Optimizing Long-Short Portfolios

Results from a study of Markowitz Optimization Models

Roy Visser

Master thesis Business Mathematics and Informatics

Shell Pensioenfonds Beheer B.V.

Braillelaan 9

2289 CL Rijswijk

Vrije Universiteit, Amsterdam

Faculty of Sciences

Business Mathematics and Informatics

De Boelelaan 1081a

1081 HV Amsterdam

June 2005

Preface

The final stage of the Business Mathematics and Informatics (BMI) study at the Vrije Universiteit (VU) in Amsterdam consists of an obliged internship. I did this internship at ORTEC bv. During my internship I worked on a problem provided by the European Equity Team of the Dutch pensionfund of Shell, called Shell Pensioenfonds Beheer B.V. (SPB).

The European Equity Team manages a part of SPB's Asset Portfolio. One element of their European equity portfolio is a long-short portfolio. In my internship I have tried to answer the question how to optimize this long-short portfolio in terms of expected return and risk taking transaction costs into account.

I would like to thank ORTEC and SPB for giving me the opportunity to do research on this subject. Furthermore, I would like to give special thanks to the following people:

- Peter Kolthof for acting as supervisor from SPB, Aad van der Vaart and Harry van Zanten for acting as supervisor and second reader from the VU.
- Hicham Najem and Laurens de Prez from the European Equity Team for all the information, comments and critical notes they gave during the weekly progress meetings and during the finalization of my Thesis.
- My roommates at SPB; Eugene Braam, Martien van der Meer and Chris Chan, for they moral support and regular supply of good ideas, humour, coffee, tea and fresh water.
- André van Vliet from ORTEC for giving me the opportunity to do my internship together with SPB.

June 2005,
Roy Visser

Executive Summary

The research project described in this thesis is provided by the European Equity Team of Shell Pensioenfonds Beheer B.V (SPB). One part of the portfolio managed by the European Equity Team is managed with a long-short strategy. This strategy tends to work on paper, but in practice most of the profit disappears due to high turnovers and related market impacts and transaction costs. The objective of the research project is to find a trading strategy resulting in fewer turnovers. Therefore an answer to the following question has to be found:

Is it possible to make a trade-off between the costs involved with a trade and the extra return, possibly generated by the trade?

In order to answer this question, first an expected return model is developed. This expected return model is based on the ranks obtained by the Intra Sector Model (ISM). The ISM is a ranking model developed within SPB. After the development of this expected return model, an optimization model is built. This optimization model makes the actual trade-off between trading costs and expected return.

It appeared to be very difficult to develop an expected return model with high predictive power. Straight forward linear regression models on bucket numbers have equal or higher predictive power than more sophisticated models based on decision trees, K-nearest neighbour search and Markov chains. The results of regressions based on individual security ranks are comparable to those based on bucket numbers. Therefore the model based on bucket numbers is used in the remainder of the thesis.

This expected return model is one of the inputs of the optimization model. This optimization model is based on the classic mean-variance model proposed by Markowitz in 1952. Different types of technical constraints are incorporated in this optimization model. For example: cash neutrality constraint, beta tilt or beta neutrality constraint, liquidity constraints and diversification constraints. It is also possible to determine operational constraints. Furthermore transaction costs and market impacts are incorporated in this model. Transaction costs will be estimated with the PRISE model. Finally different optimization functions are defined, like the mean-variance utility function, the return only function (with an additional risk constraint) and the Sharpe Ratio.

This optimization model is implemented with a combination of Visual Basic and Matlab. Furthermore Microsoft Excel is used for the communication between the optimization model and the Backtester. The Backtester is one of the key applications within SPB for testing (long-short) portfolios on historical data.

The optimization model is tested with the optimization functions mentioned above. The results obtained by most of these optimization functions are comparable to each other. However, using the Sharpe Ratio leads to very Risk Averse Portfolios. The results of the other optimization models are compared with two trading strategies

used within the Backtester. The advantages and disadvantages of the different models are captured in the following table.

Naïve Strategy		Trading Rules		Optimization model	
+	Higher returns in high volatile markets	+	Savings on transaction costs	+	Savings on transaction costs
		+	Somewhat higher returns in high volatile markets	+	Higher returns in periods with low market volatility
-	High turnover between periods and thus a high loss in transaction costs			+	Properties like beta can easily be controlled by additional constraints
-	High volatility of realized returns	-	High volatility of realized returns	+	Low volatility of realized returns
-	Difficult to control e.g. beta tilt	-	Difficult to control e.g. beta tilt		
-	Low performance in low-volatility periods	-	Low performance in low-volatility periods	-	Lower returns in high volatile markets

Furthermore the results from comparison show that the trading rule model leads to a higher average return than the optimization models. However, the risk involved with the trading rule model is much higher. Therefore by using an optimization model, extra stability can be obtained without compromising return too much. Therefore investors who care about risk (like SPB) will be better of with the optimization models. Furthermore the optimization model enables the possibility to remove unwanted properties from the portfolios, like beta tilts or small cap biases. Therefore portfolios obtained by an optimization model are much more controllable.

Table of contents

1 Introduction

This thesis discusses research executed from July 2004 to April 2005 within Shell Pensioenfonds Beheer B.V. (SPB).

1.1 Shell Pensioenfonds Beheer B.V.

SPB manages the assets and pension administration of Shell Pensionfund Foundation (Stichting Shell Pensioenfonds, SSPF), Shell's pension fund for Dutch employees.

SPB invests the pension contributions as well as the other assets of the Shell Pension Fund. Furthermore, SPB provides services in the field of pension and asset management to other companies of the Royal Dutch/Shell Group or funds associated with them around the world. SPB is located in Rijswijk (ZH) and employs some 80 staff.

SPB manages some EUR 9.6 billion of pension liabilities, invests assets in excess of EUR 14 billion and pays annually EUR 500 million in pensions to over 18,600 retirees.

The research project is provided by SPB's European Equity Team, which manages a part of SPB's total assets. One part of the portfolio managed by the European Equity Team is managed with a long-short strategy. This strategy tends to work on paper, but in practice most of the profit disappears due to high turnovers and related market impacts and transaction costs. The objective of the research project is to find a trading strategy resulting in fewer turnovers.

1.2 Problem Description

Is it possible to make a trade-off between the costs involved with a trade and the extra return, possibly generated by the trade? This is the main question that will be answered in this thesis. In order to answer this question, some sub questions will be answered.

First an answer will be found to the following sub question. *Is it possible to develop an expected return model based on the ranks from the intra sector model (ISM)?* The ISM is a security ranking model developed within SPB, used for the construction of the long-short portfolio. Further details of this model will be discussed in the subsequent chapters. Once an answer has been found to this first sub question, the next question will be answered. *Is it possible to incorporate this expected return model in an optimization model in order to make a trade-off between trading costs and return?* This optimization model must have the following properties.

- The optimization model should be derived from a mean variance model
- It should take into account operational and technical constraints
- It should take into account a current portfolio when constructing a new portfolio. Consequently, it should take into account transaction costs to change from the current to the new portfolio.

This last property is added to make the trade-off between transaction costs and extra return, as stated in the initial research question.

Obviously, the expected return model will be implemented into a computer application. This application must be integrated within the application portfolio already available within SPB. Therefore, in this thesis the following question will be answered. *What are the possibilities for integrating the optimization model within the application portfolio already available within SPB?* The answer to this question will be given with a set of recommendations for future developments.

1.3 Setup of this thesis

Chapter 2 gives an overview of the strategy currently used within SPB. Chapter 3 gives an overview of some available literature on predicting returns. The theory from chapter 3 is applied in chapter 4 in order to build an expected return model. Chapter 5 discusses the theory of optimization models. In chapter 6, the optimization model used within this thesis will be developed. Chapter 7 discusses the implementation of this model. The results of the research will be discussed in chapter 8. At the end of (almost) every chapter, local conclusions can be found. The overall conclusions of the research can be found in chapter 9. Obviously the local conclusions will be summarized repeated in this chapter. Chapter 9 gives also recommendations for further development of both the expected return and the optimization model.

A reader with limited time should at least read the chapters four (development and results of the expected return model), six (development of the optimization model), eight (results of the optimizations) and nine (conclusions) to understand the research described in this thesis. Chapter two (actual strategy) contains necessary background information for anyone not familiar with the long-short strategies used within SPB.

2 The Actual Investment Strategy

2.1 Long-Short portfolio

2.2 Determining the long and short side

2.3 Determining the Ranking

3 Literature on expected return models

The literature study to expected return models will fulfil two targets. First it is meant to give an idea about which methods are available in literature and are applied in prior studies. Secondly, it will give a general idea about the performance to be obtained by such models. Therefore not all models discussed in this section will be applied in this research. For example, the model that estimates returns on a CAPM based model is only captured in this section as a reference for results obtained by models developed during this internship.

The general structure of this chapter is as follows. First some basic theory about efficient markets will be discussed. Based on the thoughts at the end of it, this discussion is meant as a justification for the development of an expected return model. Secondly existing expected return models are examined. These models will give a feeling about the explanatory power that can be obtained by such models. After these existing models, some basic techniques will be discussed. These techniques will be applied for return estimation in the subsequent chapter. After this discussion of techniques, the Bayesian approach to return estimation will be introduced. Because this approach will not be implemented in this thesis, the introduction will be rather briefly. Finally some conclusions will be drawn.

3.1 Efficient Markets

Before the different models are discussed, first the efficient market hypothesis has to be mentioned. According to Fama (1970), there are three forms of market efficiency, the strong, semi strong and weak form.

The weak form states that share prices fully reflect all information contained in past price movements. If this is the case, trading rules only based on share price history will not have any predictive power. Therefore it does not work to define trading rules based on share price history, as the future cannot be predicted in this way.

The semi strong form states that share prices fully reflect all the relevant publicly available information. This includes not only past price movements but also earnings and dividend announcements, rights issues, technological breakthroughs, resignations of directors, and so on. The semi-strong form implies that there is no advantage in analyzing publicly available information after it has been released, because the market has already absorbed it into the price.

Finally, the strong form states that all information available, including private, is already incorporated in the price of the security. This makes it even for insiders (for example the director of a company) impossible to make abnormal profits.

If this is true, the return process becomes a martingale process, which makes it impossible to predict any future return and unnecessary to perform active management of assets.

However, we shall see in the following discussion of literature that the theory about efficient markets is not (always) applicable to real markets. This is for example indicated by the theory about bounded rationality, which states that not all information is always reflected in the share price, because investors do have limited rational capabilities. They do not always examine all open options, simply because they are not capable to deal with all the information related to those options. Therefore, the investment process is not as rational as stated by the efficient market theory.

The theory of behavioural finance goes even further. This theory states that markets are not always rational because social behaviour plays a role. Furthermore decisions are not always made on a rational basis, other aspects (for example psychology) also plays a role. Therefore at least some agents on a market do not act rational.

3.2 Existing Models

3.2.1 Predicting returns with CAPM

In Bartholdy and Peare (2003) returns are predicted with model based on the Capital Asset Pricing Model CAPM. Because the CAPM is a widely known model, it will not be discussed in detail in this thesis. A rough description can be found in this section, a more detailed description can be found in numerous of textbooks.

In the Capital Asset Pricing Model CAPM excess returns are predicted as the product of the expected excess returns of the world market portfolio (a portfolio constructed with all stocks in the world) and the beta of a particular stock with the world market portfolio by the following formula.

$$E[r_{excess}] = \beta \cdot E[r_{wm}]$$

In this sense excess return (r_{excess}) is defined as the return of a certain stock or portfolio minus the risk-free rate. r_{wm} is the risk premium on the world market portfolio. Thus stocks with a relatively high beta have a relatively high expected return.

Bartholdy and Peare first give a method to obtain unbiased return estimates based on a proxy for the world market portfolio (a proxy is used because the world market itself is unobservable). This method is the method proposed by Fama and MacBeth in 1974. After the construction of this method, they test their CAPM-based expected return model on several different portfolios of securities in order to estimate yearly returns. The portfolios used are both the securities from both the equally weighted and value weighted CRSP index, the Morgan Stanley index and the Standard and Poor index. The equally weighted CRSP index leads to the highest R^2, namely 3%. Although this indices are rather different from the one tested in this thesis, the R^2 of 3% give a general idea about the possibilities of expected return models.

3.2.2 Predicting Returns based on analyst forecasts

Brav, Lehavy and Michaely (2003) describe a scheme in which annual returns are predicted based on Value Line and First Call forecasts.

They use their prediction in order to test some of the relations of the CAPM model with ex-ante data instead of ex-post data. Therefore, predicting returns is not the primary target of their research, and so they omit the check of the expected returns on realized returns; Contrary they show that their expected return has properties which should be expected by CAPM (e.g. high beta leads to higher expected return).

Although the performance of the expected return model is not reported, and the applicability for estimating monthly returns can be questioned, the authors show some interesting relations between their expected returns and other descriptive variables, like beta, momentum and market cap.

3.3 Linear regression models based on ranks

One of the simplest methods that can be applied in order to create an expected return model based on an underlying ranking model, is the linear regression method. Because the technique behind linear regression techniques is widely known, it will not be discussed in this thesis. It can be found in numerous textbooks on statistics, e.g. De Gunst and Van der Vaart (2001).

If linear regression techniques are used for the creation of an expected return model, it should be noticed that, in order to predict returns in period t = T, only data available until period t = T-1 can be used for the determination of the model.

There are different possibilities for applying linear regression techniques. For example, the regression could be based on the final ranks obtained from the ranking model or on the underlying composite ranks. This will be illustrated with the following example.

Example 3.1

Consider a small dataset with the following securities. The (excess) returns at t=T-1 are already known, those at t=T not.

Security	Period	Rank	C1	C2	Exc. Return
A	T-1	0	0	0.33	10%
B	T-1	0.33	0.33	0.66	5%
C	T-1	0.66	1	0	-8%
D	T-1	1	0.66	1	-2 %
A	T	0.33	0.33	0.66	??
B	T	0	0	0	??
C	T	1	0.66	1	??
D	T	0.66	1	0.33	??

Table 3-A: Linear Regression Example, Data table

If we apply a linear regression to this dataset, based on the final ranks, the resulting we will get the following model:

Exp. Return = -0.15 * rank + 0.08

If we apply this to the dataset at t=T, we will get the following results.

Security	Result
A	-0.15 * 0.33 + 0.08 = 3%
B	-0.15 * 0 + 0.08 = 8%
C	-0.15 * 1 + 0.08 = -7%
D	-0.15 * 0.66 + 0.08 = -2%

Table 3-B: Linear Regression Example, Results

The performance of the expected return model will be determined by the differences between those expected returns and the realized returns over period T.

3.4 Using Artificial Intelligence techniques

In their overview of modern time methods, Focardi, Kolm and Fabozzi (2004) discuss the use of data mining and machine learning techniques. They give a short overview of common techniques, which can be applied to finance and discuss two applications in more detail. The theory behind those common techniques is discussed in Witten, Frank (2000) and Mitchell (1997) in more detail. In the following sections, an overview of techniques will be given, together with examples, which show how those techniques can be applied in finance. One common property of AI models is that they have to be trained on a dataset with known resulting values (in this case realized returns). In this training phase, the actual prediction model (for example a decision tree) is created. After this, the prediction model can estimate returns for new data. The performance of the model will be measured on a test set. This test set may only contain new data.

3.4.1 Decision Trees

Decision Trees are systems, modelled as a tree, which can be used for classification purposes. A decision tree consists of a set of hierarchical nodes. In each node, the dataset is divided into two (or more) disjunctive subsets. When the model is applied to return prediction, these subsets consist of a number of securities.

The data within a subset is more or less the same (with respect to the value that should be predicted), but data from different subsets does not have that common factor (again with respect to the value to predict). After the split, each subset follows another branch. This process continues in the next nodes. At the end of the tree, small subsets of the complete dataset end up in the leaves (end nodes) of the tree. Those leaves assign a classification to all the data in the corresponding subsets. In the case of an expected return model this classification corresponds with an expected return.

Each path in the decision tree can be described as a simple if-then-else relationship. E.g. if we take look at the tree displayed below, one can come up with a rule 'If Q1 is yes AND Q2 is no than D2'. This illustrates again the point that decision trees are able to describe conditional (If … Then … Else …) relations in the dataset.

The following example will show how a decision tree can be applied to predict returns for our dataset, created in the previous example. It will not discuss how the tree is created, this is a very technical process, which is explained in Mitchell (1997) and is already implemented in various software packages. The example will show how expected returns are generated once a decision tree has been created.

Example 3.2
Assume that we have obtained the following decision tree. This decision tree is created on data available in periods before t=T.

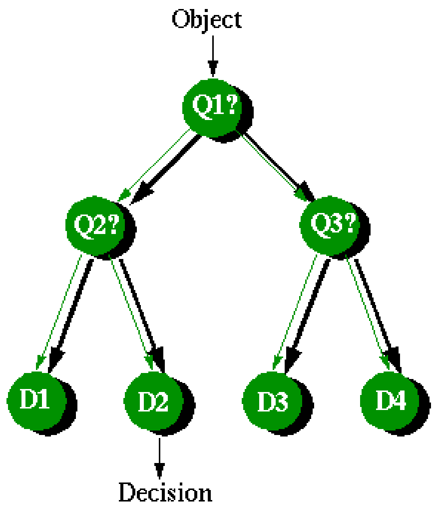

Figure 3-A: Decision Tree

Assume that the questions (Q) and decisions (D) in this tree are as listed in the following table.

Node	Value
Q1	C1 > 0.4
Q2	C2 < 0.5
Q3	C2 > 0.7
D1	-5%
D2	-2%
D3	3%
D4	8%

Table 3-C: Decision Nodes

When using this tree, the returns, predicted for the dataset at t=T, will become the following:

Security	Period	Rank	C1	C2	Class
A	T	0.33	0.33	0.6	D4 (8%)
B	T	0	0	0	D4 (8%)
C	T	1	0.66	1	D2 (-2%)
D	T	0.66	1	0.33	D1 (-5%)

Table 3-D: Results of Decision Tree

Contrary to the linear regression models, this approach will lead into a discrete set of expected returns. The next model will however create a continuous set of expected returns.

3.4.2 Regression trees

Regression trees are almost equal to the decision trees. The difference is the process in the end-nodes. Instead of a single value, which is assigned to the data corresponding to a certain end node, the regression tree assigns a regression model to that data. Contrary to the linear regression models described earlier, the regression tree creates a conditional model; securities with different properties can be classified with different regression models. This is illustrated with example 3.3.

Example 3.3

Assume the same decision tree as in example 3.2, but now with the following regression models in the decision nodes.

Node	Regression model
D1	-0.05*C2 +0.01*C1 – 0.05
D2	-0.01*C2
D3	0.05*C1 + 0.10*C2 – 0.5
D4	-0.1*C1+0.08

Table 3-E: Models of Regression Tree

The prediction on the dataset will now become the following

Security	Period	Rank	C1	C2	Class (Prediction)
A	T	0.33	0.33	0.6	D4 (-0.1*0.33+0.08 = 4.7%)
B	T	0	0	0	D4 (-0.1*0+0.08 = 8%)
C	T	1	0.66	1	D2 (-0.01*1=-1%)
D	T	0.66	1	0.33	D1 (-0.05*1+0.01*0.66-0.05 = -3.4%)

Table 3-F: Regression Tree, Results

Contrary to the original decision tree, the expected return prediction of security A and B (which both end up in node D4) differs from each other.

3.4.3 K-nearest neighbour search

In nearest neighbour search, the instances are classified based on previous learned instances, which are almost equal to the new instance. In the context of return prediction an instance can be seen as a security in a certain period, with all the corresponding descriptive values and a known or unknown return. First the model must be, as all AI models, initiated or trained with a number of instances with a known result.

After this training phase the return of new instances will be predicted based on the return of instance(s) which are 'quite similar' to the new instance. Which instances are quite similar is determined by a distance function, which calculates the difference (similarity) between the new instance and the already known instances. After this, a constant number K of nearest instances are selected, called the K-nearest neighbours.

The predicted return of the new instance will be equal to the (weighted) average of those K neighbours, as illustrated in the next figure. In example 3.4 a 2-Nearest Neighbour process is adapted to our dataset.

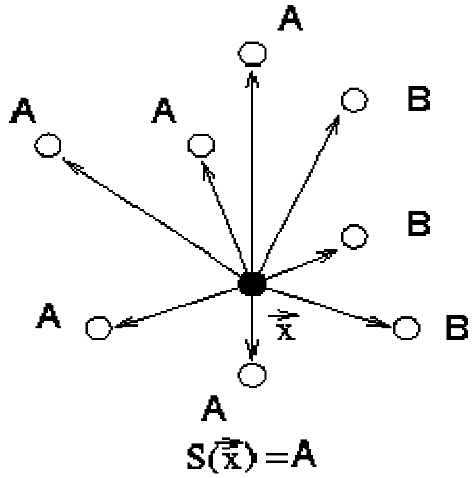

Figure 3-B: K-Nearest Neighbour Search

Example 3.4

Assume again the following dataset. The 2-Nearest Neighbour model will be trained with those instances known at t=T-1.

Security	Period	Rank	C1	C2	Exc. Return
A	T-1	0	0	0.33	10%
B	T-1	0.33	0.33	0.66	5%
C	T-1	0.66	1	0	-8%
D	T-1	1	0.66	1	-2 %
A	T	0.33	0.33	0.66	??
B	T	0	0	0	??
C	T	1	0.66	1	??
D	T	0.66	1	0.33	??

Table 3-G: Nearest Neighbour search, Data table

The distance between a new instance and another (old) instance will be calculated by the Euclidean distance function based on the composite values C1 and C2. This is:

$$D = \sqrt{(C1_{new} - C1_{old})^2 + (C2_{new} - C2_{old})^2}$$

Now, a new instance, security D at period t=T will be classified. The following table shows the distances to the four already known instances

Instance	Distance	Exc. Return	Neighbour
A, T-1	$\sqrt{(1-0)^2 + (0.33-0.33)^2} = 1.00$	10%	
B, T-1	$\sqrt{(1-0.33)^2 + (0.33-0.66)^2} = 0.747$	5%	X
C, T-1	$\sqrt{(1-1)^2 + (0.33-0)^2} = 0.33$	-8%	X
D, T-1	$\sqrt{(1-0.66)^2 + (0.33-1)^2} = 0.751$	-2%	

Table 3-H: Determination of Nearest Neighbours

The old securities B and C are closest to the new security D. Those two securities will be selected in the 2-Nearest Neighbour algorithm. If the prediction of the new return is an unweighted average, the predicted value will be the average of 5% and –8%. This is –1.5%. If a weighted average is calculated, it is common sense to weight the instances by 1/distance. In this case the prediction becomes

$$\frac{\frac{1}{0.33}\cdot -8\% + \frac{1}{0.747}\cdot 5\%}{\frac{1}{0.33}+\frac{1}{0.747}} = -4.02\%$$

In the weighted case, the prediction is more based on the relatively close instance [C,T-1].

3.4.4 Neural Networks

A Neural Network is a system modelled after the neurons (nerve cells) in a biological nervous system. A neural network is designed as an interconnected system of processing elements, each with a limited number of inputs and outputs. Rather than being programmed, these systems learn to recognize patterns. Therefore the relation between the inputs of the network and the output of the network does not have to be known in advance.

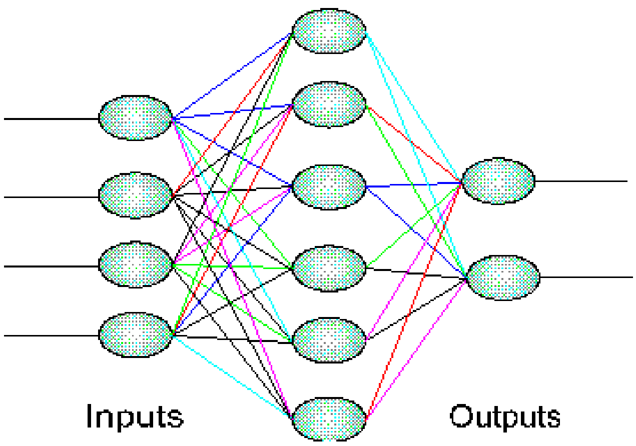

Figure 3-C: Neural Network

The technique of Neural Networks is a very complex technique. Developing an expected return model based on this technique could be a research on itself; therefore it will not be applied in this research.

3.5 Predicting returns with a Markov Chain

The usage of a decision tree leads to conditional predictions of expected returns. Another method that results in conditional predictions is the usage of a Markov Chain. This method is proposed by McQueen and Thorley (1991).

They construct a Markov Chain for testing the random walk hypothesis. This hypothesis states that the stock returns should follow a random walk. If this hypothesis is true, the probability of a high (annual) return should be independent of prior returns. The same must hold for the probability of a low (annual) return.

Their Markov Chain has two states, high return (1) and low return (0). The division between high and low returns is made by the average return over the prior 20 years, but the sensitivity analysis shows that the model is robust to the choice of this divider.

Based on this division, a time series of zeros and ones is created, representing historical annual returns. After that a second order Markov Chain (using the previous two return states for predicting the new return) is fitted on this series, with transition probabilities calculated with the log likelihood function.

The objective (testing the random walk hypothesis) and the result of the paper (there exists a significant relationship between prior return states and the new return) does not correspond with the objective of the research to expected returns. However, the Markov Chain based predicting method could be adapted in order to use it with a ranking model. How the Markov Chain is adapted to the usage of ranks will be discussed in section 4.4 of this thesis. The results of this process will be discussed in the same section.

3.6 Bayesian Approach

According to Focardi, Kolm and Fabozzi (2004), the Bayesian approach makes some improvements to the inputs of the mean-variance model. The expected return model is one of the major inputs to the Mean-Variance framework. Therefore, in this section only the usage of the Bayesian approach for expected returns is mentioned. The possibilities for improving other inputs are discussed after the introduction of the various optimization models in chapter 5.

In the classic optimization methods, expected returns and covariances matrices are regarded as fixed (but unknown). Point estimates of these values are used in the optimization model. Of course the results of the mean-variance optimization are influenced by the errors in these estimates. (See also the review of the Chopra and Ziemba article about the sensitivity to errors, discussed in chapter 5)

In the Bayesian approach not only the means and (co)variances of the returns are taken into account, the expected returns are considered to have an unknown distribution. By using the so-called Bayes rule this unknown distribution (called the posterior distribution) is calculated based on a prior distribution and new data.
The prior distribution of the expected return represents the investor belief about a certain stock.

In the Black-Litterman model, discussed in many articles, for example He and Litterman (1999), the posterior distribution is calculated by combining the CAPM equilibrium (or another market equilibrium) with one or more views of an investor, which are expressed as an absolute or relative deviation from the equilibrium. The expected return from the Black-Litterman model (which is the mean of the posterior distribution) can also be used as the input for the classic mean-variance optimization model.

3.7 Conclusions

If the strong or even the semi-strong form of the efficient market hypothesis is true, it makes no sense predicting excess returns. No model will be able to predict returns successfully. Consequently, it will not explain any of the variance observed in the returns (R^2 almost equal to 0). However, empirical evidence does not support the efficient market hypothesis. Also the amount of literature available on expected return modelling indicates that the efficient market hypothesis is not widely supported. On the other hand, the results of the CAPM-model (R^2 around 3%) show that it will not be likely to explain much of the variance in returns.

The existence of a non-normal expected return distribution is indicated by various articles. However, within this thesis returns are assumed to be jointly normal distributed. Therefore only the first (mean) and second (volatility) moment of the return distribution will be estimated (volatility will be estimated with historic volatilities, see chapter 6 for further details). Higher moments of the return distribution will not be used in this research. This implies that for example skewness and fat tails of the return distribution are ignored. Obviously this assumption limits the possibilities for the expected return and optimization model. On the other hand, the assumption of normal distributed returns provides a relatively simple starting point for the research to these models.

We have seen that there are a number of different approaches (CAPM, AI based models, Markov Chains, Bayesian methods) for predicting returns. In this thesis relatively simple expected return model will be developed, giving a point-estimate for the expected return. Therefore the following techniques will be applied:

- Linear Regression models
- Decision Tree
- Regression Tree
- K-Nearest Neighbour search
- Markov Chain

The Bayesian method could also be interesting but will not be applied in this research. Although it is possible to obtain point estimates by using the first and second moment of the posterior distribution, the general purpose of this approach is beyond the scope of this thesis. However research to this Bayesian approach is certainly recommended as a future development.

4 Developing an Expected Return Model

4.1 Data Description

4.2 Linear Regression Models

4.3 Predicting future returns with Artificial Intelligence

4.4 Predicting returns with a Markov Chain

4.5 Conclusions

5 Portfolio Optimization Literature

5.1 Introduction

In the second part of the research, an answer to the question how an optimal long-short portfolio should be selected will be given. The expected return model, which is discussed in chapter four, is one of the inputs of this optimization problem.

This chapter will discuss some of the theory available about Portfolio Optimization problems in general and in the special case of a long-short portfolio. The discussion of this literature is meant to give a general idea about what kind of studies are available about optimizing portfolios (or the special case of long-short portfolios). First the most basic models are mentioned in this section. Secondly some modern times approaches are discussed.

5.2 The Classic Mean-Variance Models

Portfolio optimization models looks at the best way for the portfolio manager to allocate his wealth between different securities or asset classes. The best way means in this sense taking into account expected portfolio return (mean) and the risk (variance) and making a trade off between risk and return. The most traditional model for this problem is the mean-variance model proposed by Markowitz (1952). First a description of this model will be given.

5.2.1 Markowitz

In his famous 1952 paper, Markowitz first describes why both return and risk should be taken into account. He shows that an optimal portfolio, created without taking risk into account, will only contain the stock with the highest expected return. However, if risk is taken into account, the model will penalize this portfolio for being too risky. Instead of using just one security, a set of securities with suitable risk properties (for example, low expected volatility, negative correlations) will be used. Therefore, the resulting portfolio will be more diversified.

After this observation, Markowitz describes how the expected return of the portfolio (PFR) and the variance (PFV) of this return should be calculated. This is done by the following formulas

$$PFR = \sum_{i=1}^{N} x_i r_i$$

$$PFV = \sum_{i=1}^{N} \sigma_{i,j} x_i x_j$$

In which the holding of security i in the portfolio is denoted by x_i, the expected return of security i by r_i, the covariance between security i and j by $\sigma_{i,j}$ and the number of securities by N.

The covariance of security i and j can be written in terms of the volatility of a security i and j and the correlation between security i and j. This can be done by the following formula.

$$\sigma_{i,j} = \rho_{i,j}\sigma_i\sigma_j$$

In which the correlation between security i and j is denoted by $\rho_{i,,j}$ and the volatility of security i by σ_i.

In his 1952 paper, Markowitz does not come up with a method to optimize the portfolio with respect to these two measures. Instead of this, he gives some graphical representations of the optimization problem, for the special cases of 3 and 4 securities. Furthermore he mentions that there are techniques available for optimizing the Mean-Variance models. However, much other literature is available about this optimization task.

5.2.2 Risk and return

Because maximizing expected return and minimizing risk are two opposite targets, the Mean Variance models has multiple objectives. These objectives can be combined by using of a utility function. A typical utility function is given by Womersly and Lau (1996)

$$f(x) = PFR - \frac{PFV}{t} = \mu^T x - \frac{x^T Cx}{t}$$

The value t is a risk tolerance parameter, which is set by the portfolio manager in order to specify his risk-aversity. The vector μ is the vector of expected returns for all N securities and the N-by-N matrix C is the covariance matrix of these securities. The vector x (size N) contains the holdings for all securities. By this the PFR and PFV elements in this formula are just the matrix notation of the same elements from Markowitz (1952) and so

$$f(x) = PFR - \frac{PFV}{t} = \sum_{i=1}^{N} x_i r_i - \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\sigma_{i,j}x_i x_j}{t}$$

Another way to take both risk and return into account is to minimize risk given a fixed amount of the return to be obtained. In this case the amount of return is set within a constraint and the target of the optimization is to minimize $x^T Cx$ (which is the portfolio variance). If this model is solved for different amounts of returns it will give the investor an idea about the relationship between risk and return.

5.2.3 Sensitivity Analysis

Both Kallberg and Ziemba (1984) and Chopra and Ziemba (1993) did research on the sensitivity of the mean-variance model. The sensitivity to errors in means, variances and covariance is investigated. Chopra and Ziemba will be referred, because they included many of the research done by the Kallberg and Ziemba in their article (in fact it is an extension to it). Furthermore they suppose some 'solutions' to deal with these kinds of problems.

Chopra and Ziemba (1993) gave a sensitivity analysis of the Mean-Variance optimization proposed by Markowitz. Their research concentrates on two types of errors, errors in the expected returns and errors in the covariance matrix (based on variances and correlations) of the securities. Their research is a quantitative analysis of the research done by Kallberg and Ziemba (1984). They use an optimization model which comes down to a basic Markowitz optimization model with optimizes the utility function specified in the previous section under the constraint of positive weights (long only optimization).

This gives the following optimization model

$$\text{maximize} \qquad \sum_{i=1}^{N} x_i r_i - \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} \sigma_{i,j} x_i x_j}{t}$$

$$\text{such that} \qquad x_i \geq 0, \sum_{i=1}^{N} x_i = 1$$

In order to measure the sensitivity to errors in forecasting returns, variances and covariances, they make use of the relative loss in Cash Equivalent (CE). The CE of a risky portfolio is the amount of risk free cash that provides the same utility as the risky portfolio. By comparing Cash Equivalents of portfolios, the investor's risk tolerance is taken into account. The percentage cash equivalent loss if holding portfolio x instead of the optimal portfolio o is given by

$$CEL = \frac{CE_o - CE_x}{CE_o}$$

They use ten years of monthly data from ten randomly selected American stocks. Assume that returns, variances and covariances in a certain period are equal to the historic averages measured in this time span. Obviously the highest portfolio return in this period will be obtained when the portfolio is optimized on these values. The Cash Equivalent of this portfolio is denoted with CE_O.

However, these values are not known in advance, therefore we have the risk of a forecasting error. In order to get a set of erroneous return forecasts, the returns are changed slightly with a normally distributed error term with mean zero and standard deviation σ. The portfolios are constructed on this set of erroneous returns. These portfolios are suboptimal because they are not based on the correct inputs. Their cash equivalent CE_x will be calculated on the correct (not erroneous) values. The same method is repeated for variances and covariances. This results in a cash equivalent loss corresponding to a change in a certain input parameter and a certain standard deviation of the error term. Because this process is stochastic (it depends on the realization of error terms) it is repeated 100 times for each setting.

Obviously the historic estimates are not the correct values for constructing the most optimal portfolios. However, as long as the erroneous returns are based on the same estimates and a random error term, the historic estimates can be used for this analysis.

For a risk tolerance of t=50, the average CEL for an error in the returns differs from 0.66% (for an error term with standard deviation 0.05) to 10.16% (for an error term with standard deviation 0.20). For an error in the variance the CEL differs from 0.05% to 0.90% and for an error in the covariance the CEL differs from 0.02% to 0.47%, so the effect of an error in the returns is 11 times bigger then the effect of an error in the variances. If the same procedure is repeated for a risk tolerance of t=25 respectively t=75 the effect of an error in the returns is 3.2 respectively 21.4 times bigger than an error in the variances. The effect of errors in variances is between 1.7 and 2.7 times bigger then the effect of an error in covariances.

Chopra and Ziemba recommend setting the return forecast all equal in order to focus on minimizing portfolios risk and so the variance does not have the error-in-means problem. But if there are superior estimates of the means, the estimation of the variances and covariances becomes less important and it may be acceptable to use just historical values.

5.3 Modelling Downside Risk

5.3.1 Downside nature of risk

One problem with minimizing risk or maximizing a utility function in the Mean-Variance framework proposed by Markowitz is the downside nature of risk, which means that an investor or a portfolio manager is indeed interested in minimizing the risk of a lower portfolio return (downside risk), but has no interest in minimizing upside risk.

Womersley and Lau (1996) derive in section three of their article two different solutions for this problem, the Semi-Variance model and the Skewness model. Both solutions are specified in a long only context. Because the Skewness model results in a nonlinear and non-convex model, it is hard to solve and will only give a local instead of a global solution (see Hillier and Lieberman (1980) for a more detailed description of this problem). Therefore the Skewness model will not be described in this thesis. Only a (short) description of the properties of the Semi-Variance model will be given.

5.3.2 Scenarios

The Semi-Variance model is constructed on a scenario based expected return. Such an expected return is constructed as follows. Define s different representative economic scenarios. All these scenarios have a certain probability to occur, and all securities have a conditional expected return, given a certain scenario will occur. This gives us an s by n matrix R (with s the number of scenarios and n the number of securities) of scenario returns. The expected return for a security is then simply the average over all scenario returns weighted by the probabilities that the scenario return will occur (which is the probability that the scenario will occur)

Based on this matrix R the covariance matrix C can be calculated. With this covariance matrix and the scenario based expected returns the default Mean-Variance model can be constructed.

5.3.3 The Semi-Variance Model

In the determination of this covariance matrix described above, the standard definition of variance (which is $E[(y-\mu)^2]$ is used. This means that both upward and downward differences from the (expected) mean are penalized in the same way. By taking only downward differences into account, the model is no longer penalized for getting unexpected high returns. Therefore the variance term is replaced by $E[\min(0,y-\mu)^2]$. The resulting model is called the semi-variance model. Because the semi-variance is still convex, the resulting problem is a convex programming problem. Any local optimum for this problem is a global optimum.

5.4 Portfolio Optimization in the Long-Short context

The optimization proposed by Markowitz (1952) is defined in the long only context, which means that for each security the holding should be positive. Jacobs, Levy and Starer(1999) give an overview on long-short portfolio management in general. One of the elements of this overview is the adjustment of the Markowitz model to long-short portfolio management. The main message of their article is to optimize both the long and the short portfolio in one optimization, instead of creating a suboptimal portfolio by optimizing the long and short leg separately.

In this description, the model as proposed in the article will be adapted to the long-short optimization problem to be solved at SPB, Furthermore the notation will be changed into the one used by Markowitz (1952) and Womersly and Lau (1996).

In the model a utility function with the following form is used.

$$f(x) = PFR - \frac{PFV}{2t}$$

In the article, the portfolio is optimized on absolute returns instead of excess returns. This is because the holdings are modelled in such a way that not all the wealth available for portfolio construction has to be invested in risky securities. It is also possible to invest a fraction of the available wealth on the risk free rate, or even to invest more then the total available wealth on the risk free rate in case of short only management.

Then the absolute returns are decomposed in the risk free rate and an expected excess return (expected absolute return minus (expected) risk free rate). The portfolio return equals the risk free rate, gained over all the wealth available, and the expected excess return, gained over the holdings of the corresponding securities. By this, the portfolio return is given by

$$PFR = r_F + \sum_{i=1}^{N} x_i r_i$$

In which r_F is the risk free return and r_i is the excess return of security i. Because the risk free rate is only a constant in the utility function, it will not influence the optimal solution and therefore it can be left out of the utility function. Furthermore, in our case we have equal long and short legs and so we do not even obtain the risk free rate. (To be precise, we gain the risk free rate on the long leg and we loose it again on the short side)

Because of this, the optimization on absolute returns and the optimization on excess returns will be the same optimization and so we can write the return part of the utility function as a function equal to the original Markowitz formula.

$$PFR = \sum_{i=1}^{N} x_i r_i$$

The risk measure in the utility function is also equal to the Markowitz formula, so the optimization becomes the same as proposed by the Womersly and Lau (1996) introduction (with a slightly different form of utility function) but without the non-negativity constraints for the holdings X_i.

Jacobs, Levy and Starer also give the trivial solution for the optimization problem with a minimal set of constraints. In the remainder of the article some basic constraints (like cash and beta neutrality) are examined and the optimal solutions under these constraints are specified. However the authors do not mention technical constraints (like the inability to short some specific stocks) and their influence on the optimal solution. Also transaction costs are not taken into account. Therefore these solutions are not applicable in situations described in this thesis.

5.5 Modern Times Methods

Focardi, Kolm and Fabozzi (2004) give an overview of some modern time approaches to portfolio optimization. One of them, the robust portfolio allocation approach will be discussed here

5.5.1 Robust Portfolio Allocation Approach

One of the problems with the basic Mean-Variance method is the use of point estimates of expected returns and (co)variances. By using these point-estimates the resulting optimization is deterministic. The goal of the Robust Portfolio Allocation framework is to get a portfolio, which will perform well under a number of different scenarios instead of one scenario. In order to obtain such a portfolio the investor has to give up some performance under the most likely scenarios in order to have some insurance for the less likely scenarios. In order to construct such portfolios an expected returns distribution is necessary instead of a point-estimate. One method to obtain such distributions is the Bayesian method discussed in chapter three.

One method discussed, by Focardi, Kolm and Fabozzi is the so-called Monte Carlo method. In this method a set of returns is drawn iteratively from the expected return distribution. In each iteration, a mean-variance optimization is run on the set of expected returns. The robust portfolio is then the average of all the portfolios created in the different iterations.

Although this method will create portfolios that are more or less robust, it is computationally very expensive because an optimization must be run for each iteration step. Furthermore there is no guarantee that the resulting average portfolio will satisfy the constraints on which the original portfolios are created.

In the Robust Portfolio Allocation Approach, the portfolio is not created with an iterative process but the distribution of the expected returns is directly taken into account, resulting in a single optimization process. Therefore this approach is computationally more effective than the Monte Carlo process.

5.6 Conclusions

There are some different methods of portfolio optimization available. The classic Mean-Variance algorithms work with point-estimates of expected returns, variances and covariances. Chopra and Ziemba (1993) have done elementary research to the sensitivity of the classic Mean-Variance to errors in the input parameters. According to their research, errors in expected returns have a much bigger influence on the performance than errors in variances or covariances.

Portfolio optimization in the long short context does not differ much from optimization in the long only context. Jacobs, Levy and Starer (1999) show how short positions could be added to a long portfolio, by removing the 'greater than zero' constraint from the model. In order to optimize a true long-short portfolio, constraints should be added to the model in order to ensure equal (in terms of total exposure) long and short legs. This will be discussed in chapter 6, when the optimization model used in this research is developed.

The robust portfolio allocation approach, discussed by Focardi, Kolm and Fabozzi (2004), can be seen as an improvement to or an extension on the classic Mean-Variance method proposed by Markowitz (1952). In order to apply this approach, a Bayesian expected return model is needed. The same holds for the downside risk measures.

As already stated in the conclusion in chapter 3, in the research described in this thesis only expected return models, giving point estimates are investigated. Obviously, this has impact on the type of optimization models that could be applied. The robust portfolio allocation approach and the downside risk measure will not be implemented during this research. The models are only discussed in this section in order to give an idea about current opinions on the (problems with) optimization models.

In the forthcoming chapter, an optimization model will be developed which will be used within this research. This model will be based on the classic mean-variance model, but it will be adjusted in order to optimize a true long-short portfolio. As already discussed in the conclusions of chapter 4, the average bucket returns will be used as expected return model.

6 Applying Portfolio Management Theory

6.1 Portfolio Management in the Backtester

6.2 The optimization model – A step by step introduction

6.3 Adding transaction costs to the model

6.4 Optimization model including (linear) transaction costs

6.5 Covariance and beta calculations

6.6 Other objective functions

6.7 Ignoring risk

6.8 Conclusions

7 Implementing the optimization model

7.1 Reasons to use MATLAB and Visual Basic

7.2 The Optimization Application

7.3 Verification of the application

7.4 Recommendations for further development

8 Testing the optimization model

8.1 Constructing an initial portfolio

8.2 Testing method

8.3 Usage of Expected Returns

8.4 Results from the different test settings

8.5 Adjusting the model

8.6 Conclusions

9 Conclusions and Further Research

The conclusions of this research can be divided in two different parts, the conclusions about the construction of an expected return model and conclusions about the optimization model. The same holds more or less for the recommendations for further research.

9.1 Expected return model

9.1.1 Conclusions

Based on the findings in chapter 3, it could be expected that the development of an expected return model would be tough. Some theories state that it is impossible to find a working expected return model. In other articles some methods for predicting returns are described. But even these methods do not explain much of the variance observed in returns.

As we saw in chapter 4 of this thesis, it is difficult to derive an expected return model based on the rankings from the sector model. This task becomes a little bit easier for longer-term returns (which have a lower volatility). But the percentage of the variance that can be explained by such a model remains low.

Models based on the ranks are able to recognize a group of securities which will outperform another group of securities. Determining if a single security will outperform another security is more difficult. This is visible on two occasions. First the regression of bucket numbers on excess returns has a better performance than the regression of individual ranks. Secondly, in Backtest models with e.g. 20 instead of 5 buckets bucket 1 does not necessary outperform bucket 2.

Furthermore we saw that a regression on composite ranks or even variable ranks does not lead to results.

These two observations (buckets perform better than ranks and including composite ranks or variable ranks does not lead to better results) lead to the same conclusion. Incorporating more information in the model leads to the introduction of much more noisy data. Therefore the performance of such a model is lower.

This effect is confirmed by the results from the K-nearest neighbour technique. Because much randomly distributed data points are close to the new data point, the prediction of almost every new point is set to (almost) zero, the average of all the excess returns. Other techniques (decision trees, Markov chains) do not lead to better results. A summary of the most important results is given in **Error! Reference source not found.**.

9.1.2 Further research

One element, which was not really in the scope of this research, was making adjustments to the sector models. However, if a sector model could be found which has better performance or less noisy data, it could be easier to find an appropriate expected return model. One improvement that could be made is the development of models that vary over time. Unlike the current models, these models have the probability to adapt their definition to changing (market) circumstances. One can think about methods that update the weights of the different variables or composites in such a way that the explanatory power of the models is maximal. Such a model, with dynamic weights, is described by Hartman, Wesselius, Aldred and Steel (2003).

In this research, only expected return models that result in a point-estimate are investigated. Models based on Bayesian statistics result in an expected return distribution instead of a point-estimate. Therefore these models are not tested in this research. However research to these models is certainly recommended as a future development.

9.2 Optimization model

9.2.1 Conclusions

As already stated in the introduction, the main objective of the optimization models developed during this research is to reduce trading costs. Therefore the optimization models will make a trade-off between the extra return that can be generated by a trade and the costs related with this trade. It is obvious that the results of those optimizations are closely related to the performance of the expected return model used within the optimization.

First one remark should be made about the performance of trading strategies in general. In some periods none of the trading strategies is profitable. This is especially observed in the high volatile year 2001. In this year, only the naïve strategy was profitable. In the last 2 years (2003, 2004) each trading strategy is very profitable. Furthermore, in this period the optimization models really outperform the trading rule strategy currently used within SPB and implemented in the Backtester.

The different forms of the optimization models do all lead to comparable results, except from the model in which the expected information ratio is maximized. This model leads to very risk-averse portfolios. The choice among the other models depends really on the personal behaviour of the investor.

If the optimization models are compared with the trading rules, the following two observations can be made.

- All trading strategies lead to lower volatilities. Consequently, the results of the optimization models are more stable.
- The usage of an optimization model does not lead to a reduction of trading costs.

Obviously if we compare the trading costs of the optimization models with the trading costs of the naïve strategy, there is an enormous save on trading costs.

If we try to save more on trading costs, by constraining the amount of trading costs in the optimization model, we see that we have a loss in realized return before trading costs, which cannot be compensated by the savings in trading costs. Thus, limiting trading costs explicitly is not profitable.

In section 8.6, an overview is given of the advantages and disadvantages of the various types of models. In order to give complete conclusions in this chapter, this table is repeated here.

Naïve Strategy		Trading Rules		Optimization model	
+	Higher returns in high volatile markets	+	Savings on transaction costs	+	Savings on transaction costs
		+	Somewhat higher returns in high volatile markets	+	Higher returns in periods with low market volatility
-	High turnover between periods and thus a high loss in transaction costs			+	Properties like beta can easily be controlled by additional constraints
-	High volatility of realized returns	-	High volatility of realized returns	+	Low volatility of realized returns
-	Difficult to control e.g. beta tilt	-	Difficult to control e.g. beta tilt		
-	Low performance in low-volatility periods	-	Low performance in low-volatility periods	-	Lower returns in high volatile markets

Table 9-A: Error! Reference source not found. **(copy of** Error! Reference source not found.**)**

9.2.2 Further research

The direction in which the research to optimization models can be extended depends on the direction in which the research to expected return models will be extended.

Because only point-estimates of expected returns are created, the possibilities for optimization models are limited. However if more sophisticated models or approaches are applied, other optimization models can be developed. Examples are optimization models using downside risk measures or models using the robust portfolio allocation approach.

However, even without the usage of expected return distributions, several developments are imaginable. One can think about other utility functions and constraint sets. For example a set in which beta is no longer constrained but part of the utility function. Other constraints can be treated in the same way. In order to move elements to the utility function a weight should be attached to the element. Therefore a method to determine this weight must be developed.

Another extension that could be made to the model is the addition of new constraints, e.g. a constraint that ensures Market Cap neutrality. Another option is the addition of operational constraints that could vary over time.

The model could also be extended in the direction of transaction costs estimates. In the current models linear transaction cost estimates are used. However the original

PRISE model is not linear. This can be read in Appendix D. Furthermore, currently execution costs are treated as a single value. The volatility of those costs is not taken into account. From Rust and Ramaswami (1999) we know that those costs are volatile.

A completely different direction, which could be investigated in the future, is a research to other techniques to create (optimal) portfolios. All models treated in this research are (linear or quadratic) optimization models. However, as already reported by Focardi, Kolm and Fabozzi (2004), genetic algorithms could also be applied to portfolio construction tasks.

9.2.3 Integrating the models within SPB applications

Another research question was to find a method to integrate those models within SPB. Most recommendations are already discussed in section 7.4, but before the optimization models can be applied in practice some performance issues must be solved. Currently the optimizations need a long time in order to run to completion, which could speed up more or less by another model formulation. Research to this has to be done before those models can be used in practical applications like the Backtester.

9.3 Final Conclusions

If we look at the longer-term (2000-2004) results as discussed in chapter 8, the trading rule model leads to a higher average return than the optimization models. However, the risk involved with the trading rule model is much higher. Any investor that does totally not care about risk may want to use the model based on trading rules. However, investors which do care about risk can obtain a stable outperformance with the optimization models developed in this thesis. By using an optimization model, this extra stability can be obtained without compromising the realized returns too much. Therefore these investors will be better of with the optimization models. SPB certainly belongs to the category of investors who care about risk.

Furthermore the optimization model enables the possibility to remove unwanted properties from the portfolios, like beta tilts or small cap biases. Therefore portfolios obtained by an optimization model are much more controllable.

Appendix A Trading Rules

A.1 Trading rules model

Appendix B Dataset Banks

Appendix C Comprehensive Examples

C.1 Two stage model

This example shows how the two stage model, described in this thesis works, therefore consider the following multi-period dataset

Security	Period	Rank	Return
A	1	0	5%
B	1	0.33	1%
C	1	0.66	-4%
D	1	1	-2%
A	2	0	10%
B	2	0.66	-12%
C	2	0.33	5%
D	2	1	-3%
A	3	0.33	-2%
B	3	0	0%
C	3	1	-1%
D	3	0.66	3%
A	4	0	-2%
B	4	0.33	6%
C	4	0.66	1%
D	4	1	-5%

Now, we can calculate a regression model for each period. This will give us the following models

Period	Model	Volatility of returns	Lagged Volatility
1	return = -0.078*rank + 0.039	3.9%	6.3% *
2	return = -0.167*rank + 0.083	9.6%	3.9%
3	return = -0.005*rank + 0.003	2.2%	9.6%
4	return = -0.043*rank + 0.021	4.7%	2.2%

* volatility based on non-displayed returns of period 0.

Goal of the two-stage model is to predict the constants alpha and beta of the regression models (expressed as return = rank*beta + alpha)with a second regression. There could be different choices for the descriptive variables used in this regression. In this example, volatility is used. Obviously, the realized volatility of the period is not available when the prediction model is determined. Therefore we will regress alpha and beta on the one period lagged volatility. This will give us the following models:

alpha = 1.077 * [lagged volatility] – 0.02
beta = -2.17 * [lagged volatility] + 0.04

The expected return of a new security will be determined as follows. First the alpha and beta for the corresponding prediction model will be estimated (first stage). Second, with this regression model the return for the new security will be predicted (second stage)

C.2 Markov Chain

In this thesis, the Markov Chain proposed by McQueen and Chorley (1991) is adapted to a model, which is able to predict returns based on prior ranks and returns. This example demonstrates how returns can be predicted with this model. Therefore, consider the following dataset.

Security	Period	rank	rank state	return	Return state	Previous Return	Prev. Return State
A	1	0	1	2%	1	-3%	0
B	1	0.25	2	3%	1	-8%	0
C	1	0.5	3	4%	1	10%	1
D	1	0.75	4	-5%	0	5%	1
E	1	1	5	-4%	0	-4%	0
A	2	0.5	3	0%	0	2%	1
B	2	0.25	2	4%	1	3%	1
C	2	1	5	-2%	0	4%	1
D	2	0	1	1%	1	-5%	0
E	2	0.75	4	-3%	0	-4%	0
A	3	0.5	3	2%	1	0%	0
B	3	0.25	2	-6%	0	4%	1
C	3	0	1	-2%	0	-2%	0
D	3	1	5	-4%	0	1%	1
E	3	0.75	4	10%	1	-3%	0

The average returns for the two return states are calculated by all realized returns with the corresponding state: This give the following averages

State	Average realized return
0	-3%
1	4%

The original states consist of the rank state and the previous return state. A state is denoted as rank-return (for example, 5-1 indicates rank state 5, return state 1). Now the probabilities of the transitions to the new return states have to be calculated.

First the occurrences of the original states will be counted:

State	Count
1-0	3
1-1	0
2-0	1
2-1	2
3-0	1
3-1	2
4-0	2
4-1	1
5-0	1
5-1	2

Then the transitions have to be counted. The transition probabilities will be calculated by dividing all transition counts by the number of observations of the original state. For example, if we observe state A-B five times and we observe three transitions from state 4-0 to state 1, the probability for a transition from 4-0 to 1 is 3/5 = 0.6

The following table shows both the observations and probabilities of the transitions

Observations			**Probabilities**		
State	0	1	State	0	1
1-0	1	2	1-0	0.33	0.67
1-1	0	0	1-1	0	0
2-0	0	1	2-0	0	1
2-1	1	1	2-1	0.5	0.5
3-0	0	1	3-0	0	1
3-1	1	1	3-1	1	1
4-0	1	1	4-0	0.5	0.5
4-1	1	0	4-1	1	0
5-0	1	0	5-0	1	0
5-1	2	0	5-1	1	0

Now assume the return of a new security with a return in the previous period of –3% and a rank of 0.1 has to be predicted. The state of this security is 1-0. The table with transition probabilities displays a probability of 0.33 of return state 0 (expected return –3%) and a probability of 0.67 of return state 1 (expected return 4%). The expected return of the security becomes 0.33 * -3% + 0.67 * 4% = 1.7%

Appendix D Calculating Transaction costs

D.1 Calculating costs – Examples

D.2 Different types of costs

D.3 Estimating Price Impacts

D.4 Linear prediction

Appendix E Screenshots Optimization Application

E.1 Screen ‘Load data from Backtest results’

E.2 Screen ‘Load additional data’

E.3 Screen ‘ Specify optimization parameters’

E.4 Screen ‘Optimize’

Appendix F SQP algorithm

The optimization model implemented in this thesis will be executed with the SQP algorithm, implemented in the 'fmincon' function of the Optimization Toolbox of MATLAB. The following text describes this algorithm. It is an integral copy of the text available in the MATLAB documentation, and is also available via the url: **http://www.mathworks.com/access/helpdesk/help/toolbox/optim/tutor18b.html**

F.1 SQP Implementation

The SQP implementation consists of three main stages, which are discussed briefly in the following subsections:

- Updating of the Hessian matrix of the Lagrangian function
- Quadratic programming problem solution
- Line search and merit function calculation

F.1.1 Updating the Hessian Matrix

At each major iteration a positive definite quasi-Newton approximation of the Hessian of the Lagrangian function, *H*, is calculated using the BFGS method, where $\lambda_i\ (i = 1, \ldots, m)$ is an estimate of the Lagrange multipliers.

$$H_{k+1} = H_k + \frac{q_k q_k^T}{q_k^T s_k} - \frac{H_k^T H_k}{s_k^T H_k s_k} \qquad \text{(3-27)}$$

where

$$s_k = x_{k+1} - x_k$$

$$q_k = \nabla f(x_{k+1}) + \sum_{i=1}^{n} \lambda_i \cdot \nabla g_i(x_{k+1}) - \left(\nabla f(x_k) + \sum_{i=1}^{n} \lambda_i \cdot \nabla g_i(x_k)\right)$$

Powell [35] recommends keeping the Hessian positive definite even though it might be positive indefinite at the solution point. A positive definite Hessian is maintained providing $q_k^T s_k$ is positive at each update and that *H* is initialized with a positive definite matrix. When $q_k^T s_k$ is not positive, q_k is modified on an element-by-element basis so that 0 -->$q_k^T s_k > 0$. The general aim of this modification is to distort the elements of q_k , which contribute to a positive definite update, as little as possible. Therefore, in the initial phase of the modification, the most negative element of $q_k .* s_k$ is repeatedly halved. This procedure is continued until $q_k^T s_k$ is greater than or equal to 1*e*-5. If, after this procedure, $q_k^T s_k$ is still not positive, modify q_k by adding a vector *v* multiplied by a constant scalar *w*, that is,

$$q_k = q_k + wv \qquad \text{(3-28)}$$

Where

$$v_i = \nabla g_i(x_{k+1}) \cdot g_i(x_{k+1}) - \nabla g_i(x_k) \cdot g_i(x_k),$$

if $(q_k)_i \cdot w < 0$ and

$(q_k)_i \cdot (s_k)_i < 0 (i = 1, \ldots m)$

$v_i = 0$ otherwise

and increase w systematically until $q_k^T s_k$ becomes positive.

The functions `fmincon`, `fminimax`, `fgoalattain`, and `fseminf` all use SQP. If `Display` is set to `'iter'` in `options`, then various information is given such as function values and the maximum constraint violation. When the Hessian has to be modified using the first phase of the preceding procedure to keep it positive definite, then `Hessian modified` is displayed. If the Hessian has to be modified again using the second phase of the approach described above, then `Hessian modified twice` is displayed. When the QP subproblem is infeasible, then `infeasible` is displayed. Such displays are usually not a cause for concern but indicate that the problem is highly nonlinear and that convergence might take longer than usual. Sometimes the message `no update` is displayed, indicating that $q_k^T s_k$ is nearly zero. This can be an indication that the problem setup is wrong or you are trying to minimize a noncontinuous function.

F.1.2 Quadratic Programming Solution

At each major iteration of the SQP method, a QP problem of the following form is solved, where A_i refers to the `i`th row of the *m*-by-*n* matrix A .

$$\underset{d \in \Re^n}{\text{minimize}} \quad q(d) = \frac{1}{2}d^T H d + c^T d \tag{3-29}$$

$$A_i d = b_i \qquad i = 1, \ldots, m_e$$

$$A_i d \le b_i \qquad i = m_e + 1, \ldots, m$$

The method used in the Optimization Toolbox is an active set strategy (also known as a projection method) similar to that of Gill et al., described in [20] and [19]. It has been modified for both Linear Programming (LP) and Quadratic Programming (QP) problems.

The solution procedure involves two phases. The first phase involves the calculation of a feasible point (if one exists). The second phase involves the generation of an iterative sequence of feasible points that converge to the solution. In this method an active set, $\bar{A}_k$, is maintained that is an estimate of the active constraints (i.e., those that are on the constraint boundaries) at the solution point. Virtually all QP algorithms are active set methods. This point is emphasized because there exist many different methods that are very similar in structure but that are described in widely different terms.

$\overline{A}_k$ is updated at each iteration *k*, and this is used to form a basis for a search direction $\hat{d}_k$. Equality constraints always remain in the active set $\overline{A}_k$. The notation for the variable $\hat{d}_k$ is used here to distinguish it from d_k in the major iterations of the SQP method. The search direction $\hat{d}_k$ is calculated and minimizes the objective function while remaining on any active constraint boundaries. The feasible subspace for $\hat{d}_k$ is formed from a basis Z_k whose columns are orthogonal to the estimate of the active set $\overline{A}_k$ (i.e., $\overline{A}_k Z_k = 0$). Thus a search direction, which is formed from a linear summation of any combination of the columns of Z_k, is guaranteed to remain on the boundaries of the active constraints.

The matrix Z_k is formed from the last $m - l$ columns of the QR decomposition of the matrix $\overline{A}_k^T$, where *l* is the number of active constraints and $l < m$. That is, Z_k is given by

$$Z_k = Q[:, l+1:m] \quad \textbf{(3-30)}$$

where

$$Q^T \overline{A}_k^T = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

Once Z_k is found, a new search direction $\hat{d}_k$ is sought that minimizes $q(d)$ where $\hat{d}_k$ is in the null space of the active constraints. That is, $\hat{d}_k$ is a linear combination of the columns of Z_k: $\hat{d}_k = Z_k p$ for some vector *p*.

Then if you view the quadratic as a function of *p*, by substituting for $\hat{d}_k$, you have

$$q(p) = \frac{1}{2} p^T Z_k^T H Z_k p + c^T Z_k p \quad \textbf{(3-31)}$$

Differentiating this with respect to p yields

$$\nabla q(p) = Z_k^T H Z_k p + Z_k^T c \quad \textbf{(3-32)}$$

$\nabla q(p)$ is referred to as the projected gradient of the quadratic function because it is the gradient projected in the subspace defined by Z_k. The term $Z_k^T H Z_k$ is called the projected Hessian. Assuming the Hessian matrix *H* is positive definite (which is the case in this implementation of SQP), then the minimum of the function *q(p)* in the subspace defined by Z_k occurs when $\nabla q(p) = 0$, which is the solution of the system of linear equations

$$Z_k^T H Z_k p = -Z_k^T c \qquad \textbf{(3-33)}$$

A step is then taken of the form

$$x_{k+1} = x_k + \alpha \hat{d}_k \qquad \text{where } \hat{d}_k = Z_k^T p \qquad \textbf{(3-34)}$$

At each iteration, because of the quadratic nature of the objective function, there are only two choices of step length α. A step of unity along $\hat{d}_k$ is the exact step to the minimum of the function restricted to the null space of $\overline{A}_k$. If such a step can be taken, without violation of the constraints, then this is the solution to QP (Eq. 3-30). Otherwise, the step along $\hat{d}_k$ to the nearest constraint is less than unity and a new constraint is included in the active set at the next iteration. The distance to the constraint boundaries in any direction $\hat{d}_k$ is given by

$$\alpha = \min_i \left\{ \frac{-(A_i x_k - b_i)}{A_i \hat{d}_k} \right\} \qquad (i = 1, \ldots, m) \qquad \textbf{(3-35)}$$

which is defined for constraints not in the active set, and where the direction $\hat{d}_k$ is towards the constraint boundary, i.e., $A_i \hat{d}_k > 0,\ i = 1, \ldots, m$.
When n independent constraints are included in the active set, without location of the minimum, Lagrange multipliers, λ_k, are calculated that satisfy the nonsingular set of linear equations

$$\overline{A}_k^T \lambda_k = c \qquad \textbf{(3-36)}$$

If all elements of λ_k are positive, x_k is the optimal solution of QP (Eq. 3-30). However, if any component of λ_k is negative, and the component does not correspond to an equality constraint, then the corresponding element is deleted from the active set and a new iterate is sought.

Initialization. The algorithm requires a feasible point to start. If the current point from the SQP method is not feasible, then you can find a point by solving the linear programming problem

$$\underset{\gamma \in \Re, x \in \Re^n}{\text{minimize}} \quad \gamma \tag{3-37}$$

$$A_i x = b_i \qquad i = 1, \ldots, m_e$$

$$A_i x - \gamma \le b_i \qquad i = m_e + 1, \ldots, m$$

The notation A_i indicates the *i*th row of the matrix *A*. You can find a feasible point (if one exists) to Eq. 3-37 by setting *x* to a value that satisfies the equality constraints. You can determine this value by solving an under- or overdetermined set of linear equations formed from the set of equality constraints. If there is a solution to this problem, then the slack variable γ is set to the maximum inequality constraint at this point.

You can modify the preceding QP algorithm for LP problems by setting the search direction to the steepest descent direction at each iteration, where g_k is the gradient of the objective function (equal to the coefficients of the linear objective function).

$$\hat{d}_k = -Z_k Z_k^T g_k \tag{3-38}$$

If a feasible point is found using the preceding LP method, the main QP phase is entered. The search direction $\hat{d}_k$ is initialized with a search direction $\hat{d}_1$ found from solving the set of linear equations

$$H\hat{d}_1 = -g_k \tag{3-39}$$

where g_k is the gradient of the objective function at the current iterate x_k (i.e., $Hx_k + c$).
If a feasible solution is not found for the QP problem, the direction of search for the main SQP routine $\hat{d}_k$ is taken as one that minimizes γ .

F.1.3 Line Search and Merit Function

The solution to the QP subproblem produces a vector d_k , which is used to form a new iterate

$$x_{k+1} = x_k + \alpha d_k \tag{3-40}$$

The step length parameter α_k is determined in order to produce a sufficient decrease in a merit function. The merit function used by Han [24] and Powell [35] of the following form is used in this implementation.

$$\Psi(x) = f(x) + \sum_{i=1}^{m_e} r_i \cdot g_i(x) + \sum_{i=m_e+1}^{m} r_i \cdot \max\{0, g_i(x)\} \quad \text{(3-41)}$$

Powell recommends setting the penalty parameter

$$r_i = (r_{k+1})_i = \max_i\left\{\lambda_i, \frac{1}{2}((r_k)_i + \lambda_i)\right\}, \qquad i = 1, \ldots, m \quad \text{(3-42)}$$

This allows positive contribution from constraints that are inactive in the QP solution but were recently active. In this implementation, the penalty parameter r_i is initially set to

$$r_i = \frac{\| \nabla f(x) \|}{\| \nabla g_i(x) \|} \quad \text{(3-43)}$$

where $\| \ \|$ represents the Euclidean norm.
This ensures larger contributions to the penalty parameter from constraints with smaller gradients, which would be the case for active constraints at the solution point.

F.1.4 The references in this text are the following:

[19] Gill, P.E., W. Murray, M.A. Saunders, and M.H. Wright, "Procedures for Optimization Problems with a Mixture of Bounds and General Linear Constraints," *ACM Trans. Math. Software*, Vol. 10, pp 282-298, 1984.

[20] Gill, P.E., W. Murray, and M.H. Wright, *Numerical Linear Algebra and Optimization*, Vol. 1, Addison Wesley, 1991.

[24] Han, S.P., "A Globally Convergent Method for Nonlinear Programming," *J. Optimization Theory and Applications*, Vol. 22, p. 297, 1977.

[35] Powell, M.J.D., "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations," *Numerical Analysis*, G.A.Watson ed., Lecture Notes in Mathematics, Springer Verlag, Vol. 630, 1978.

Literature

Barthold, J. and Peare, P. (2003) Unbiased Estimation of Expected Return using CAPM, *International Review of Financial Analysis 12 (2003) pages 69-81*

Brav, A., Lehavy, R. and Michaely, R. (2003) Expected Return and Asset Pricing, *draft version*

Block, S.B. and Hirt, G.A. (2000) Foundations of Financial Management, international edition, *McGraw-Hill Higher Education*

Chopra, V.K. and Ziemba, W.T. (1993) The Effect of Errors in Means, Variances, and Covariances on Optimal Portfolio Choice, *Journal of Portfolio Management Winter 1993 pages 6-11*

Fama, E.F. (1970), Efficient Capital Markets: A Review of Theory and Empirical Work, *The Journal of Finance, pages 383-417*

Focardi, S.M., Kolm, P.N. and Fabozzi, F.J. (2004) New Kids on the Block, *Journal of Portfolio management, 30th Anniversary Issue 2004 pages 42-54*

Gunst, M.C.M. de and Vaart, A.W. van der (2001) Reader Statistische Data Analyse, *Vrije Universiteit, Amsterdam*

Hartmann, S., Wesselius, P., Aldred, N. and Steel, D. (2003) Seperating the winners from the losers, *ABN-AMRO*

He, G., and Litterman, D. (1999) The Intuition Behind the Black-Litterman Model Portfolios, *Goldman-Sachs, investment management division*

Hillier, F.S. and Lieberman, G.J. (1980) Introduction to Operations Research (third edition), *Holden-Day Inc*

Jacobs, B.I., Levy, K.N. and Starer, D. (1999) Long-Short Portfolio Management: An Integrated Approach, *Journal of portfolio management, Winter 1999 pages 21-32*

Markowitz, H. (1952) Portfolio Selection, *The Journal of Finance, Vol 7. No.1 (March 1952), pages 77-91*

McQueen, G. and Thorley, S. (1991) Are Stock Returns Predictable? A Test Using Markov Chains, *The Journal of Finance, Vol. 46 No.1 pages 239-263.*

Mitchell, J.E. and Braun, S. (2003) Rebalancing an Investment Portfolio in the Presence of Transaction Costs, *draft version*

Mitchell, T.M. (1997) Machine Learning, *McGraw-Hill*

Rust, C. and Ramaswami, M. (1999) Introducing PRISE - The Lehman Brothers Pre-Trade Implementation Shortfall Estimation Model, *Lehman Brothers*

Sharpe, W.F. (1994), The Sharpe Ratio, *Journal of Portfolio management, Fall 1994*

Witten, I.H. and Frank, E. (2000) Data Mining, *Morgan Kaufmann Publishers*

Womersley, R.S. and Lau, K. (1996) Portfolio Optimisation Problems, *Computational Techniques and Application, World Scientific Publishing 1996*