gülpinar, Berç Rustem, and Reuben Settergren. Optimization and simulation approaches to scenario tree generation. *Journal of Economics Dynamics and Control*, 28:1291–1315, 2004. [cited at p. 12, 30]

[9] Hoyland, Kjetil and Wallace, Stein W. Generating scenario trees for multistage decision problems. *Management Science*, 47(2):295–307, February 2001. [cited at p. 12]

[10] John Hull and Alan White. Incorporating volatility updating into the historical simulation method for value at risk, 1998. [cited at p. 8]

[11] Carlos Jabbour, Javier F. Pena, Juan C. Vera, and Luis F. Zuluaga. An estimation-free, robust CVaR portfolio allocation model. 2007. [cited at p. 15]

[12] Philippe Jorion. *The New Benchmark for Controlling Derivatives Risk: Value at Risk*. McGraw-Hill, 1997. [cited at p. 7]

[13] Reuters J.P.Morgan. Riskmetrics technical document. Technical report, December 1996. [cited at p. 7]

[14] Vinay Kaura. Portfolio optimisation using value at risk. Master's thesis, Imperial College London, 2005. [cited at p. 14]

[15] Pieter Klaassen. Comment on "generating scenario trees for multistage decision problems". *Management Science*, 48(11):1512–1516, November 2002. [cited at p. 12]

[16] Konno, Hiroshi and Yamazaki, Hiroaki. Mean-absolute deviation portfolio optimization model and its applications to tokyo stock market. *Management Science*, 37(5):519–531, May 1991. [cited at p. 13]

[17] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, March 1952. [cited at p. 1, 5, 7]

[18] Jonas Palmquist, S. Uryasev, and Pavlo Krokhalm. Portfolio optimization with conditional value-at-risk objective and constraints. Research report 99-14, University of Florida, Center for Applied Optimization, November 1999. [cited at p. 11]

[19] Andre F. Perold. Large-scale portfolio optimization. *Management Science*, 30(10):1143–1160, October 1984. [cited at p. 13]

[20] G. Pflug. Some remarks on the value-at-risk and the conditional value-at-risk, 2000. [cited at p. 11]

[21] RiskMetrics. Return to riskmetrics: The evolution of a standard. Technical report, April 2001. [cited at p. 7]

[22] Rockafellar R.T. and S. Uryasev. Optimization of conditional value at risk. *Journal of Risk*, 2:21–41, 2000. [cited at p. 10, 11]

[23] Rockafellar R.T. and S. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance*, 26:1443–1471(29), July 2002. [cited at p. 11]

[24] S. Uryasev. Lecture notes: Esi 6912 stochastic optimization: Optimization using cvar - algorithms and applications. [cited at p. 7]

Appendices

Appendix A

Exponential Growth Rate and Covariance

This section explains how the exponential growth rates and covariance matrix can be calculated using historical asset prices.

A.1 Exponential Growth Rate

The general exponential growth curve can be defined as

$$P = Ae^{\lambda t}$$

where P is the price of the asset and t is the time when the price P was observed.

By taking logarithms of both sides, we get

$$\log(P) = \log(A) + \lambda t$$

The parameters A and λ can be found through Least Square fitting:

$$\begin{aligned} \lambda &= \frac{\sigma_{P,t}}{\sigma_t^2} \qquad \text{(A.1)} \\ \log(A) &= \overline{P} - \lambda \bar{t} \\ A &= e^{\overline{y} - \lambda \bar{t}} \end{aligned}$$

The exponential growth rate, λ is thus given by Equation A.1.

A.2 Covariance Matrix

Let P_i^n be the price of asset n at time t_i^n for times $i = 1, 2, ..., T$; and λ^n be the exponential growth rate of asset n.

For each time $i = 1, 2, ..., T$, we calculate the series of estimated prices using the exponential growth rate λ_n.

$$\forall i = 1, 2, ..., T \qquad E_i^n = A_n e^{\lambda_n t_i}$$

We then calculated the mean of the estimated series of prices for asset n, $\overline{E_i}^n$.

The covariance between assets x and y is thus

$$\sigma_{x,y} = \frac{1}{T} \sum_{i=1}^{T} E[P_i^x - \overline{E_i^x}]E[P_i^y - \overline{E_i^y}]$$

Note that the series of historical prices for both assets x and y need to be from the same time period.

The covariance matrix Σ is thus

$$\begin{pmatrix} \sigma_{1,1} & \sigma_{1,2} & \cdots & \sigma_{1,T} \\ \sigma_{2,1} & \sigma_{2,2} & \cdots & \sigma_{2,T} \\ \cdots & \cdots & \cdots & \cdots \\ \sigma_{T,1} & \sigma_{T,2} & \cdots & \sigma_{T,T} \end{pmatrix}$$

Appendix B

Cross Evaluation Results

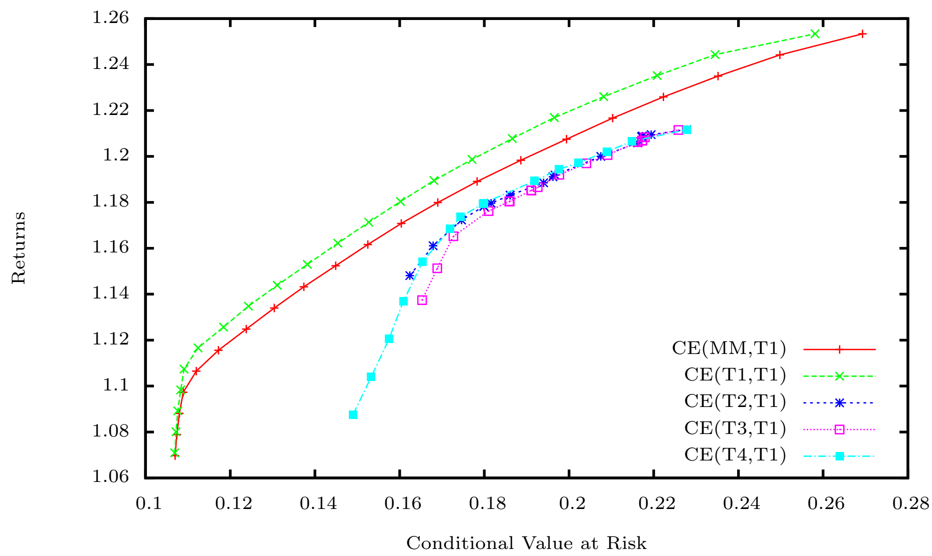

Figure B.1: Cross Evaluation of Implementing Various Strategies on Tree 1 (Without bounds)

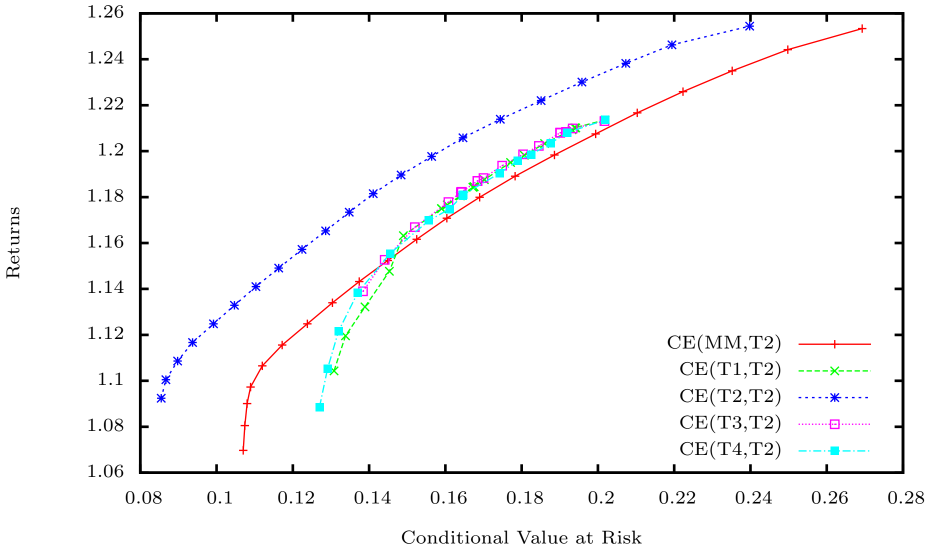

Figure B.2: Cross Evaluation of Implementing Various Strategies on Tree 2 (Without bounds)

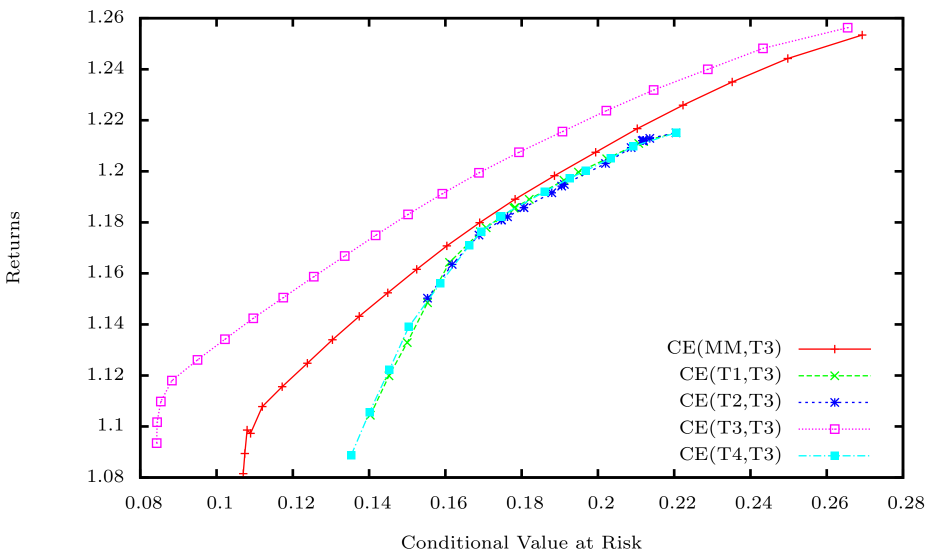

Figure B.3: Cross Evaluation of Implementing Various Strategies on Tree 3 (Without bounds)

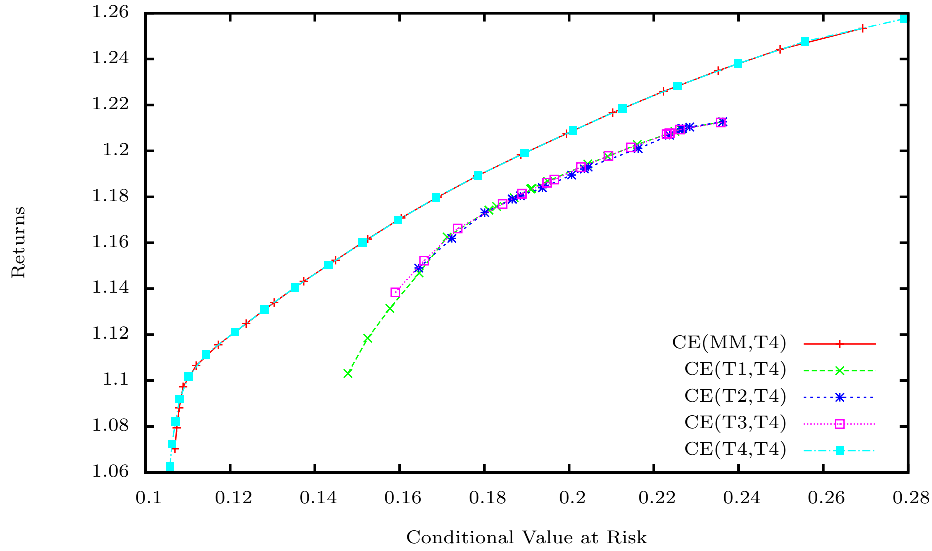

Figure B.4: Cross Evaluation of Implementing Various Strategies on Tree 4 (Without bounds)

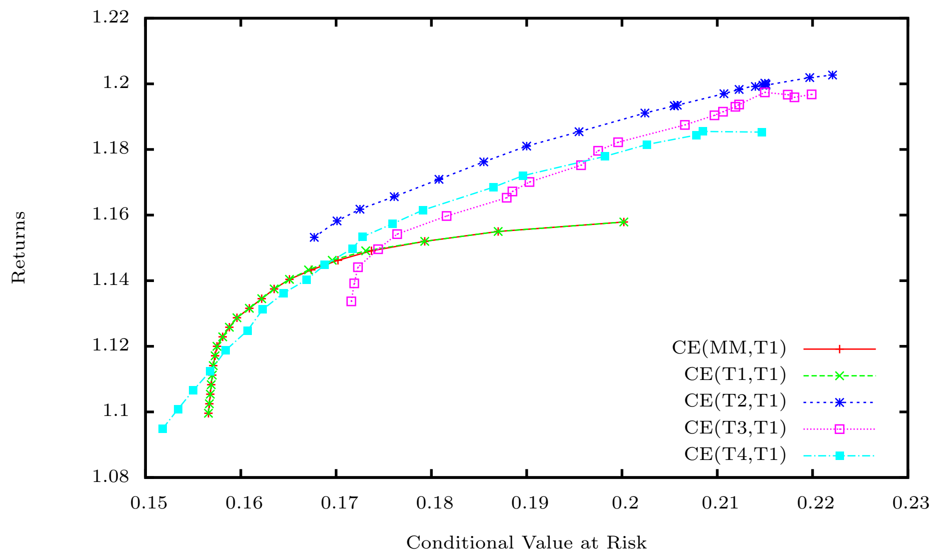

Figure B.5: Cross Evaluation of Implementing Various Strategies on Tree 1 (With bounds)

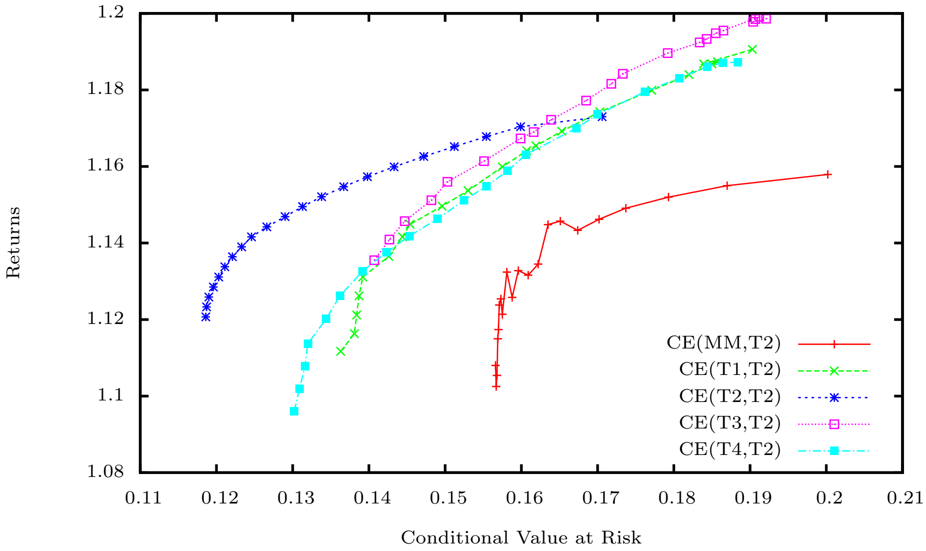

Figure B.6: Cross Evaluation of Implementing Various Strategies on Tree 2 (With bounds)

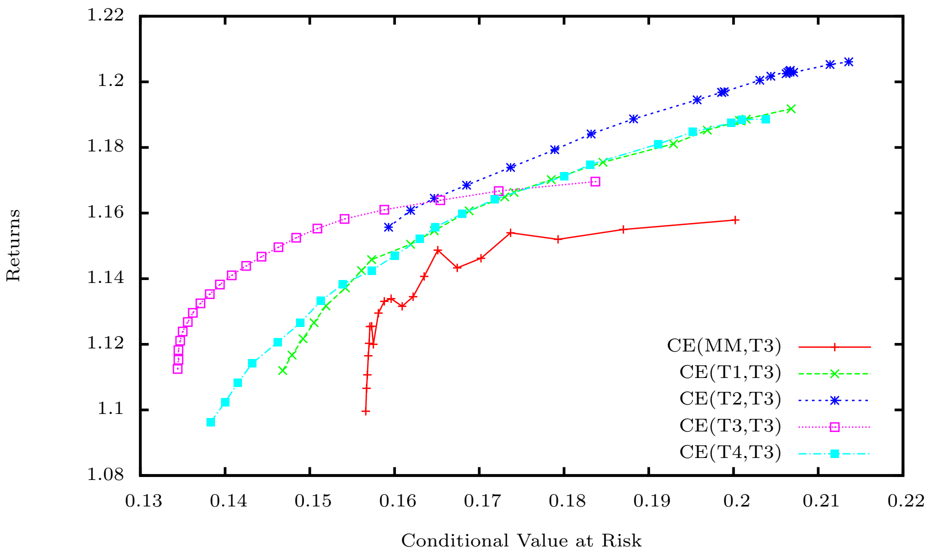

Figure B.7: Cross Evaluation of Implementing Various Strategies on Tree 3 (With bounds)

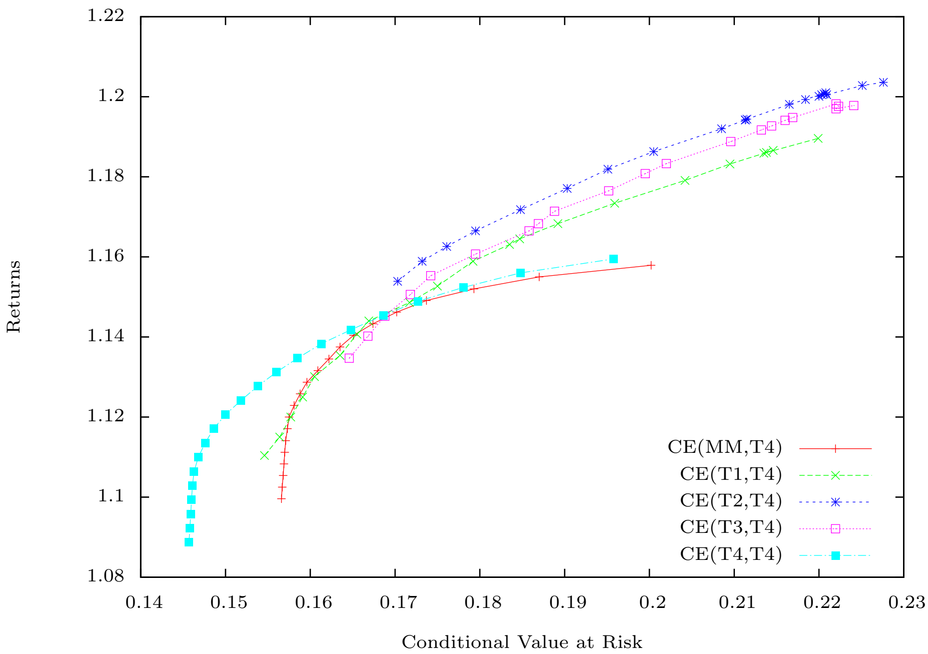

Figure B.8: Cross Evaluation of Implementing Various Strategies on Tree 4 (With bounds)

Appendix C

List of Assets

List of FTSE 100 assets used for backtesting

- Amec
- Aviva
- Barclays
- Bg Group
- Bp
- British Airways
- Diageo
- Glaxosmithkline
- Home Retail Group
- Itv
- Kingfisher
- Marks & Spencer Group
- Rio Tinto
- Reuters Group
- Rolls-Royce Group
- Royal Dutch Shell B
- Sainsbury (J)
- Schroders
- Tesco
- Vodafone Group

List of Symbols and Abbreviations

Symbol	Description	Definition
n	Number of Assets	page 3
ξ_n	Rate of Returns of Asset n	page 3
$E[\xi_n]$	Expected Rate of Returns of Asset n	page 3
w_n	The asset balance	page 3
σ_n^2	Variance of asset n	page 4
$\sigma_{i,j}$	Covariance between asset i and j	page 4
Σ	Covariance matrix	page 4
P	Current portfolio position	page 4
R	Required Return	page 5
ΔV_t	The profit and loss in time period t	page 8
ζ_α	The value at risk with a $\alpha\%$ confidence level	page 7
$CVaR_\alpha$	The conditional value at risk with a $\alpha\%$ confidence level	page 10
$WCVaR$	The worst-case conditional value at risk with a $\alpha\%$ confidence level	page 16
$[k]^+$	$max\{0, k\}$	page 10
$\mathcal{N}$	The set of all nodes in the scenario tree	page 12
$\mathcal{N}_0$	The root node of the scenario tree	page 12
$\mathcal{N}_t$	The set of nodes of the scenario trees representing possible events at time t	page 12
$\mathcal{N}_T$	The set of terminal nodes of the scenario tree	page 12
$\mathcal{N}_I$	The set of interior nodes of the scenario tree. $\mathcal{N}_\mathbf{I} \equiv \mathcal{N} - \mathcal{N}_\mathbf{0} \cup \mathcal{N}_\mathbf{T}$	page 13
$\mathbf{e}$	An event (a node in the scenario tree). $\mathbf{e} \equiv (s, t)$	page 12
s	Index denoting a scenario (a path from the root node to a leaf node)	page 10
t	Index denoting a time period	page 12
$a(\mathbf{e})$	ancestor of event $\mathbf{e}$	page 12
S	Number of Scenarios	page 10

Symbol	Description	Definition
$P_{\mathbf{e}}$	Probability of event e occuring	page 10
$\mathcal{K}$	The set of Scenario Trees	page 15
$\mathcal{N}^k$	The set of nodes of the scenario tree k	page 16
$\mathbf{w_t}$	The asset balances at time t	page 13
$\mathbf{b_t}$	The assets bought in time t	page 13
$\mathbf{s_t}$	The assets sold in time t	page 13
c_b	Unit transaction cost to buy assets	page 13
c_s	Unit transaction cost to sell assets	page 13
$\mathbf{w^U}$	Upper bound for the portfolio balance	page 13
$\mathbf{w^L}$	Lower bound for the portfolio balance	page 13
b^U	Upper bound for the number of assets that could be bought	page 13
s^U	Upper bound for the number of assets that could be bought	page 13
$\mathbf{1}$	(1,1,1,...,1)'	page 14
$\mathbf{P}$	Initial Portfolio	page 14
W_0	Initial Wealth of the portfolio	page 14
λ	The exponential growth rate of the asset	page 53

List of Figures

List of Tables