MODELLING AND SOLUTION METHODS FOR PORTFOLIO OPTIMISATION

A THESIS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By

Marion Guertler

Brunel University
Department of Mathematical Sciences
Uxbridge, UB8 3PH, Middlesex
September 2003

Abstract

In this thesis modelling and solution methods for portfolio optimisation are presented. The investigations reported in this thesis extend the Markowitz mean-variance model to the domain of quadratic mixed integer programming (QMIP) models which are 'NP-hard' discrete optimisation problems. In addition to the modelling extensions a number of challenging aspects of solution algorithms are considered. The relative performances of sparse simplex (SSX) as well as the interior point method (IPM) are studied in detail. In particular, the roles of 'warmstart' and dual simplex are highlighted as applied to the construction of the efficient frontier which requires processing a family of problems; that is, the portfolio planning model stated in a parametric form. The method of solving QMIP models using the branch and bound algorithm is first developed; this is followed up by heuristics which improve the performance of the (discrete) solution algorithm. Some properties of the efficient frontier with discrete constraints are considered and a method of computing the discrete efficient frontier (DEF) efficiently is proposed. The computational investigation considers the efficiency and effectiveness in respect of the scale up properties of the proposed algorithm. The extensions of the real world models and the proposed solution algorithms make contribution as new knowledge.

Acknowledgement

I would like to thank my supervisor, Prof. Gautam Mitra, for the numerous stimulating discussions of my work and his continuous scientific and financial support throughout the course of this Ph.D.

I also would like to express my appreciation to UBS Warburg for providing insights into the real world and their financial support.

I wish to thank the members of the Department of Mathematical Sciences, especially Dr. Cormac Lucas and Frank Ellison for their support and collaboration.

Mein größter Dank jedoch gilt meiner ganzen Familie und meinen Freunden, die mich auf meiner 'PhD-Reise' unterstützt und ermutigt haben.

Abbreviations

MP	:	Mathematical programming
MOP	:	Multi objective programming
MOLP	:	Multi objective linear programming
NLP	:	Nonlinear programming
LP	:	Linear programming
PLP	:	Primal linear programming
DLP	:	Dual linear programming
QP	:	Quadratic programming (convex case)
PQP	:	Primal quadratic programming
DQP	:	Dual quadratic programming
IP	:	Integer programming
PIP	:	Pure integer programming
MIP	:	Mixed integer programming
QMIP	:	Quadratic mixed integer programming
KKT	:	Karush-Kuhn-Tucker conditions
IPM	:	Interior point method
SSX	:	Sparse simplex method

BI	:	Basis identification
XO	:	Cross-over
B&B	:	Branch and bound
s.t.	:	subject to
min	:	minimise
max	:	maximise
MV	:	Mean-Variance
EF	:	Efficient frontier
CEF	:	Continuous efficient frontier
DEF	:	Discrete efficient frontier

Contents

List of Tables

List of Figures

Chapter 1

Introduction

"The great decisions of human life have as a rule far more to do with the instincts and other mysterious unconscious factors than with conscious will and well-meaning reasonableness. The shoe that fits one person pinches another; there is no recipe for living that suits all cases. Each of us carries his own life–form–an indeterminable form which cannot be superseded by any other."
Carl Gustav Jung, Modern Man in Search of a Soul, 1933, p. 69

This thesis is concerned with the modelling and solution of the portfolio optimisation problem. The original continuous portfolio optimisation problem can be viewed as a quadratic programming problem. When real world extensions are considered, the problem becomes a quadratic mixed integer programming problem.

A general introduction to optimisation problems is set out in section 1.1 and a summary of the historical evolution of the solution algorithms is made. In section 1.2 examples and applications of the quadratic or quadratic mixed integer programming problem are described providing the motivation of this study. Since the portfolio optimisation problem is a special case of multi-objective optimisation, parallels are drawn in section 1.3. Finally, section 1.4 provides the outline of this thesis.

1.1 Optimisation models

An optimisation problem that either minimises or maximises some function over a set of real or integer variables is known as an *unconstrained programming* problem. When the set of variables is restricted in respect to their value then the resulting problem is called a *constrained optimisation programming* problem. Both problems are called *mathematical programming* (MP) problems. The most general formulation is set out as a nonlinear programming (NLP) problem and is defined in the following way.

NLP:

$$\begin{aligned} \text{min (or max)} \quad & f(x) & \\ \text{s.t.} \quad & g_i(x) = & b_i, i = i, ..., l \\ & h_i(x) \geq & b_i, i = l + 1, ..., m \\ & x \in \mathbb{R}^n & , \end{aligned} \tag{1.1}$$

In this statement $f(x)$ is called the objective function and $g_i(x)$ and $h_i(x)$ are other functions defining equality and inequality constraints respectively. In general the functions f, g_i, h_i may be nonlinear.

If the objective function f and the constraint functions g_i, h_i are linear, the corresponding problem is known as a linear programming (LP) problem. In the case that f is a quadratic function and g_i, h_i are linear restrictions, then this is known as a quadratic programming (QP) problem. An NLP is a convex programming problem if (a) the objective function (minimisation) is convex and (b) the constraints define a convex set. In the case of QP, if the objective functions is convex then this becomes a convex programming problem since the linear constraints define a convex set. If for the LP and the QP problem some of the variables are restricted to take integer values then, the linear programming problem is called a mixed integer programming (MIP) problem and the quadratic programming problem becomes a quadratic mixed integer programming (QMIP) problem.

In the development of computational algorithms for MP problems, solving linear programming problems plays a central role. From the computational point of view the most significant development was the simplex method proposed and actively developed by Dantzig in 1947. Since then the methods for solving LP's have been widely studied and are described in Dantzig [21], Beale [4], Mitra [77], Bixby et al. [12], Vanderbei [89] and others. In 1984, Karmarkar [47] inspired the research world with a new polynomial time algorithm for LP referred to as the interior point method. Further investigation led to Mehrotra's predictor corrector method [73]. Amongst others, Andersen [1], Gondzio [33], Ross et al. [84], Wright [95] reconnoitred the interior approach further. The achievements of the investigations for LP solution algorithms has enabled problems with millions of constraints and variables to be solved.

In 1952, Markowitz introduced QP but up until the 90's implementations could only cope with portfolios over a small asset universe. However, incorporating the advances from LP during the 90's, the finance industry has shown a considerable interest in applying quadratic programming methods for solving portfolio problems. This thesis considers the solution of quadratic programming problems. Wolfe [92, 93], Van-de-Panne and Whinston [87, 88], Cottle and Dantzig [18], Beale [4] kindled the interest and made the initial contributions of the solution methods for QP problems. With the work of Carpenter et al. [14], Vanderbei [89], Jones [44], Meszaros [74], Guertler et al. [34] and others, the solution of large scale QP problems can now be carried out on workstations.

Since the early nineties, portfolio optimisation models of Markowitz have gained wide acceptance in industry. This in turn has led to the introduction of sophisticated extensions which require the use of discrete (zero-one and integer) variables to represent these problems. The use of discrete variables and discrete constraints extends the continuous Markowitz problem into a mixed integer programming problem. Heuristic techniques such as simulated annealing and MP based methods typically branch & bound, branch & cut have been adapted to process this class of QMIP problems. Bienstock [11], Beasley [7], Chang [16], Guertler et al. [35] describe approaches for the solution of QMIP problems.

1.2 Instances of quadratic and quadratic mixed integer programming problems

Quadratic and quadratic mixed integer programming problems are special cases of NLP's and arise in many applications when the objective function is quadratic and the decision variables are subject to linear restrictions.

In 1959, Wolfe [92] identified the following four typical quadratic programming problems:

1. *Minimum Variance*: For a given range of expected returns, the variance of these returns are minimised [64].

2. *Regression Analysis*: The least square fit to given data is to be found, where certain parameters are known a priori to satisfy linear inequality constraints [30].

3. *Efficient production*: The profit is maximised with linear production functions and linearly varying marginal costs [22].

4. *Convex programming*: A convex function is minimised subject to linear constraints and quadratic approximations [91].

Since then, the different areas have been enhanced and quadratic programming techniques have become more important. In this thesis the minimum variance problem applied in finance plays a central role.

The minimum variance model

This problem, introduced by Markowitz [63] is a classic case of a QP problem and is referred to as portfolio optimisation. This is the process of analysing a portfolio and managing the assets within it to obtain the highest return for a given level of risk or the lowest risk for a certain level of return. The basic portfolio optimisation theory hinges on the mean-variance (MV) or Markowitz paradigm explained in [64, 65, 66, 67].

In the model proposed by Markowitz, there are N risky assets each with a mean return μ_i. The covariance between security i and j is σ_{ij} and σ_{ii} is the variance of security i.The portfolio weights are x_i and the Markowitz model is stated as

$$\min \quad \sum_{i=1}^{N}\sum_{j=1}^{N} \sigma_{ij} x_i x_j \tag{1.2}$$

$$s.t. \quad \sum_{i=1}^{N} x_i = 1, \tag{1.3}$$

$$\sum_{i=1}^{N} \mu_i x_i = \rho, \tag{1.4}$$

$$x_i \geq 0, \forall i \tag{1.5}$$

where ρ is the desired level of expected return for the portfolio.

The objective function 1.2 minimises the portfolio variance. Markowitz's major contribution was to postulate that this measures 'risk' which should be minimised. The constraint 1.3 ensures that the portfolio is fully invested and restriction 1.4 guarantees the level of expected return desired by the investor. It is common practice to exclude short sales i.e. the investor is not allowed to borrow cash to build the portfolio. The non-negativity constraint 1.5 ensures that there are no short sales.

Furthermore, in this approach it is assumed that none of the assets are perfectly correlated and also that there are no risk free securities. This ensures a non-singular positive definite covariance matrix. As the variance of a risky portfolio is strictly positive, this covariance matrix is also positive definite and hence, the objective function is convex. As for the linear constraints, they define a convex set. Thus, the problem has a unique optimal solution presuming that the returns μ_i are not equal.

Extensions to the Markowitz model

The basic Markowitz model does not reflect the requirements of the real world. As a result the findings by the continuous MV model are exploited and expanded towards the needs of industry either by adding 'discrete' variables or diversifying the data. The main supplementary discrete constraints added to the model are:

- Buy-in threshold constraints: These constraints define the minimum level at which an asset can be traded. It eliminates the problem where unrealistically small trades can be included in an optimum portfolio.

- Cardinality constraint: To find a manageable portfolio out of a large universe of stocks with low transaction and taxation costs, it is sensible to limit the number of unique stocks in the portfolio. The cardinality is implicitly linked to the Buy-in threshold.

Further restrictions which constrain the basic unit of investment (or roundlot) or try to match the duration can be introduced. The theory and application of various restrictions is investigated by Chang et al. [16], Jobst et al. [43].

Accommodating realistic properties of the portfolio selection process into the model increases the size of the QP/QMIP problem extensively and thus its complexity. In this thesis, approaches are suggested and computational studies are performed to overcome the obstacle arising from a realistic portfolio optimisation problem.

Not only has the mean variance problem been rediscovered but also in the regression analysis new research directions are pursued since quadratic programming and discrete quadratic programming problems can now be solved.

The regression model

The problem of regression analysis is to minimise the sum of squares of the residuals. This approach is very sensitive towards outliers infringing against the normality assumption. Recently, Camarinopoulos and Zioutas [13] revisted the idea of the regression model and compare established methods of robust regression estimation with the modified Mallows-type approach [61] which restricts the downweight resource. With quadratic programming techniques, the bound influenced estimates are minimised.

Given n points $(y_i, x_{1i}, ..., x_{pi})$ in a $p+1$ Euclidean space, the problem is to solve the weighted least squares curve-fitting problem, finding the estimators $\beta = \beta_1 - \beta_2$ with $\beta_1 = (\beta_{11}, \beta_{12}, ..., \beta_{1p})^T$ and $\beta_2 = (\beta_{21}, \beta_{22}, ..., \beta_{2p})^T$ which solve the following problem:

$$\begin{array}{lll}
\min & \frac{1}{2}\sum\limits_{i=1}^{n}(u_i^{*2} + 2cw_i\sigma\epsilon_i) & \\
\text{s.t.} & x_i^T\beta_1 - x_i^T\beta_2 + u_i^* + \epsilon_i & \geq y_i, \\
& x_i^T\beta_1 - x_i^T\beta_2 - u_i^* - \epsilon_i & \leq y_i, \\
& \sum\limits_{i=1}^{n} \epsilon_i & \leq E, \\
& \beta_1, \beta_2, u_i^*, \epsilon_i \geq 0, & i = 1, ..., n
\end{array}$$

where $2cw_i\sigma\epsilon_i$ is the weighted penalty cost associated with the weights w_i, c a tuning parameter and σ the standard deviation. u_i^* are the "metrically Winsorised" [15]residuals obtained by pulling y_i towards the fitted value by a distance ϵ_i. The constraint $\sum_{i=1}^{n} \epsilon_i \leq E$ limits the pulling resource E. It is shown by Camarinopoulos and Zioutas [13] that compared with other methods it is beneficial to use quadratic programming techniques to obtain these estimators. Furthermore, the use of mathematical programming allows to perform post-optimality analysis in respect of the standard deviation σ.

1.3 Multi-objective models

Many real-world decision problems take into account more than one objective. Typical (multiple) objectives are to minimise risk, maximise reliability, minimise deviations from a desired level, minimise cost. A multi objective programming (MOP) problem with conflicting objectives results in different optimal solutions or a compromised solution. A multiple objective LP is of the form:

MOLP:

$$\begin{aligned} \min \quad & z = \{c^1x, c^2x, ..., c^rx\} = z_1, ..., z_r \\ s.t. \quad & Ax = b \\ & x \geq 0, \end{aligned}$$

where $c^1, c^2, ..., c^r$ are a set of r objective vectors. In most cases it would not be possible to find one or more solutions which simultaneously optimise all of the objectives. Multi-objective problems are usually investigated by optimising each objective function individually to obtain bounds and then building a single objective optimisation problem. This combines all the different objectives into one *or* entails setting all except one of them as constraints in the MP problem.

Pareto optimality

The optimal solution of a multi objective programming problem is said to be (globally) Pareto optimal or (globally) efficient or non-dominated or non-inferior if there exists no other solution that is at least as good in respect to all the objectives and is strictly better to at least one objective or in other words:

Let x^p be feasible to MOLP. Then x^p is a pareto-optimal solution to MOLP if for *any other feasible solution $\tilde{x}$, if there is some index k for which $c^k\tilde{x} < c^kx^p$. there is another index l for which $c^l\tilde{x} > c^lx^p$.*

For any class of a multi-objective optimisation decision problem, the concept of "Pareto optimality" or "efficiency" plays a key role in decision making. In such cases the decision problem becomes that of computing a set of Pareto optimal points showing the trade-off between the different objectives.

The following methods can be used to obtain a MOLP solution see Kornbluth [48], Ignizio [41], Ehrgott [24].

Goal programming

The origins of Goal Programming date back to the work of Charnes, Cooper and Ferguson [17]. The idea of goal programming is to set goals for the objective values and try to meet these goals rather than optimise all of the objectives.

Let z_i be set of multiple objective functions which need to be optimised. Then for each of the z_i a goal or target value g_i is assigned.

MOLP$_g$:

$$\begin{array}{lcl}
\min & \sum_i(\alpha_i u_i^+ + \beta_i u_i^-) & \\
s.t. & Ax = b & \\
 & z_i - g_i = u_i^+ - u_i^- & \forall i \\
 & x_i, u_i^+, u_i^- \geq 0, & \forall i
\end{array}$$

where α_i, β_i are penalty terms. If $z_i - g_i \leq 0$ then $u_i^+ = 0$ and $u_i^- = -z_i + g_i$. In this case, the added constraint was an equality. Depending on which goal type constraints are introduced, different objectives are formed. Table 1.1 summarises the three different categories:

Goal type	Goal constraint	Goal variable to be minimised
$z_i \leq g_i$	$z_i - (u_i^+ - u_i^-) = g_i$	u_i^+
$z_i \geq g_i$	$z_i - (u_i^+ - u_i^-) = g_i$	u_i^-
$z_i = g_i$	$z_i - (u_i^+ - u_i^-) = g_i$	$u_i^+ + u_i^-$

Table 1.1: Goal programming categories

The MOP problem can be restated as a goal programming problem where each of the objectives has to be verified as a goal constraint. This can be extended to two-sided bounding of goal values. This is an effective tool for dealing with inconsistent objectives and constraints. MOP formulated in goal programming terms is often referred to as elastic programming.

Preemptive goal programming

Preemptive goal programming minimises each constraint in a given priority order, maintaining all previous objective function values while optimising the next constraint. More specifically, the following four steps are carried out:

1. Prioritise the objectives in to a set order $z^1, z^2, \ldots, z^r$.

2. Minimise the first objective z^1 subject to the original set of constraints and

set z_*^1 to be the optimal objective value.

3. Add the constraint $\sum_{j=1}^n c_j^1 x_j \leq z_*^1$, and minimise z^2 including the additional constraint.

4. Continue adding constraints $\sum_{j=1}^n c_j^i x_j \leq z_*^i$ and minimising z^{i+1} subject to the original constraints and all of the added constraints until all objectives are optimised.

This will give a goal-optimal solution. If any of the LPs solve with a unique optimum, then none of the succeeding objectives are relevant in determining the solution.

Weighting objectives

Another way of dealing with multiple objectives is to give each objective i a positive weight α_i, and minimise the weighted sum $(\alpha_1 c^1 + \cdots + \alpha_r c^r)x$.

Preemptive goal-programming solutions can also be modelled by weighted objectives, simply by giving weight $\alpha_i = \epsilon^i$ to objective c^i , for sufficiently small $\epsilon > 0$. This guarantees that the objectives will be minimised in priority order.

MOLP$_\alpha$:

$$\begin{aligned} \min \quad & \sum_{i=1}^{r} \alpha_i c^i x \\ s.t. \quad & Ax = b, \\ & x \geq 0, \end{aligned}$$

Pareto-optimal paths

Identifying the set of all Pareto-optimal solutions for a multi objective LP is fairly complex, especially when there are several objectives. The Pareto-optimal solutions form a path in the polyhedron.

Suppose the Pareto-optimal solutions for two objectives c^1 and c^2 are of interest. Assuming that there exist the positive multipliers α_1, α_2, the set of of optimal solutions to the LPs are generated by:

$$\begin{aligned} \max \quad & z = z_1 + z_2 = (\alpha^1 c^1 + \alpha^2 c^2)x \\ s.t. \quad & Ax = b, \\ & x \geq 0, \end{aligned}$$

Since the optimal solution is not affected by the positive multipliers of the objective coefficients then, without loss of generality, they can be normalised by $\alpha^1 + \alpha^2 = 1$. Thus, the objective function vector will be $(1 - \alpha c^1) + \alpha c^2 = c^1 + \alpha(c^2 - c^1)$ in the range $0 < \alpha < 1$. In this context the parameterised cost trade-off curves are found, by setting $c = c^1$ and $c^\Delta = c^2 - c^1$. Technically the parameterised cost curve analysis does not give the entire set of solutions, since it only gives the basic optimal solution associated with a given parameterised cost.

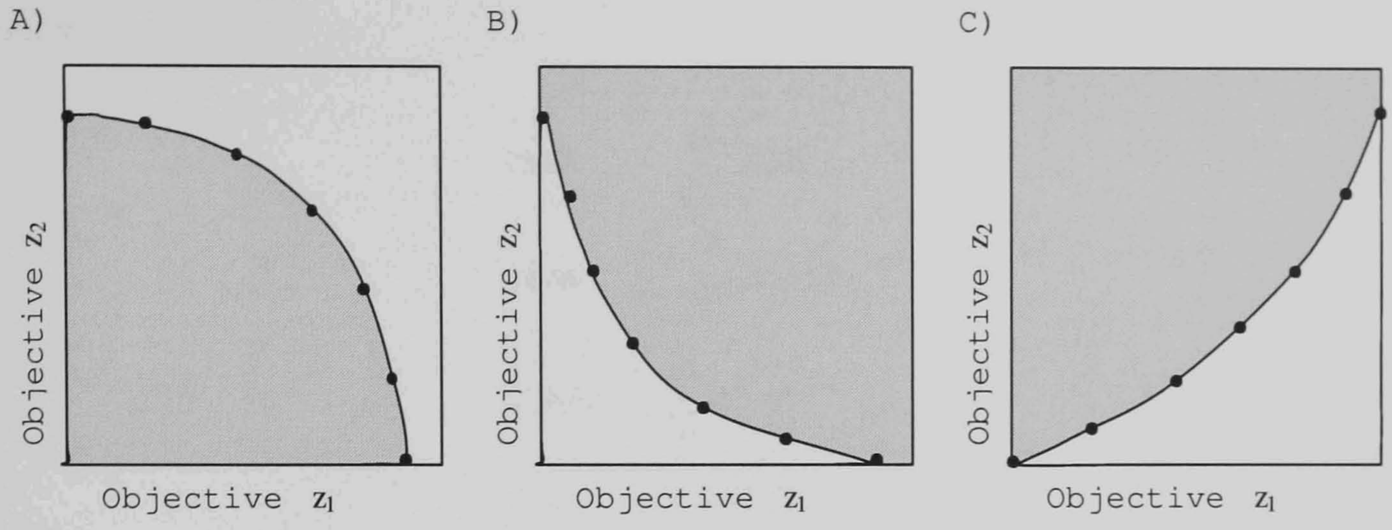

Figure 1.1: Optimisation of two objectives

In figure 1.1, graph A) shows the maximisation of z^1, z^2, B) gives their minimisation whereby C) reflects the trade-off of minimising z^1 and maximising z^2. These solutions are unique between the bend points in the cost curve. The entire edge associated with the given pivot will also be optimal for the parameterised objective, or in this case, Pareto-optimal for the given pair of objectives. Thus the Pareto-optimal path is the set of all Pareto-optimal solutions.

Markowitz model viewed as a bi-objective model

The concepts of multi objective programming especially Pareto optimality are of interest for the portfolio optimisation problem which is investigated in this study. For simplicity, the basic Markowitz model is considered. Observing this model from a multi objective point of view, there can be two objectives identified. The two objectives for the investor are to maximise the return and to minimise the risk of the portfolio. Thus, the described Markowitz model can be written as follows:

$$\begin{array}{ll} \max & \sum\limits_{i=1}^{N} \mu_i x_i \\ \min & \sum\limits_{i=1}^{N} \sum\limits_{j=1}^{N} \sigma_{ij} x_i x_j \\ s.t. & \sum\limits_{i=1}^{N} x_i = 1, \\ & x \geq 0, \end{array}$$

The given formulation is a bi-objective optimisation model. The problem can be solved by applying the given approaches earlier. The commonly used approach computes a family of parametric quadratic programming problems which traces out a curve in the risk-return space. If the solutions - risk versus return - are plotted, it forms the efficient frontier (EF) which is Pareto-optimal. Figure 1.2 visualises the trade-off between the risk an investor has to take to gain a certain return. The EF is the upper boundary of the minimum variance set which is the left boundary of the feasible set.

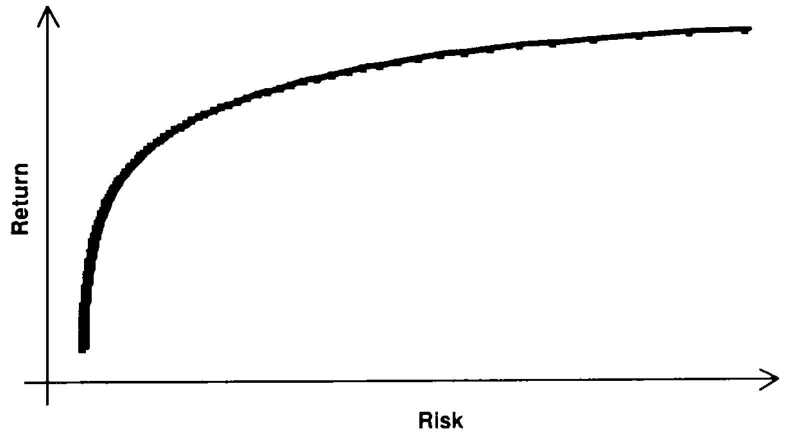

Figure 1.2: Trade-off between risk and return

The mean-variance criterion can be associated with the expected utility approach. Under the assumption that the future wealth of a portfolio is uncertain. utility functions are used to rank the random wealth levels. By Luenberger [59], a utility function is a function U defined on the real numbers representing possible wealth levels and giving a real value. After the definition of the utility function, all alternative random wealth levels are ranked by evaluating their expected utility values. In the case of the Markowitz portfolio problem, there are two ways of applying the expected utility approach to map the mean-variance criterion. Either a quadratic utility function is assumed or the random returns follow a normal distribution. Already in 1969, Hanoch and Levy [38] showed that in the case of a normal distribution the mean variance criterion is valid for the efficiency criterion for any investor's utility function. Later, a comparative study of utility functions was done by Kallberg and Ziemba [46] in which they show that choosing absolute risk aversion indices results in an optimal portfolio. This is regardless of the functional form and the parameters of the utility function. Thus, by building a concave utility function, the mean variance approach is justified.

In this thesis, a portfolio is modelled using the MV approach and solved using MP techniques. The family of portfolios which traces out the efficient frontiers are then solved. The efficient frontier is computed for the continuous QP and discrete QMIP using MP based algorithms.

1.4 Structure of the thesis

This thesis addresses the computational aspects related to portfolio optimisation.
In chapter 2, the mathematical programming aspects that are the foundation for the subsequent chapters are presented.
In chapter 3, the basic concepts for portfolio planning are stated and its extensions. This leads up to the global model proposed for UBS Warburg. For the later studies, the test data sets are described which include the QP repository by Maros et al. [69] and QP/QMIP data sets obtained from the portfolio optimisation applications.
In chapter 4, an overall description about existing QP solvers are discussed where in depth the sparse simplex and interior point method are stated followed by computational experiments of the two solution algorithms. On the basis of these results, a crossover from IPM to SSX is described. This is followed by a computational study on the provided data sets of the three solution algorithms.
Chapter 5 gives a description of the existing solution algorithms for discrete problems and subsequently, a computational study is made. This chapter shows as well a specialised heuristic for the discrete portfolio problem and results.
In chapter 6, a discussion of the continuous and discrete efficient frontier is presented. A computational study stating different approaches for the computation and its results is expatiated.
Chapter 7 concludes the investigations, summarising the thesis and its contributions.

Chapter 2

Mathematical programming

Mathematical Programming refers to the study of optimisation problems with regard to their mathematical properties and formulations. It also includes the development and implementation of algorithms to solve these problems along with methodology of testing these algorithms. This thesis investigates NLPs in which the objective function is quadratic and the constraints linear. These are often referred to as QPs. The problems can be categorised by the decision variables used, into discrete or continuous problems. This thesis discusses the discrete and continuous QP problems which are a special case of constrained optimisation problems.

In this chapter, a subset of the well understood mathematical problems are defined and their optimality criteria set out. These results are used in the algorithmic related sections of this thesis.

2.1 Convex programming

Convex programming is a class of optimisation problem concerned with the minimisation of a convex function on a convex set. For the formulation of the convex programming problem, the definition and properties of a convex set and function are defined. The basic concepts for convex analysis are summarised by Rockafellar in [83], Papadimitriou and Steiglitz [81].

Convex analysis

Definition 2.1. A subset C of $\mathbb{R}^n$ is said to be convex if

$$(1-\lambda)x + \lambda y \in C$$

forall $x \in C$, $y \in C$ and $0 < \lambda < 1$.

Definition 2.2. A function f is said to be convex if and only if

$$f((1-\lambda)x + \lambda y) \leq (1-\lambda)f(x) + \lambda f(y)$$

where $0 < \lambda < 1$.

Proposition 2.1. *If the function $g(x)$ is concave then $h(x) = -g(x)$ is convex on $C = \{x : g(x) > 0\}$.*

Keeping these statements in mind, the convex programming problem can be now formulated.

Convex programming problem

The nonlinear programming problem is reconsidered:

$$\min{}_{x \in X} f(x) \tag{2.1}$$

where $X := \{x \in \mathbb{R}^n | g_i(x) = b_i; i = 1, ...l; h_i(x) \geq b_i, i = l + 1, ..., m\}$.

If

1. the objective function $f(x)$ is a convex function and

2. the set X as defined above is a convex set

then the corresponding optimisation problem is known as a convex programming problem see Luenberger [58], Bertsekas [10].

An important property of a convex programming problem is that a local optimum solution of the problem is also a global optimum.

The above set X may be considered to be the intersection of m sets in the following way:

Let

$$X_i = \{x | g_i(x) = b_i\} \text{ for } i = 1, ..., l$$

and

$$X_i = \{x | h_i(x) \geq b_i\} \text{ for } i = l + 1, ..., m$$

Then

$$X = \bigcap_{i=1,...,m} X_i$$

If X_i is convex for all x then for our example (NLP) this implies that g_i are linear functions and $-h_i$ are convex functions.

2.2 Optimality conditions and duality

The Karush-Kuhn-Tucker conditions

The Karush-Kuhn-Tucker (KKT) conditions [50] specify the mathematical properties of an optimal solution. The first order necessary optimality conditions are fundamental for many solution methods. For the NLP problems with inequality and equality constraints and a smooth objective function $f(x)$, the strong local, weak local and global minimum are defined such that:

Definition 2.3. Let $N(x^*, \delta)$,$\delta > 0$ be the set of all feasible points in the neighbourhood of x^*. Then x^* is a strong local minimum if $f(x^*) < f(x) \forall x \in N(x^*, \delta), x \neq x^*$. If $f(x^*) \leq f(x) \forall x \in N(x^*, \delta), x \neq x^*$ then x^* is a weak local minimum. If $f(x^*) < f(x)$ for all feasible points, then x^* is a global minimum.

In the case of a convex objective function and constraints defining a convex set, there are no local minima since any minimum is a global minimum.

Definition 2.4. Let x^* be a point satisfying the constraints of the NLP(1.1) and let I' be the set of indices i for which $g_i(x^*) = 0$. Then x^* is said to be a *regular point* of the NLP constraints of 1.1 if the gradient vectors $\nabla h_i(x^*)$ for $1 \leq i \leq l$, $\nabla g_i(x^*)$ for $i \in I'$ are linearly independent.

The KKT necessary conditions for the NLP problem where f, g_i, h_i are continuously differentiable functions are:

Proposition 2.2. *Let x^* be a relative minimum point for the NLP problem 1.1 and suppose x^* is a regular point for the constraints. Then there is a vector $\lambda \in E^l$*

and a vector with $\mu \in E^m$ with $\mu \geq 0$ such that

$$\nabla f(x^*) + \lambda^T \nabla h(x^*) + \mu^T \nabla g(x^*) = 0 \tag{2.2}$$

$$\mu^T g(x^*) = 0 \tag{2.3}$$

The KKT conditions reflect the complementarity of the variables of the primal and dual form of the underlying problem. Certain optimisation problems, such as LP and QP take advantage of the dual form of the given problem. The dual problem is constructed with the coefficients that describe the original problem, known as the primal problem and possesses symmetry relations with the original problem. The duality provides an alternative means of solving the original problem.

Duality

Consider the NLP problem with linear constraints where e_i, a_j and d_i, b_j are given vectors and scalars respectively. The function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a convex continuously differentiable function and X is a polyhedral set. The Lagrangian function is defined as

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{l} \lambda_i(e_i'x - d_i) + \sum_{j=1}^{m} \mu_i(a_i'x - d_i);$$

then the dual function is defined by

$$q(\lambda, \mu) = \ \inf{}_{x \in X} L(x, \lambda, \mu). \tag{2.4}$$

and the dual problem is

$$\begin{aligned} \max \quad & q(\lambda, \mu) \\ \text{s.t.} \quad & \lambda \in \mathbb{R}^m, \mu \geq 0. \end{aligned} \tag{2.5}$$

The essential duality properties can be summarised in the following way:

- If the primal problem has an optimal solution, the dual problem also has an optimal solution and the corresponding optimal values are equal.
- In order for x^* to be an optimal primal solution and (λ^*, μ^*) to be an optimal dual solution, it is necessary and sufficient that x^* is primal feasible,$\mu^* \geq 0, \mu_j^* = 0$ for all $j \notin A(x^*)$ and

$$x^* = argmin_{x \in X} L(x, \lambda^*, \mu^*). \tag{2.6}$$

2.3 Quadratic programming

The quadratic programming (QP) problem is a special case of the convex NLP problem. The objective function $f(x)$ combines a linear term with a quadratic form. An essential result of convex analysis is that the minimisation of a convex function subject to a convex constrained set leads to a convex programming problem. If the quadratic form, given by the matrix Q, is positive semi definite, then it is convex. Therefore, the combination of the linear term which is a special case of a convex function and the quadratic term, leads to a convex objective function. Given a linear (convex) constrained set, this becomes a convex programming problem and the KKT optimality criteria apply.

The mathematical formulation of the standard QP problem is:

QP1:

$$\begin{aligned} \min\ & c^T x + \frac{1}{2} x^T Q x \\ \text{s.t. } & Ax \geq b, \\ & x \geq 0, \end{aligned} \tag{2.7}$$

where $c, x \in \mathbb{R}^n; b \in \mathbb{R}^m; A \in \mathbb{R}^{m \times n}$ and $Q \in \mathbb{R}^{n \times n}$ is positive semi definite.
However, an earlier form due Wolfe [92], van de Panne and Whinston [87] is the following quadratic programming representation:

QP2:

$$\begin{aligned} \max\ & c^T x - \frac{1}{2} x^T Q x \\ \text{s.t. } & Ax \leq b, \\ & x \geq 0, \end{aligned} \tag{2.8}$$

The two presented QP problems are both commonly used in the literature whereby QP1 is widely used amongst the interior point researchers and QP2 by the simplex community.

The first order partial derivatives of the objective functions which relate to the optimality conditions are linear satisfying the KKT conditions and require the solution of linear equations in non-negative variables. Consequently, most of the well established approaches for solving LPs can be applied to the solution of QPs.

2.4 Mixed integer programming

The integer programming (IP) problem is defined by a linear objective function, linear constraints and discrete decision variables. If all decision variables are integer, the IP problem is called a pure integer programming (PIP) problem while if some (but not all) decision variables are discrete, then it is a mixed integer programming (MIP) problem.

Let $N = \{1, ..., n\}$ denote the set of indices of all decision variables and N' a subset of these indices corresponding to the discrete decision variables then the MIP problem is stated as:

MIP:

$$\begin{array}{lll} \min & \sum\limits_{j=1}^{n} c_j x_j & \\ \text{s.t.} & \sum\limits_{j=1}^{n} a_{ij} x_j = b_i & i = 1, ..., m \\ & x_j \geq 0, & j = 1, ..., n \\ & j \in N & \text{and integer for } j \in N'; N' \subset N. \end{array} \tag{2.9}$$

Note for $j \in N'; N' = N$, the MIP problem is a PIP problem. The set N can be divided further into the following sets:

$B = \{1, ..., \lvert B \rvert\}$	Index set for binary variables
$I = \{\lvert B \rvert + 1, ..., \lvert I \rvert + \lvert B \rvert\}$	Index set for integer variables
$C = \{\lvert I \rvert + \lvert B \rvert + 1, ..., \lvert I \rvert + \lvert B \rvert + \lvert C \rvert\}$	Index set for continuous variables
$N = B \cup I \cup C$	Index set for all variables
$N' = B \cup I$	Index set for the discrete variables N

(2.10)

A feasible solution x is any vector which satisfies the restrictions of the MIP. Whereby the feasible region is the set of all feasible points; the infimum of all feasible points yield the optimal solution.

2.5 Quadratic mixed integer programming

The integer programming problem with a quadratic objective function, linear constraints and some discrete decision variables is known as a quadratic mixed integer programming (QMIP) problem. The QMIP problem is formulated as:

QMIP:

$$\begin{array}{lll} \min & \sum_{j=1}^{n} c_j x_j + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_i x_j & \\ \text{s.t.} & \sum_{j=1}^{n} a_{ij} x_j = b_i & \forall i = 1, ..., m \\ & x_j \geq 0, & \\ & j \in N & \text{and integer for} j \in N'; N' \subset N. \end{array} \quad (2.11)$$

where the index sets are defined in section 2.4 conditions 2.10.

The stated QMIP problem is a integer problem therefore the feasible region is not a convex set. Thus, the properties of the global optimality for convex programming problems cannot be immediately applied.

2.6 Computational complexity

In the 1970's, the theory of computational complexity was developed by Garey and Johnson (1979) [27], Horowitz and Sahni (1978) [40]. The described method attempts to predict the level of difficulty of solving a problem with respect to its size. A function $f(n)$ for a problem of size n is to be determined which represents the maximum number of computer operations such as addition, subtraction, division, multiplication used to solve the given problem. The assumption is made that each operation is completed in a unit time. This ensures that the solution time for every instance of a model of size n does not exceed $f(n)$.

Consider a problem of size n, then the function $f(n)$ is of complexity $\mathcal{O}(g(n))$ if

$$|f(n)| \leq c|g(n)|$$

where c is a positive value that is independent of the problem size. The constant n is determined by the type of the computer processor being used to solve the model.

The complexity function is linear for $\mathcal{O}(n)$, polynomial for $\mathcal{O}(n^k)$ and exponential for $\mathcal{O}(k^n)$ where k is a constant.

A given problem is said to be

- P-class if it can be solved in polynomial time and
- NP-class if it is solvable in exponential time.

Another way of describing the complexity is by looking into the reducibility or transformability of a problem. A problem is known to be

- P_1 intractable in polynomial time but could be reduced in polynomial time to another problem,
- P_2 using a polynomial time algorithm

If a problem P is such that every problem in NP is polynomially transformable to P, then P is referred to as NP-hard which is also known as NP-complete. Generally, MIP's fall into the class of NP-hard.

If the portfolio problem covers the buy-in, cardinality and roundlot constraints, the problem increases in size and becomes computationally complex. In 1999, Mansini and Speranza [62] showed that finding a feasible solution to the portfolio selection problem with roundlots is NP-complete.

Chapter 3

Portfolio planning in practice and the collection of data sets

Portfolio planning is an important problem for the finance industry and has gained considerable acceptance by (fund) managers in this industry sector. This thesis focuses on solving portfolio planning models. In this chapter, the portfolio planning models are described and data sets taken from real world models are summarised. These data sets are used in the subsequent chapters dealing with solving these real world applications.

The first two sections of this chapter describe the mean-variance model and the factor model. The first model is traditionally used and minimises the risk exposure ensuring a certain level of return. The variance/co-variance which is in this case the symmetric measure for risk, is decomposed into identified factors resulting in the factor representation of the portfolio planning model. The subsequent section shows how the portfolio models can be extended to capture the real world requirements. In the last part of the chapter, an implemented model is presented.

3.1 The mean-variance model

The classical mean variance (MV) model of Markowitz [63, 66] assumes that there are N risky assets and no risk free asset in the liquid markets. The prices of all assets are exogenous given within one time period. All markets are liquid. Furthermore, the portfolios are selected according to the mean-variance criterion as follows:

MV1:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N}\sum_{j=1}^{N} \sigma_{ij} x_i x_j && (3.1) \\ \text{s.t.} \quad & \sum_{i=1}^{N} x_i = 1, && (3.2) \\ & \sum_{i=1}^{N} \mu_i x_i = \rho, && (3.3) \\ & x_i \geq 0, \qquad i = 1, ..., N && (3.4) \end{aligned}$$

where $x \in \mathbb{R}^N$ denotes the portfolio weights, $\mu \in \mathbb{R}^N$ the return, $V \in \mathbb{R}^{N \times N}$ the covariance matrix with elements σ_{ij} which is positive definite and ρ the desired level of return. The objective function (3.1) is a quadratic function minimising the risk of the portfolio subject to the requirement of complete investment (3.2) and that a certain level of return needs to be met (3.3). The non-negativity constraint (3.4) does not allow short sales.

Estimation of data

The estimation of the data parameters is traditionally done using historical data. This approach calculates the average of returns from a certain time period and then the covariance is built. Another way of estimating the data uses the Bayes-Stein estimators [45] assuming that most of the returns of the assets are negatively skewed. Lower partial moments are used and the semi-variance constructed. By using the Bayes-Stein estimators, it tries to reduce the degree of estimation error of correlated variables and to eliminate extreme inputs which in turn gives a lower variation of parameters. Furthermore, the means of the assets are shrunk towards a global mean. In this study, the focus lies in symmetric risk measures which guaranties a positive definite matrix in the QP(QMIP) framework.

Diversification

Systematic risk (due to market behaviour) refers to risk which cannot be eliminated. The remaining risks are collectively called unsystematic risk which by diversification can be reduced or eliminated. In order to provide a wide scope of diversification of a portfolio, it is better to consider a relatively large universe of stocks. In section 3.3, models are introduced which address some of these shortcomings.

Investors are mainly interested in the risk-return trade-off. This is modelled by introducing a parameter $\lambda \in [0, 1]$ and putting the budget constraint into the objective. The MV-model can then be reformulated in the following way:

MV2:

$$\min \quad \lambda \sum_{i=1}^{N} \sum_{j=1}^{N} \sigma_{ij} x_i x_j - (1-\lambda) \sum_{i=1}^{N} \mu_i x_i \tag{3.5}$$

$$s.t. \quad \sum_{i=1}^{N} x_i = 1, \tag{3.6}$$

$$x_i \geq 0, \qquad i = 1, ..., N \tag{3.7}$$

This model comprises a composite objective, the minimisation of risk and the maximisation of the return of the given portfolio. The techniques described in the introduction can be used to solve this problem and the Pareto optimal path can be traced out showing the trade-off between risk and return. This line is called the efficient frontier and is further discussed in chapter 6.

3.2 The factor model

The dense matrix of the quadratic objective in the MV model is from a computational point of view undesirable. The aim is to either reduce the size of the quadratic matrix or make it more sparse. Sharpe [86] introduced in 1971 a single index model. This approach was extended by Rosenberg [85] and others; such that it is assumed that a number of M factors explain a stock's return. Thus, the return r_i for asset i can be split with following the component factor contributions:

$$r_i = \alpha_i + \sum_{m=1}^{M} \beta_{im} f_m + e_i \tag{3.8}$$

In 3.8, α_i is the expected value of the specific return of asset i, β_{im} the sensitivity of asset i to factor m, f_m being the level of the mth factor and e_i the random component of the specific return; Using this representation of returns, the risk matrix can be decomposed in the following way

$$V = B^T F B + D$$

where B is the $M \times N$ matrix of factor sensitivities, F is the $M \times M$ diagonal matrix of factor variances $\sigma^2_{f_M}$, and D is the $N \times N$ diagonal matrix of specific variances σ^2 . Exploiting the composition of the covariance matrix, the portfolio model can be written as follows:

$$\begin{aligned}
\min \quad & \sum_{i=1}^{M} \sum_{j=1}^{M} \beta_i \beta_j \sigma^2_{f_{ij}} + \sum_{i=1}^{N} x_i^2 \sigma^2_{e_i} && && (3.9) \\
s.t. \quad & \sum_{i=1}^{N} x_i = 1 && && (3.10) \\
& \sum_{i=1}^{N} \mu_i x_i \geq \rho, && && (3.11) \\
& \beta_m = \sum_{i=1}^{N} x_i \beta_{im} && m = 1, ..., M && (3.12) \\
& x_i \geq 0 && i = 1, ..., N && (3.13)
\end{aligned}$$

In comparison to the earlier MV1/MV2 model, the objective function 3.9 is now a quadratic function minimising the risk which is decomposed into the factor related and the random risk and additional constraints. This reduces the objective function to a sum of squares with $M+N$ terms and in turn, M additional variables and M constraints 3.12 are added. The restrictions 3.11, 3.13 remain the same as in MV1.

3.3 Extensions of the portfolio optimisation problem

The classical Markowitz model does not restrict for instance investment nor takes into account time lags in trading. The introduced portfolio optimisation model only restricts the budget and does not allow short sales. Obviously, these restrictions don't reflect the reality. In practical investment problems, restrictions are often imposed either by regulatory authorities or there are internal restrictions in the firm which are to be respected. Zenios [96], Jobst et.al. [43], Luenberger [59] amongst others provide a good insight into the possible portfolio optimisation extensions.

Restrictions can be imposed in various respects:

- *Liquidity restrictions*:

  Typically, this kind of restriction has to be addressed in dynamic models of portfolio selection in which borrowing and short selling of stocks are allowed up to a fixed proportion.

- *Buy-in threshold constraints*:

  These constraints define the minimum level at which an asset can be purchased. It eliminates the problem where unrealistically small trades can be included in an optimum portfolio.

- *Upper threshold constraints*:

  These constraints are similar to the above with the only difference that the portfolio weights have an upper limit.

- *Sector/Industry (joint) constraint*:

  The securities in a portfolio are grouped according to their industry and sector. With joint constraints the minimum and maximum exposure of a sector or industry are limited.

- *Penalties*:

  In large portfolio optimisation models not every bound can be met. In case of breaking the introduced minimum or maximum exposures, respectively the under- or overshooting is penalised with a defined cost.

- *Cardinality constraint*:

  To find a manageable portfolio out of a large universe of stocks, it is sensible to limit the number of stocks in the portfolio.

In this thesis the above constraints are used and by all means these are just a subset of all possible constraints which could be considered. The following are a number of restrictions which could be added to the initial portfolio optimisation problem.

- *Roundlots*

  Roundlots are restrictions used to define the basic unit of investment. Investors are allowed only to make transactions in multiples of these roundlots. reflecting cash or a certain number of stocks.

- *Portfolio dedication*:

  Portfolio dedication matches the cash flow between borrowing and reinvestment. In other words the stream of liabilities needs to be met by holding at each maturity a certain amount of investment.

- *Portfolio immunisation*:

  An investment strategy that tries to protect the expected yield from a security or portfolio of securities by acquiring those securities whose duration equals the length of the investors planned holding period .

The discussed possible restrictions on a portfolio optimisation model are widely accepted.

3.4 Industrial implementation: A discrete portfolio optimisation model

An industrial strength implementation of a portfolio optimisation model was developed using the model specification format defined by Northfield System Inc. [42]. This model includes the requirements given by UBS Warburg.
There are two aspects of the model, namely, (1) index tracking and (2) portfolio rebalancing.

1. Index tracking

   This implies that the portfolio should behave in a way similar to an index or a given benchmark. The implementation of this technique is achieved by minimising the 'tracking error' in respect to the mean and co-variance.

2. Portfolio rebalancing

   Since the portfolio model is used as a decision tool it is applied to a given fund at regular time intervals. This means the current asset composition of the portfolio needs to be 'rebalanced' by buying and selling stocks.

The implementation of the model incorporating the data mart of Northfield System requires as an initial step the definition of the following sets, indices and data sets. For diversification, each stock in the universe belongs to a certain industry which in turn belongs to a certain sector. The return behaviour of the stocks on the other hand are given by a factor model which has at most 22 factors. The first factor is the reference factor, the next seven belong to an index, the subsequent 7 factors are associated with countries and the last 7 are structural. The decomposed variance/covariance has the form $V = B^T F B + D$ where $V \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{M\times N}$, $F \in \mathbb{R}^{M\times M}$ is a block angular and $D \in \mathbb{R}^{N\times N}$ is a diagonal matrix. The explicit representation of the variance/covariance matrix computed through a regression model has two shortcomings (a) the Q-matrix is dense and (b) the coefficients may contain numerical errors. The factor model has the benefit of creating a sparse and compact representation of the objective function. The solver algorithms, SSX or IPM, are able to take advantage of the sparse factored form representation of the portfolio variance/covariance and process the model efficiently. As mentioned in earlier sections, the sparsity plays an essential role for fast computations using IPM or SSX.

The data mart for the basic structure of the portfolio optimisation model is endogenous given by Northfield Systems and the following sets, indices and data tables can be identified:

Sets and indices:

$i \in I$ denotes the universe of stocks

$j \in J$ denotes the set of industries

$k \in K$ denotes the set of sectors

$i' \in I'_j$ denotes the set of stocks which belong to industry j $\forall j \in J$

$i'' \in I''_k$ denotes the set of stocks which belong to sector k $\forall k \in K$

$f \in \phi$ denotes the set of model factors

Data tables:

$b_i \in \Re$ denotes the benchmark weight for stock $i \in I$

$r_i \in \Re$ denotes the expected return for stock $i \in I$

$o_i \in \Re$ denotes the initial portfolio weight for stock $i \in I$

B denotes the model factor coefficient matrix $|I| \times |\phi|$

F denotes the model factor correlation matrix $|\phi| \times |\phi|$

D denotes the variance matrix $|I| \times |I|$

l_i denotes the minimum weight of stock i in the portfolio

u_i denotes the maximum weight of stock i in the portfolio

ln_j denotes the minimum desired exposure for industry j

un_j denotes the maximum desired exposure for industry j

ls_k denotes the minimum desired exposure for sector k

us_k denotes the maximum desired exposure for sector k

$l\phi_f$ denotes the minimum desired exposure for factor f

$u\phi_f$ denotes the maximum desired exposure for factor f

lt_i denotes the lower threshold value of stock i

ut_i denotes the upper threshold value of stock i

p_1^+ denotes the unit penalty for overshooting the upper bound portfolio weight for factor f $\forall f \in \phi$

p_1^- denotes the unit penalty for undershooting the lower bound portfolio weight for factor f $\forall f \in \phi$

p_2^+ denotes the unit penalty for overshooting the upper bound portfolio weight for industry j $\forall j \in J$

p_2^- denotes the unit penalty for undershooting the lower bound portfolio weight for industry j $\forall j \in J$

p_3^+ denotes the unit penalty for overshooting the upper bound portfolio weight for sector k $\forall k \in K$

p_3^- denotes the unit penalty for undershooting the lower bound portfolio weight for sector k $\forall k \in K$

c denotes the cardinality

$\lambda \in [0, 1]$ denotes a risk aversion coefficient

The decision variables are:

x_i denotes the portfolio weight for stock i

xs_i denotes weight of stock i sold

xl_i denotes weight of stock i bought

y_f denotes amount of factor f in the portfolio

xo_i^+ denotes the amount by which the desired lower bound $l\phi_f$ is under achieved

xu_i^- denotes the amount by which the desired upper bound $u\phi_f$ is over achieved

d_i^- denotes the range of the portfolio weight x_i

xon_j^+ denotes the amount by which the desired lower bound ln_j is under achieved

xun_j^- denotes the amount by which the desired upper bound un_j is over achieved

dn_j^- denotes the range of the industry $x_{i'}$

xok_k^+ denotes the amount by which the desired lower bound lk_k is under achieved

xuk_k^- denotes the amount by which the desired upper bound uk_k is over achieved

dk_k^- denotes the range of the sector $x_{i''}$

δ_i denotes a binary variable indicating if stock i in the portfolio or not

The objective function

The given data sets and indices result in the following formulation of the objective function:

$$
\begin{aligned}
\mathbf{min} \quad (- \quad & r^T x + r^T b)(1-\lambda) && (3.14)\\
- \quad & 2\lambda b^T B^T F B x - 2\lambda b^T D x && (3.15)\\
+ \quad & \lambda y^T F y + \lambda x^T D x && (3.16)\\
+ \quad & \lambda b^T B^T F B b + \lambda b^T D b && (3.17)\\
+ \quad & p_1^- xon^- + p_1^+ xun^+ + p_2^- xok^- + p_2^+ xuk^+ && (3.18)\\
+ \quad & p_3^+ xo^+ + p_3^- xu^- && (3.19)
\end{aligned}
$$

The original objective function has the form $r^T(x-b)+(x-b)V(x-b)$ with $V = BFB^T + D$. Writing the objective function explicitly in its risk component results in:

$$
\begin{aligned}
- \quad & r^T x + r^T b && (3.20)\\
- \quad & 2b^T B^T F B x - 2b^T D x && (3.21)\\
+ \quad & y^T F y + x^T D x && (3.22)\\
+ \quad & b^T B^T F B b + b^T D b && (3.23)
\end{aligned}
$$

The first terms of (3.20) track the index in respect to its return whereby the remaining terms are index tracking the covariance. The actual implemented objective function has additional terms considering penalties for over- and undershooting of given desired limits and takes as well into account the risk aversion in the form of the constant λ . The objective function can be split up into a linear

(3.14), a quadratic (3.15, 3.16, 3.17) and a penalty term (3.18, 3.19). This representation of the objective allows a large universe of stocks to be processed. reduces the size of the quadratic part of the objective and gives insight into the return and risk of the optimised portfolio.

The constraints

s.t.:

$$x_i = o_i - xs_i + xl_i \qquad \forall i \in I \tag{3.24}$$

$$\sum_{i \in I} x_i = 1 \tag{3.25}$$

$$y_f = \sum_{i \in I} B_{if} x_i \qquad \forall f \in \phi \tag{3.26}$$

$$x_i + xu_i^- - xo_i^+ + d_i = u_i \qquad \forall i \in I \tag{3.27}$$

$$\sum_{i' \in I'_j} x_{i'} + xun_j^- - xon_j^+ + dn_j = un_j \qquad \forall j \in J \tag{3.28}$$

$$\sum_{i'' \in I''_k} x_{i''} + xuk_k^- - xok_k^+ + dk_k = us_k \qquad \forall k \in K \tag{3.29}$$

$$\sum_{i \in I} \delta_i \leq c \tag{3.30}$$

$$x_i \leq ut_i \delta_i \qquad \forall i \in I \tag{3.31}$$

$$x_i \geq lt_i \delta_i \qquad \forall i \in I \tag{3.32}$$

and

$$d_i \leq u_i - l_i \qquad \forall i \in I \tag{3.33}$$

$$dn_j \leq un_j - ln_j \qquad \forall j \in J \tag{3.34}$$

$$dk_k \leq us_k - ls_k \qquad \forall k \in K \tag{3.35}$$

$$xs_i, xl_i \geq 0 \qquad \forall i \in I \tag{3.36}$$

The restriction(3.25) is present in each of the shown models and ensures that everything is invested. The rebalancing constraint(3.24) keeps the balance between the initial portfolio, the stocks bought and sold and the resulting stocks held in the portfolio. Together with the positive restriction (3.36) it makes sure that short selling is not allowed.

The buy-in constraints (3.31) and (3.32) define the upper and lower threshold for a stock in the universe. These are represented by binary variables and a pair of bounding restrictions. The introduction of buy-in thresholds enables binary variables to limit the number of unique stocks in the portfolio by counting the binary variables and fixing it to a fixed number of assets c.

From an algorithm point of view, the sparsity of the quadratic matrix plays an essential role and therefore, the restriction (3.26) is introduced substituting Bx in the objective function and adding a new decision variable y. This reduces the quadratic term of the objective and therefore the computation of the factorisation of a dense Q is avoided.

Earlier in this chapter, it was shown that diversification of the portfolio is very important. This is achieved by dividing the universe of stocks into industries and sectors and applying bounds to the value of the decision variables for the portfolio weight x_i held in different industries ($x_{i'}$) and sectors ($x_{i''}$). The constraints (3.27), (3.28) and (3.29) are responsible for the enforced diversification.

For large models, these tight bounds make the problem harder to solve and in some cases impossible to obtain an optimal solution. By making the lower and upper bounds on the decision variables flexible, the problem is solvable to optimality or near optimality. Therefore, elastic or goal constraints are formulated.

In case of not meeting the limits set by the investor, the so called over or undershooting of the bounds is penalised. Depending on the preferences of the investor, the violation of these bounds is severe or not and treated accordingly with different penalties in the objective function (3.18, 3.19) .

3.5 Summary of test models

In this section a summary of all the test models which have been collected to support this investigation are given.

A convex quadratic programming problems repository

This set of models were collected by Maros and Meszaros [69] as benchmark models to report properties of QPs.

In reality, they are adaptations of NETLIB LP models to which quadratic objective terms have been added. In this thesis, a subset of 11 benchmark problems is taken to investigate the properties of different methods. Table 3.1 shows the dimensions of the chosen problems and where the headings are as described below

NAME is the name of the problem,

M is the number of rows in A,

N is the number of variables,

NZ is the number of nonzeros in A,

QN is the number of variables associated with the quadratic term,

QNZ is the number of off-diagonal entries in the lower triangular part of Q and

OPT is the solution value (BPMPD solver [74]).

NAME	M	N	NZ	QN	QNZ	OPT
25fv47	820	1571	10400	446	59053	1.3744448E+07
e226	223	282	2578	67	897	2.1265343E+02
fffff80	524	854	6227	278	1638	8.7314747E+05
scfxm2	660	914	5183	74	1057	2.7776162E+07
scfxm3	990	1371	7777	89	1132	3.0816355E+07
scsd6	147	1350	4316	96	1308	5.0808214E+01
scsd8	397	2750	8584	140	2370	9.4076357E+02
shell	536	1775	3556	405	34385	1.5726368E+12
sierra	1227	2036	7302	122	61	2.3750458E+07
stair	356	467	3856	66	952	7.9854528E+06
standat	359	1075	3031	138	666	6.4118384E+03

Table 3.1: QP model statistics (in *.qps format)

Test data set of practical portfolio models

The portfolio optimisation model explained in the earlier section of this chapter are the basis of the alternative data sets supplied by UBS Warburg. The models dimensions have been carefully chosen so that scale up properties can be studied and where the headings are as described below.

NAME	Name of the problem
M	Number of rows in A
N	Number of variables
NZ	Number of nonzeros in A
QN	Number of variables associated with quadratic terms
QNZ	Number of off-diagonal entries in the lower triangular part of Q
BN	Number of binaries
OPT	Value of the objective function using *FortMP*

For the later comparisons of different solution algorithms, the following models are chosen. QPF$\#n$ indicates that the underlying problem is a factor model and QPMV$\#n$ shows that it is a mean-variance model. Thus, QPF100 problem is the factor model with a stock universe of 100. The formal statement of the models is given in section 6.2.

NAME	M	N	NZ	QN	OPT
QPF50	24	72	436	228	0.19518419
QPF100	24	122	784	278	0.11742109
QPF500	24	522	3960	678	0.86850727E-02
QPF1800	24	1022	8037	1178	-.14128493E-01
QPF4500	24	4522	35641	4678	-.22604763E-01
QPMV50	2	50	100	1275	0.20299250
QPMV100	2	100	200	5050	0.12043239
QPMV500	2	500	1000	125250	0.84802004E-02
QPMV1000	2	1000	2000	500500	-.39287655E-02
QPMV4500	-	-	-	-	-

Table 3.2: Model statistics for the continuous portfolio selection models

Table 3.3 gives the model statistics for discrete factor models either in lambda formulation (QMIPL) or in ρ-formulation (QMIPR). Both models of the QMIPL and QMIPR problem are stated in section 6.3. Depending on the universe of stocks a different cardinality is associated which is summarised in the next table

Number of stocks	50	150	300	600	1200	1800	4500
Cardinality	3	20	30	30	40	40	40

The QMIP problems have the following dimensions:

NAME	M	N	NZ	QN	QNZ	BN
QMIPFL50	125	122	686	72	228	50
QMIPFL150	325	322	1878	172	328	150
QMIPFL300	625	622	3775	322	478	300
QMIPFL600	1225	1222	7746	622	778	600
QMIPFL1200	2425	2422	15606	1222	1378	1200
QMIPFL1800	3625	3622	23352	1822	1978	1800
QMIPFL4500	9025	9022	58141	4522	4678	4500
QMIPFR50	126	122	686	72	228	50
QMIPFR150	326	322	1878	172	328	150
QMIPFR300	626	622	3775	322	478	300
QMIPFR600	1226	1222	7746	622	778	600
QMIPFR1200	2426	2422	15606	1222	1378	1200
QMIPFR1800	3626	3622	23352	1822	1978	1800
QMIPFR4500	9026	9022	58141	4522	4522	4500

Table 3.3: Model statistics for discrete factor models

Chapter 4

Solution methods for QP problems

Because of its many applications, quadratic programming is considered to be an area of specialisation in itself. Moreover, it forms the basis of several general nonlinear programming algorithms.

In this section 4.1, first solution algorithms for linearly constrained optimisation problems with quadratic objective function are described and the first order optimality condition is applied to the QP case. The simplex method, the sparse simplex method and the interior point method are therefore described in sections 4.3, 4.4, 4.5 respectively followed by a computational study which is reported in section 4.6. It follows naturally from these results that a 'cross over' step (also known as basis recovery) from IPM to SSX is desirable; computational investigations of a basis recovery procedure is reported in section 4.7. In section 4.8, a discussion of the algorithms is given for the continuous portfolio selection model.

4.1 Solution algorithms for QP problems

In the literature various methods are described for solving quadratic programming problems. These approaches are usually based on solving the system of equations derived from the KKT optimality conditions. The following approaches can be applied to solve a quadratic programming problem:

Gradient method

Common gradient methods are described in Luenberger [58], Golub and Van Loan [31] and are steepest descent and conjugate gradient. Both are fundamental techniques that are often incorporated into various iterative algorithms.

Steepest descent is one of the oldest and simplest methods. At each iteration of steepest descent, the search direction is taken as the negative gradient $-g_k$ of the objective function at the specific point where the descent direction p_k satisfies $g_k^T p_k > 0$. This inner product is negative if $p_k = -g_k$ and also minimises the inner product for unit-length vectors and, thus gives rise to the name steepest descent. Steps of the steepest descent method are often incorporated into other methods (e.g., Conjugate Gradient, Newton).

The conjugate gradient method was originally designed to minimise convex quadratic functions. The first conjugate gradient iteration is the same as that of the steepest descent, but successive directions are constructed so that they form a set of mutually conjugate vectors with respect to the (positive-definite) Hessian of a general convex quadratic function. Whereas the rate of convergence for steepest descent depends on the ratio of the extremal eigenvalues of the Hessian, the convergence properties of conjugate gradient depend on the entire matrix spectrum. Faster convergence is expected when the eigenvalues are clustered. In particular, if the Hessian has m distinct eigenvalues, convergence to a solution requires m iterations.

The cutting-plane method

The goal of the cutting-plane method is to find a point in a convex set called the target set or to determine that the convex set is empty. The target set can be taken as the set of optimal (or ϵ-suboptimal) points for the problem, and the goal is to find an optimal (or ϵ-suboptimal) point for the given problem. There exists no direct access to any description of the target set (such as the objective and constraint functions in an underlying problem) except through an oracle. A query returns if the point is an element of the convex set or if it returns a separating hyperplane between the point and the convex set. This hyperplane is called a cutting-plane or cut; no such point could be in the target set.

Penalty and barrier method

The main idea is firstly to solve a constrained optimisation problem by solving a sequence of unconstrained optimisation problems, and in the limit, the solutions of the unconstrained problems will converge to the solution of the constrained problem. Secondly, an auxiliary function is used incorporating the objective function together with penalty terms that measure violations of the constraints. Two groups of classical methods can be identified one being the barrier method which imposes a penalty for reaching the boundary of an inequality constraint and the other being the penalty method which imposes a penalty for violating a constraint. Barrier and penalty methods solve a sequence of unconstrained subproblems that gradually approximate the given problem in which an infinite penalty for violating feasibility is replaced by a continuous function that gradually approaches the given function. Barrier (Penalty) methods generate a sequence of iterates that converge to a solution of the constrained problem from the interior (exterior) of the feasible region.

These methods have been applied to solve quadratic programming problems. However, almost all QP solution algorithms are designed to solve KKT in one or the other way.

4.2 KKT conditions and the LCP formulation for the QP problem

Reconsider problem QP2 and assume that a finite optimum solution exists. Then the primal dual formulation of the QP problem using the duality theorem can be stated in the following way:

PQP2:

$$\begin{aligned} \max \quad & f_{PQP2}(x) = c^T x - \tfrac{1}{2} x^T Q x \\ \text{s.t.} \quad & Ax + y = b, \\ & x \geq 0, \end{aligned} \tag{4.1}$$

DQP2:

$$\begin{aligned} \min \quad & f_{DQP2}(x) = b^T v + \tfrac{1}{2} x^T Q x \\ \text{s.t.} \quad & u - A^T v - Qx = -c, \\ & u \geq 0, \end{aligned} \tag{4.2}$$

where $c, x, u \in \mathbb{R}^n; b, y, v \in \mathbb{R}^m; A \in \mathbb{R}^{m \mathrm{x} n}$ and $Q \in \mathbb{R}^{n \mathrm{x} n}$ is symmetric positive semi-definite.

When the objective function $f(x)$ is strictly convex the problem has a unique local minimum which is also the global minimum. To guarantee strict convexity it is a sufficient condition for the quadratic matrix Q to be positive definite.

For this problem the first order necessary conditions are also sufficient for a global minimum. The Lagrangian for the quadratic programming problem is stated as:

$$L(x, v) = c^T x - \frac{1}{2} x^T Q x - v^T (Ax + y - b) + u^T x \tag{4.3}$$

where $v \in \mathbb{R}^m$. Non-negative surplus variables (vector) $u \in \mathbb{R}^n$ and non-negative slack variables (vector) $y \in \mathbb{R}^m$ are introduced forming the following constraints:

$$c - Qx - v^T A + u = 0, \tag{4.4}$$

$$Ax + y - b = 0. \tag{4.5}$$

where I is the identity matrix. Then the KKT conditions for the local maximum can be formulated such that

$$c - Qx - v^T A + u = 0, \tag{4.6}$$

$$Ax + y = b, \tag{4.7}$$

$$v^T (Ax - b) = v^T y = 0, \tag{4.8}$$

$$u^T x = 0, \tag{4.9}$$

$$u, v, x, y \geq 0. \tag{4.10}$$

The KKT conditions for the system can be summarised as a set of $(n + m)$ equations in $2(n + m)$ variables; to which $(n + m)$ complementary conditions are imposed. The Karush-Kuhn-Tucker conditions for the QP turn out to be a set of linear equalities and complementarity constraints. The following relationships are apparent:

- x_j and u_j are complementary for j = 1, . . . , n
- v_i and y_i are complementary for i = 1, . . . , m

The LCP formulation

An alternative representation of the QP problem is the so called linear complementarity problem (LCP).

Consider the linear complementarity problem (LCP).

$$\text{Find } x; s \in \mathbb{R}^n \text{ such that .}$$

$$-Mx + s = q; x \geq 0; s \geq 0; x^T s = 0; \tag{4.11}$$

where $q \in \mathbb{R}^n andM \in \mathbb{R}^{n \times n}$.

4.3 Tableau simplex for QP

The statement and the convergence proof for the simplex method for QP dates back to Dantzig and Cottle [18], Wolfe [94]. Subsequently, van de Panne and Whinston [87], [88] described a comparable simplex (pivotal) method. The results of these papers can be summarised in the following way. The convex QP problem is (a) first set up as a tableau based on the system of equations required to specify the KKT conditions for the primal QP and the dual QP problem and (b) then this system is solved in non-negative variables and the complementary KKT conditions are satisfied. A complementary pivoting strategy is used to tackle the large QP problem.

A number of investigations have used the LCP formulation for the QP problem and have proposed a pivotal algorithm (principal pivot method, Cottle [18] and an alternative pivotal sequence, Lemke [54]) to process and solve QP's. Since these are pivotal methods, they also qualify as simplex approaches.

It is well known that even for relatively small models (100 variables and constraints) the tableau simplex method is not very efficient. To illustrate the general ideas of the simplex algorithm for QP, the simplex method in tableau form is described. Conceptually the standard or the non-standard tableau form is used for solving QP problems. The standard tableau form is given if for each primal variable in the basis its corresponding dual variable is non-basic and a non-standard tableau if the primal and its dual variable appear together in the basis. However, both tableaus generate in each iteration the same solution with different pivot choices. The main difference between those two tableaus is that within a standard tableau the symmetry/skew-symmetry is preserved at each stage, whereby it is not in the non-standard case. Therefore, the non-standard tableau is known as an unsymmetric tableau.

Consider the QP2 problem and the derived KKT's, a new variable u_0 is introduced to represent the objective function value in the tableau such that

$$\begin{aligned}
u_0 &= 2f_{PQP2}(x) = 2c^Tx - x^TQx = c^Tx + x^T(-u + Qx + v^TA) - x^TQx \\
&= c^Tx - x^Tu + x^TAv = c^Tx - x^T + v^TAx \\
&= c^Tx - x^Tu + v^T(b - y) = c^Tx + v^Tb - (x^Tu + v^Ty) \\
&= c^Tx + v^Tb.
\end{aligned}$$

The additional relations which show the linear system of equations and form the tableau can be written in the following way:

$$\begin{aligned}
u_0 &= 2f_{PQP2}(x) = \sum_{j=1}^{n} c_jx_j + \sum_{k=1}^{m} v_kb_k, && (4.12) \\
u_i &= c_i + \sum_{j=1}^{n} q_{ij}x_j + \sum_{k=1}^{m} a_{ki}v_k, \quad i = 1, 2, .., n && (4.13) \\
y_i &= b_i - \sum_{j=1}^{n} a_{ij}x_j, \quad i = 1, 2, ..., m. && (4.14)
\end{aligned}$$

Also the complementary conditions and the non-negative restrictions must hold. A given problem is then set up as an initial tableau which has a symmetry and a skew symmetry property:

Basic variable	Value basic variable		
	x_0	$-x$	$-v$
u_0	0	$-c^T$	$-b^T$
u	$-c$	$-Q$	$-A^T$
y	b	A	0

A solution which satisfies the complementarity relationship, and non-negativity of the solution for the primal variables x, y and dual variables u, v yield the optimal solution by the KKT conditions. The starting point guarantees the full complementarity. For the starting point, certain techniques are used for variables to enter or leave the basis. To enter the basis, the new basis variable is chosen in such a way that variable x or v has the largest negative corresponding u or y variable value. And the variable to leave the basis is chosen by selecting the pivot row such that,

$$\min\left\{\frac{\alpha_{i0}}{\alpha_{ij}}, i \in I_P \middle| \frac{\alpha_{i0}}{\alpha_{ij}} \geq \epsilon_{ratio}, \text{ and } \alpha_{ij} \neq \epsilon_{piv}\right\} = \frac{\alpha_{r0}}{\alpha_{rj}}.$$

where ϵ_{ratio} is the zero tolerance for the ratio test, ϵ_{piv} is the zero tolerance for the pivot, α_{ij} denotes the (i, j) element of any tableau and j is the column corresponding to the variable chosen to enter the basis whereby i is an element of a set of indices I_P for the rows in which the primal variable x or y variable is pivoted.

Van-de Panne and Whinston [88], [87] show that for the standard form the procedure terminates in a finite number of iterations if degeneracy is excluded. The standard simplex tableau for the primal method or the dual method starts with a feasible solution of the primal problem or the dual problem as appropriate.

Taking into consideration that Q is positive definite, the initial tableau for the dual method is restated in the table below. The initial tableau for the dual method is built without drawing any conclusions about correspondence between dual and primal variables.

Basic variable	Value basic variable				
		u	v	x	y
u	$-c$	I	$-A^T$	$-Q$	0
y	b	0	0	$-A$	I

In the dual method, It has to be considered that the y variable can be a basic variable which means that the solution is feasible for the dual problem satisfying the constraints in the DQP2 problem. Since the v variable is not restricted, they are as well candidates to enter the basis. Therefore, the rules for entering and leaving a basis are different to the primal method. The variable to enter the basis is selected among the u-variables, which have negative corresponding x-variables, and among the y-variables which has a corresponding variable with the largest absolute value. Whereby the variable to leave the basis is chosen by applying the following rule:

$$\min \left\{ k\frac{\alpha_{i0}}{\alpha_{ij}}, i \in I_D | k\frac{\alpha_{i0}}{\alpha_{ij}} \geq \epsilon_{ratio}, \text{ and } \alpha_{ij} \neq \epsilon_{piv} \right\} = \frac{\alpha_{r0}}{\alpha_{rj}}. \tag{4.15}$$

where

$$k = \begin{cases} 1 & \text{Basic variable is introduced in a positive direction} \\ -1 & \text{Basic variable is introduced in a negative direction} \end{cases},$$

and where ϵ_{ratio} is the zero tolerance for the ratio test, ϵ_{piv} is the zero tolerance for the pivot, α_{ij} (i, j) element of any intermediate tableau, j the column corresponding to the variable chosen to enter the basis and i the element of the set of indices I_D of the rows in which the dual u or v variable is pivoted. The index set I is partitioned into I_P, I_D where $I = I_P \cup I_D$ and $I_P \cap I_D = \emptyset$

Example

Beale's [4] example, below, is presented to illustrate the set out simplex tableau rules.

$$\begin{aligned} \max f(x) \quad & = 6x_1 - 2x_1^2 + 2x_1x_2 - 2x_2^2 \\ \text{s.t.} \quad & x_1 + x_2 \leq 2, \\ & x_1, x_2 \geq 0, \end{aligned} \tag{4.16}$$

The figure 4.1 shows the unconstrained and constrained maximum of $f(x)$ defining a system of ellipses. Furthermore, the three steps of the iterative process is illustrated.

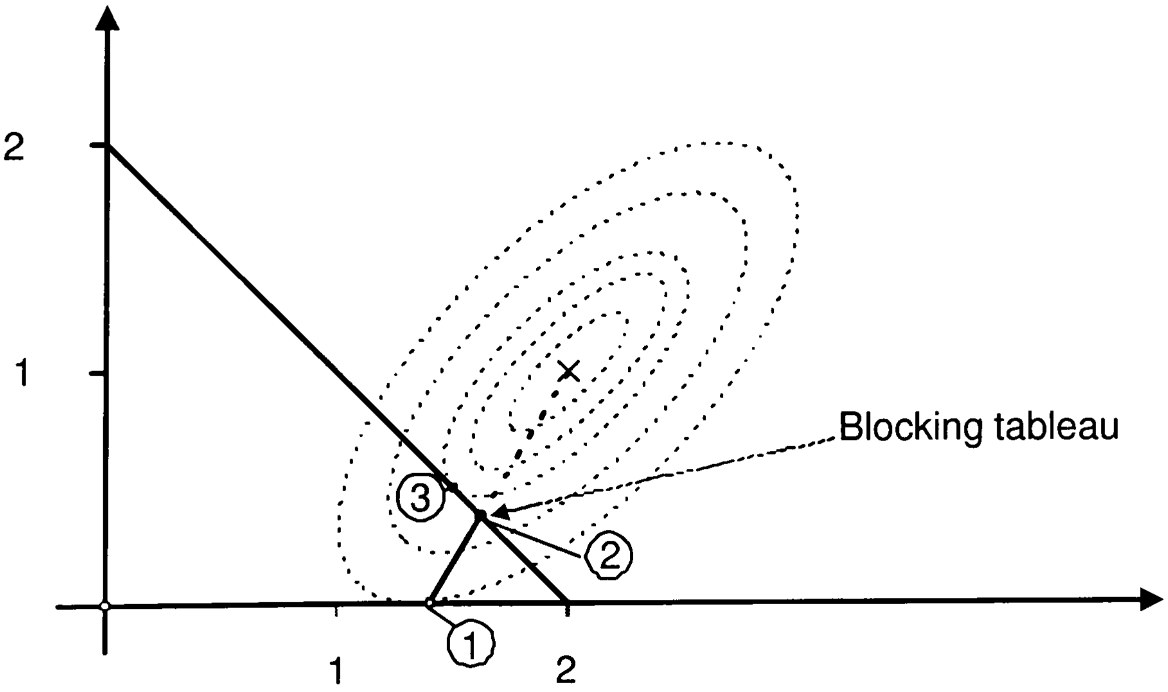

Figure 4.1: Simplex steps in the tableau

The standard tableau form of the problem is

	$-x_0$	$-x_1 \downarrow$	$-x_2$	$-v_1$
u_0	0	-6	0	-2
$u_1 \uparrow$	-6	-4	2	-1
u_2	0	2	-4	-1
y_1	2	1	1	0

Table 4.1: Tableau 1 (standard)

From the tableau above, $u_1 = -6$ and according to rule 1 x_1 enters the basis which leads to the next tableau.

	$-x_0$	$-u_1$	$-x_2 \downarrow$	$-v_1$
u_0	9	$-\frac{3}{2}$	-3	$-\frac{1}{2}$
x_1	$\frac{3}{2}$	$-\frac{1}{4}$	$-\frac{1}{2}$	$\frac{1}{4}$
u_2	-3	$\frac{1}{2}$	-3	$-\frac{3}{2}$
$y_1 \uparrow$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{3}{2}$	$-\frac{1}{4}$

Table 4.2: Tableau 2 (standard)

Tableau 4.2 is in standard form and the next variable entering the basis is x_2

	$-x_0$	$-u_1$	$-y_1$	$-v_1$ ↓
u_0	10	-1	2	-1
x_1	$\frac{5}{3}$	$-\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{6}$
u_2 ↑	-2	1	2	-2
x_2	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{2}{3}$	$-\frac{1}{6}$

Table 4.3: Tableau 3 (non-standard)

	$-x_0$	$-u_1$	$-y_1$	$-u_2$
u_0	11	$-\frac{3}{2}$	1	$-\frac{1}{2}$
x_1	$\frac{3}{2}$	$-\frac{1}{12}$	$\frac{1}{2}$	$\frac{1}{12}$
v_1	1	$-\frac{1}{2}$	-1	$-\frac{1}{2}$
x_2	$\frac{1}{2}$	$\frac{1}{12}$	$\frac{1}{2}$	$-\frac{1}{12}$

Table 4.4: Tableau 4 (standard and optimal)

The last tableau is optimal.

4.4 Sparse simplex (SSX) for QP

Sparse simplex (SSX) refers to the representation of the simplex method exploiting the sparsity of the data structure. In the book by Beasley [6], Maros and Mitra [70] summarised the computational development of the sparse simplex method applied to LP chronologically in the following way

Year	Topic	Main researchers
1947	Tableau simplex method	Dantzig
1953/54	Revised simplex method	Dantzig Orchard-Hays Wolfe
1954	Simple upper bound algorithm	Dantzig
1967	Generalised upper bound algorithm	Dantzig, Van Slyke
1954 1971 1971/72	Basis factorization and the elimination form of the inverse (EFI)	Markowitz Beale Hellerman, Rarick
1969 1972 1976	Sparse update procedures	Bartels, Golub Forrest, Tomlin Reid
1975	Presolve procedures	Brearley Mitra, Williams
1965 1986	Composite Phase-I procedures	Wolfe Maros

see next page

Year	Topic	Main researchers
1973	Combined price and pivot choice DEVEX procedure	Harris
1977 1992	Steepest edge pricing	Goldfarb, Reid Forrest, Goldfarb

Table 4.5: Chronological summary of the simplex development

For LP, the sparse simplex is introduced for problems with a sparse constraint matrix and for QP if the constraint matrix A and quadratic matrix Q of the objective are sparse.

The SSX is based on the revised simplex method (RSM). The standard form of a linear programming (LP) problem is given by the primal dual pair.

Primal linear programming (PLP) problem:

$$\begin{aligned} &\max c^T x \\ \text{s.t. } &Ax = b, \\ &x \geq 0, \end{aligned} \tag{4.17}$$

Dual linear programming (DLP) problem:

$$\begin{aligned} &\min b^T v \\ \text{s.t. } &A^T v \leq c. \end{aligned} \tag{4.18}$$

where $c, x \in \mathbb{R}^n$; $b, v \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m \times n}$.

In the LP case, the following steps are taken to perform SSX

1. *Initialise*

   A starting basis for the primal-dual problem is determined either by the unit matrix in A or an advanced basis.

2. *Factorise*

   The inverse of the basis $B^{-1} = U^{-1}L^{-1}$ is built and then a number of revised simplex steps are performed.

3. *Form price vector*

   In phase I the solution is infeasible or in phase II the solution is feasible. The price form vector is computed depending on the phase.

4. *BTRAN*

   The backward transformation (BTRAN) is used to calculate the simplex multipliers.

5. *Price*

   The reduced cost of the non-basic variable is observed. If a variable improves the reduced cost, then it is a candidate to enter the basis. In case of no improving variable the optimal solution is reached or there is no feasible solution.

6. *FTRAN*

   The forward transformation (FTRAN) gives the sequence in which the elementary elimination matrices compute the current coordinates of the current basis.

7. *Choose pivot*

   A pivot row is chosen to leave the basis. For phase I and II different logical steps are performed.

8. *Update*

   If there was a change in the basis, an update is determined to the basis inverse. Only the solution is updated if there was no change in the basis. Step 3 or 2 are the subsequent steps depending on the need for refactorisation.

The SSX for LP and QP is very similar. If the Q matrix of the quadratic problem is set to zero, this leads to the LP primal-dual problem.

SSX for QP

The sparse simplex method for QP problems follows the identical steps as SSX for LP which is stated above. The sparse method for the solution of QP problems goes in this case back to the tableau procedure introduced by Van-de-Panne and Whinston [88] which is set out in the earlier section.
A fully complementary starting basis is selected in other words each basic variable has a non-basic variable counter part which guaranties that the starting basis is fully complementary. It has also the same skew-symmetric properties. The pivot choice for entering and leaving the basis is according to the rules presented by Van-de-Panne and Whinston and described in the last preceding section.

Throughout the major iterations one standard tableau goes over to another one. However, in the minor steps in which *one* non-basic basis exchange is processed, the tableau is non-standard.

4.5 Interior Point Method (IPM) for QP

In 1984, Karmarkar [47] introduced the interior point method for linear programming problems. Since then, various basic approaches have been introduced which are summarised in [60], [89].

To solve the convex QP problem, the logarithmic barrier Primal-Dual interior point algorithm is stated and then extended by applying the idea of predictor-corrector method which was proposed by Mehrotra for linear programming problems in [73].

The primal and dual form of problem QP1 can be stated in the following way:

Primal quadratic programming (PQP1) problem:

$$\begin{aligned} \min \quad & f_{PQP1}(x) = c^T x + \tfrac{1}{2} x^T Q x \\ \text{s.t.} \quad & Ax - y = b, \\ & x, y \geq 0, \end{aligned} \tag{4.19}$$

Dual quadratic programming (DQP1) problem:

$$\begin{aligned} \max \quad & f_{DQP1}(x) = b^T v - \tfrac{1}{2} x^T Q x \\ \text{s.t.} \quad & u - A^T v - Qx = c, \\ & v, u \geq 0, \end{aligned} \tag{4.20}$$

where $c, x, u \in \mathbb{R}^n; b, y, v \in \mathbb{R}^m; A \in \mathbb{R}^{m\times n}$ and $Q \in \mathbb{R}^{n\times n}$ is symmetric positive semi-definite.

Consider the primal and dual QP problem, 4.19, 4.20 respectively taking into consideration the properties and definition of 2.2. A nonnegative vector y for the surplus variables is introduced. The given primal problem is formulated into a primal barrier problem by subtracting a barrier term for each of the nonnegative variables of the PQP problem.

$$\begin{aligned} \min \quad & c^T x + \tfrac{1}{2}x^T Q x - \mu \textstyle\sum_j logx_j - \mu \sum_j logy_j \\ \text{s.t.} \quad & Ax - y = b, \end{aligned} \tag{4.21}$$

The Lagrangian is built and 4.21 becomes

$$L(x, y, v) = c^T x + \frac{1}{2}x^T Q x - \mu \sum_j logx_j - \mu \sum_j logy_j + v^T(b - Ax + y) \tag{4.22}$$

The KKT conditions for the barrier problem are obtained by differentiating the Lagrangian with respect to each of its variables and setting these derivatives to zero. Removing redundancies from the system of equations built from the derivatives results in:

$$Ax - y = b, \tag{4.23}$$

$$-Qx + A^T v + u = c \tag{4.24}$$

$$XUe = \mu e, \tag{4.25}$$

$$VYe = \mu e, \tag{4.26}$$

where X, Y, U, V are diagonal matrices whose diagonal entries are the components of x, y, u, v; respectively. In solving this system of equations, x, y, u, v is replaced by $x + \Delta x, y + \Delta y, u + \Delta u, v + \Delta v$ and the following system is formed:

$$A\Delta x - \Delta y = b - Ax + y, \tag{4.27}$$

$$-Q\Delta x + A^T\Delta v + \Delta u = c + Qx - A^T v - u \tag{4.28}$$

$$U\Delta x + X\Delta u + \Delta U\Delta Xe = \mu e - XUe, \tag{4.29}$$

$$V\Delta y + Y\Delta v + \Delta V\Delta Ye = \mu e - VYe, \tag{4.30}$$

The variables $\Delta u, \Delta y$ are replaced by

$$\Delta u = X^{-1}(\mu e - XUe - U\Delta x), \tag{4.31}$$

$$\Delta y = V^{-1}(\mu e - VYe - Y\Delta v), \tag{4.32}$$

and the following system can be defined in matrix notation:

$$\begin{bmatrix} X^{-1}U + Q & A^T \\ A & V^{-1}Y \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta v \end{bmatrix} =$$

$$\begin{bmatrix} c - A^T v + \mu X^{-1}e + Qx \\ b - Ax - \mu V^{-1}e \end{bmatrix} \tag{4.33}$$

The path following method for QP problems solves this reduced system and performs the following principal steps:

```
**Set initial point** $(x, y, u, v) > 0$
**If not optimal**
{
    Build the right hand side;
    Calculate $\gamma = u^T x + v^T y$;
    Calculate the barrier parameter $\mu = \delta \frac{\gamma}{n+m}$;
    Solve
    $\begin{bmatrix} X^{-1}U + Q & A^T \\ A & V^{-1}Y \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta v \end{bmatrix} = \begin{bmatrix} c - A^T v + \mu X^{-1} e + Qx \\ b - Ax - \mu V^{-1} e \end{bmatrix};$
    Calculate $\Delta u, \Delta y$;
    Calculate step length $\Theta$:
    $\Theta = r \left( \max{}_{ij} \left\{ -\frac{\Delta x_j}{x_j}, -\frac{\Delta y_j}{y_j}, -\frac{\Delta v_j}{v_j}, -\frac{\Delta u_j}{u_j} \right\} \right)^{-1} \wedge 1;$
    $x \longleftarrow x + \Theta \Delta x$;    $v \longleftarrow v + \Theta \Delta v$;
    $y \longleftarrow y + \Theta \Delta y$;    $u \longleftarrow u + \Theta \Delta u$;
}
```

The stopping criteria of the interior point method is determined by a small duality gap which is linked with the complementarity. For a small γ which translates into a small duality gap, the computation is stopped. The reader is referred to Maros [68], Vanderbei [90].

The interior point method is in fact the repeated application of the Newton step solving a nonlinear system and gives the solution of a symmetric system of linear equations. The iterative process solves either a 'normal equations system' or an 'augmented system'. In the normal equation approach, the Newton equation system is processed by pivoting and Cholesky factorisation. In the augmented system approach, the analysis and the factorisation of the symmetric indefinite system is carried out dynamically. In other words, the pivots are chosen taking into consideration both, the sparsity and stability of the triangular factors. The number of iterations is nearly invariant of the model size. In normal circumstances, there are no more than 100 iterations. As in the simplex method, the sparse representation is an issue and determines which of the two forms is appropriate.

4.6 Computational study: IPM versus SSX

For the computation of the results for IPM as well as SSX, the following hardware and software was used:

CPU: Pentium III, 700 MHz,

Memory: 128 MB RAM,

Compiler: Digital Fortran, MS C,

Solver: FortMP.

The next table shows the comparison between IPM and SSX using the problems described in table 4.6.

		IPM			SSX	
	Number of IPM iterations	Value of objective function	Time [secs]	Number of SSX iterations	Value of objective function	Time [secs]
25fv47	36	1.3744E+07	34.16	13776	1.3744448E+07	246.64
e226	16	2.1265E+02	0.74	922	2.1265343E+02	0.84
fffff80	25	8.7315E+05	2.77	919	8.7314746E+05	1.67
scfxm2	39	2.7776E+07	1.56	1640	3.0816354E+07	4.28
scfxm3	91	3.0816E+07	4.32	87	1.8805096E+03	0.32
scsd6	11	5.0808E+01	0.85	810	5.0808214E+01	1.14

see next page

		IPM			SSX	
	Number of IPM iterations	Value of objective function	Time [secs]	Number of SSX iterations	Value of objective function	Time [secs]
scsd8	10	9.4076E+02	1.72	1726	9.4076357E+02	4.71
shell	35	1.5726E+12	1.79	722	1.5726368E+12	7.2
sierra	19	2.3750E+07	1.31	2576	2.3751139E+07	4.2
stair	30	7.9855E+06	1.92	544	7.9854528E+06	0.91
standat	15	6.4118E+03	0.68	384	6.4118384E+03	0.35

Table 4.6: Computational results: IPM versus SSX

Table 4.6 shows that the interior point method performs overall faster than the SSX method. The fast convergence and good computation time makes the interior point method a desirable algorithm for methods that need to solve continuous problems. The solution produced by IPM is optimal but not a basic solution where for SSX it is a basic optimal solution. Thus there is reasonable cause to study the cross-over from the interior point solution to a basic solution using the simplex method. The implementation and a computational study is performed in the next section.

4.7 IPM to SSX crossover (XO)

The advances applied to linear programming methods made it competitive and attractive to practitioners in respect to computational time and problem size. As a consequence of the increasing demand for solutions to QP problems resulting from portfolio optimisation, regression analysis, etc., researchers within the OR community tried to enhance the QP solvers profiting from the LP ideas such as advanced basis, cross-over, basis recovery.

Solution algorithms such as conjugate gradient, cutting plane methods, KKT based methods are known. However, the quadratic programming problem is mostly solved using the interior point method or sparse simplex method. In the case of the solution of a family of QP problems e.g. such as computing the efficient frontier, methods based on advanced basis restart give a high speed up. The solution of the interior point method however gives only an optimal solution which is usually not a basic solution. Applying a cross-over procedure gives an optimal basic solution which can be used in the framework for the solution of a family of QP problems.

Levkovitz et.al. [56] performed a computational investigation of the combination between IPM and SSX for LP solvers. A basis recovery technique is used to obtain a restart basis from the interior point method in order to cross over to the sparse simplex solver. As in linear programming, the performance of IPM for QP problem is much better for medium to large sized problems. However, the solution of IPM may not be basic. In order to use this solution for further computations, typically sensitivity analysis, restart, for the repeated use (efficient frontier) in general mixed integer programming , etc. it is necessary to recover the basis and "crossing over" to a pivotal algorithm. Based on the earlier study of IPM versus SSX for QP problems and the desirable speed ups achieved with the LP cross-over, a technique for the cross-over from an interior solution to a basic solution is theoretically investigated and then implemented in the $FortMP$ solver. In this thesis, the IPM solution is used to identify the restart basis which can be used as an advanced restart basis for SSX.

In the LP case, Megiddo [72] introduced a strongly polynomial algorithm. Berkelaar, Jansen, Roos, Terlaky extended these ideas for QPs and LCPs and presented theoretically a strongly polynomial basis identification algorithm for QP and LCPs for sufficient matrices in [9].

The set of optimal solutions of PQP1 (4.19) and DQP1 (4.20) can be characterised by maximal complementary solutions and the corresponding tripartition. The tripartition is defined by the set $\{B,\ N,\ T\}$

- $B := \{i : x_i > 0$ in at least one solution $x\}$,
- $N := \{i : u_i > 0$ in at least one solution $u\}$,
- $T := \{1, ..., n\} \backslash (B \cup N)$.

and is denoted by $\pi := \{B,\ N,\ T\}$. A maximally complementary solution (x, u) is a solution for which

$$x_i > 0 \Leftrightarrow i \in B; u_i > 0 \Leftrightarrow i \in N$$

the existence of such a solution is a consequence of the convexity of the optimal sets of PQP and DQP. McLinden [71], Güler and Ye [36] showed that such a maximal complementary solution is generated by the interior point method.

The basis identification (BI) algorithm starts from a complementary solution and has two general phases. The first phase consists of diagonal and exchange pivoting, the basic variables with value zero are replaced by the nonzero nonbasic variables. In the second phase, the set B is reduced iteratively by using the orthogonality property of the tableau, and if necessary some principal pivots are performed. This continues until B is empty.

The implementation of the cross-over from an interior point solution to sparse simplex method is employed. A set of initial variables σ_i, ξ_i are given by IPM. Additionally, it is assumed that the initial vectors ξ, σ for the given problem are complementary with corresponding tri-partition $\pi = \{B, N, T\}$. Given the matrix $M = [0, A; -A^T, -Q]$ with the elements m_{ij} and the initial values $\xi^0 = \xi, \sigma^0 = \sigma$ with the initial tri-partition $\pi^0 = \{B^0, N^0, T^0\}$ then the following principal steps performed for the crossover:

```
Major cycle:
for $B^j \neq 0$ and $\sigma^j > 0$
{
Choose$\{k \in B^j\}$
if $m_{kj} \neq 0 \wedge l \in T^j \cup B^j$
    if $k = l \wedge m_{kl} < 0$
        Diagonal pivot:
        $\sigma_k^j$ leaves the basis and $\xi_k^j$ enters the basis
        $\Longrightarrow B^j$reduces by one
    else
        $k \wedge m_{kk} = 0 \wedge m_{kl} \neq 0$
        Off diagonal pivot
        exchange pivot on pair$(k, l)$
        $\sigma_k^j, \sigma_l^j$leave the basis
        $\xi_k^j, \xi_l^j$ enter the basis
        $\Longrightarrow B^j$reduces by one
else
    Minor cycle:
    for $B^j \neq 0$
    {
        1. Choose arbitrary $k \in B^j$ corresponding to a non-basic $\xi_k^j$
        $J_B$ is the set of basic indices and $J_N$ the set of nonbasic variables
        $t_{(k)}$ is a vector which is the column of the corresponding tableau
        with $\tau_{ki}$ the coefficients of $B$
        2. Get corresponding $t_{(k)_i}$
        $t_{(k)_i} = \begin{cases} \tau_{ki} & \text{if} & i \in J_B \\ -1 & \text{if} & i = K \\ 0 & \text{if} & \textit{neither} \end{cases}$
        3. Eliminate one of the positive values in$\xi_{B^j}$ or$\sigma_{N^j}$
        Since the one element of $t_k$ is nonzero, the corresponding element
        in $(\xi_j, \sigma_j)$ is positive. The positive $\xi_{B^j}$ are eliminated by
        $((\xi^j, \sigma^j) + \vartheta t_{(k)})$. $\vartheta$ is an appropriate scalar obtained by a ratio test
        if $B^j$ reduces by one index and $N^j$ remains the same
          => Then update $B^j, N^j, T^j, \xi, \sigma$, $j \rightarrow j + 1$
         and start the minor cycle again
        else
         $N^j$ reduces by either one or more indices
          =>Leave minor cycle and start major cycle again
        $j \rightarrow j + 1$ and start the minor cycle again
    }
$j \rightarrow j + 1$
Update $B^j, N^j, T^j, \xi, \sigma$
}
```

The presented cross-over technique is implemented within the *FortMP* framework and the following results were obtained using the same platform as before:

	Number of IPM iterations	Time [secs]	Value of objective function	Number of XO iterations	Total time [secs]
25fv47	36	34.16	1.3744E+07	12423	226.98
e226	16	0.74	2.1265E+02	1068	1.29
ffff80	25	2.77	8.7315E+05	1522	4.34
scfxm2	39	1.56	2.7776E+07	1444	3.26
scfxm3	91	4.32	3.0816E+07	2154	8.31
scsd6	11	0.85	5.0808E+01	886	1.71
scsd8	10	1.72	9.4076E+02	2018	6.39

see next page

	Number of IPM iterations	Time [secs]	Value of objective function	Number of XO iterations	Total time [secs]
shell	35	1.79	1.5726E+12	1127	7.82
sierra	19	1.31	2.3750E+07	1457	3.56
stair	30	1.92	7.9855E+06	533	2.42
standat	15	0.68	6.4118E+03	108	0.77

Table 4.7: Computational results of IPM to SSX cross-over

From the table above, it can be seen that there are hardly any improved solution times. The enthusiasm about improving the solution times by implementing a cross-over is not proven. Implementing more advanced techniques for crashing an advanced basis for SSX can be done but due to the very good performance of SSX it is doubtful that for any problem a crash routine would be useful. This does not imply that a desirable improvement with an advanced basis would be unsuccessful for all classes of problems. Therefore, further research can be done.

4.8 Computational study for the continuous portfolio selection problems

The following computational study shows the results using first sparse simplex, secondly the interior point method and lastly the cross-over to solve the factor models using the problems stated in table 3.2.

NAME	Iterations	QPSSX time	Total time
QPF50	64	0.00	0.00
QPF100	117	0.00	0.01
QPF500	528	0.30	0.34
QPF1000	1016	1.17	1.20
QPF4500	4102	45.02	45.14

Table 4.8: Computational results for factor models using SSX

NAME	Iterations	QPIPM time	Total time
QPF50	10	0.00	0.02
QPF100	10	0.00	0.01
QPF500	12	0.11	0.11
QPF1000	16	0.42	0.42
QPF4500	26	9.33	9.36

Table 4.9: Computational results for factor models using IPM

NAME	Total number of	IPM	QP XO	Total
	iterations(IPM)	time	time	time
QPF50	55(9)	0.03	0.05	0.06
QPF100	55(9)	0.02	0.05	0.05
QPF500	53(11)	0.11	0.36	0.44
QPF1000	172(15)	0.44	2.23	2.78
QPF4500	4180(25)	9.64	19.63	63.42

Table 4.10: Computational results for factor models using XO

The tables 4.8, 4.9 and 4.10 show the computational results solving the QPF problems with SSX, IPM or XO. The results suggest that IPM is the best algorithm for large problems whereas SSX is better for smaller problems. It is for the user to decide which solver to apply, IPM or SSX depending on the size of the problem.

As mentioned earlier, there are two portfolio models considered for the computation. Due to limitations of the computer capacity as well as limits on the computational time frame, the QPMV4500 is not computed.

The following tables describe the results for QPMV problems using SSX, IPM and XO.

NAME	Iterations	QPSSX time	Total time
QPMV50	68	0.02	0.02
QPMV100	121	0.05	0.08
QPMV500	525	10.91	11.70
QPMV1000	1012	163.16	167.22
QPMV4500	-	-	-

Table 4.11: Computational results for MV models using SSX

	Iterations	QPIPM time	Total time
QPMV50	9	0.03	0.03
QPMV100	8	0.05	0.06
QPMV500	8	7.67	7.95
QPMV1000	14	232.92	234.22
QPMV4500	-	-	-

Table 4.12: Computational results for MV models using IPM

NAME	Total number of iterations(IPM)	IPM time	QP XO time	Total time
QPMV50	40(8)	0.02	0.03	0.03
QPMV100	58(7)	0.06	0.09	0.13
QPMV500	109(7)	8.05	14.09	22.66
QPMV1000	286(13)	238.13	317.69	425.59
QPMV4500	-	-	-	-

Table 4.13: Computational results for MV models using XO

From the above results for the MV models, it can be seen that the computation of the mean variance model follows a similar behaviour as the factor model. However, the dense quadratic covariance matrix is computationally challenging for both IPM and SSX and in respect of computation time and size, it is for neither of the algorithms desirable to compute the MV problem with a universe of 4500 stocks.

The main difference between the factor problems and the MV problems is the speed up property of the factor model. Therefore, the factorisation does not only pay off from the analysis point of view it also produces results much quicker. From a practical point of view, a fast computation is eligible particularly for back-testing, sensitivity analysis.

Chapter 5

Solution methods for QMIP problems

The classic portfolio selection model is expressed as a quadratic programming problem in continuous variables. Including realistic restrictions which involve discrete decisions expands the QP problem to a quadratic mixed integer programming problem.

The computational aspects of integer (discrete) linear programming programming are challenging and their solution algorithms have been studied extensively. Quadratic mixed integer programming (QMIP) are natural extensions of the mixed integer programming (MIP) problems and show some common algorithm structures. Section 5.1 gives an overview about existing methods for solving discrete linear programming. In this thesis, the main focus is on problems with quadratic objectives. The solution methods for QMIP problems which are based on MIP approaches are discussed in section 5.2. In section 5.3 a branch and bound method for quadratic mixed integer programming problems is described. In the subsequent section 5.4, a customised approach for the discrete portfolio planning problem is developed and investigated which can process large models efficiently. A detailed computational study using the discrete portfolio model collection is given in section 5.5

5.1 Mixed integer programming: An analysis of algorithms

The success of the simplex method for linear programming problems and its adoption within solution methods for mixed integer programming problems are closely related. In the early research of solving mixed integer programming problems the focus lay in finding a general method to solve any mixed integer programming problem to optimality. Later heuristic methods have been introduced which are able to find quickly discrete, feasible but often suboptimal solutions; in general heuristic methods do not adopt well to prove optimality.

Exact methods

The term exact method refers to an algorithm which is guaranteed to solve a given problem to optimality. Exact methods include cutting plane techniques, enumerative techniques and partitioning algorithms.

In the cutting plane technique, a current and integer infeasible solution is cut off by adding constraints. The first to introduce this method were Dantzig, Fulkerson and Johnson [20]. The latter method was for the special case of the travelling salesman problem. In 1963, Gomory [32] developed a formal theory of cutting planes and described a convergent algorithm which processed general integer programming problems. Because of (a) slow convergence and (b) no solution is available until the final step, the cutting plane method was initially unpopular. Recently, Mitchell [75], Balas et al. [2] extended the cutting plane theory and developed an advanced cutting plane method.

An alternative to the cutting plane method is branch and bound which is an implicit enumeration technique introduced by Land and Doig [51]. The branch and bound method uses the solution space property of an IP which is a finite number of possible integer feasible solutions. All integer feasible points of the IP may be potentially enumerated. Obviously searching all the feasible points is computationally expensive. Therefore, only a fraction of the feasible points are enumerated which, for a bounded problem, is called implicit enumeration e.g. branch and bound.

Branch and bound follows the idea of divide and conquer. In other words, the solution space is partitioned and smaller sub-problems are solved and then the solutions are combined. Little et.al. [57], Balinski [3], Beale [5], Mitten [78], Mitra [76] are among the first researchers to introduce the branch and bound (B&B) method.

The partitioning method was proposed by Benders [8] for mixed integer programming methods. The method divides the problem into a (continuous) linear problem and a pure integer problem. Applying the duality theory, the partitioning algorithm solves a LP, calculates an extreme point. This gives an lower bound for the mixed integer programming problem and solving the pure integer programming problem gives an upper bound. The optimal solution is found when the two bounds coincide.

In the last 20 years, hybrids of these methods have become more and more important. One of the most promising approaches is the combination of the cutting plane with the branch and bound algorithm for instance see Crowder and Padberg [19]. This approach has the benefit of cutting plane yet the branch and bound framework makes it possible to obtain feasible integer solutions at an intermediate stage of the search process.

5.2 Quadratic mixed integer programming solution algorithms

The solution methods for QMIP problems with discrete and continuous variables are essentially extensions of MIP solution algorithms, since the solution space has the same structure only the objective function is quadratic.

The quadratic mixed integer programming problem without distinguishing between the various types of integer variables the problem can be stated as

$$\begin{array}{llll} \min & \sum\limits_{j=1}^{n} c_j x_j + \frac{1}{2} \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} q_{ij} x_i x_j & & \\ \text{s.t.} & \sum\limits_{j=1}^{n} a_{ij} x_j = b_i & i = 1, ..., m & (5.1) \\ & x_j \geq 0, & j \in N \text{ and integer for} j \in N' ; N' \subset N. & \end{array}$$

In general, outer approximation method, Benders decomposition and the branch and bound method are proposed and investigated as exact approaches for the solution of QMIP problems. Additionally, a few heuristic methods have been proposed.

The Branch&Bound method

Initially, Land and Doig [51] proposed the first branch&bound (B&B) routine and later Lawler and Wood [52] gave a survey of branch and bound methods. The B&B approach is to divide the feasible region, to solve over the smaller spaces and then to put the results together. Garfinkel and Nemhauser [28] argue that B&B does not require linearity of the objective function. Gupta and Ravindran [37] implemented for mixed integer non-linear programming problem a B&B routine. By observing the behaviour for the implemented algorithm, they discovered that the solution time increases linearly with the number of integer variables as well as the number of constraints in the model. In 1985 a branching heuristic was proposed by Körner [49] aiming to choose the branching variables in an order which minimises the size of the tree. A LP/QP based branch and bound method is reported in 1972 by Quesada and Grossmann [82].

Bienstock [11] presents in 1996 a branch and cut algorithm for QMIP problems with an upper bound on the number of positive variables. The given branch and cut algorithm computes one round of cutting at each node and using the 'node' strategy to choose the next node to branch on. From all variables not yet branched on, the one furthest from its bounds is chosen as the next branching variable.

Benders decomposition

A further important exact method is the Benders decomposition [8]. The original minimisation problem is decomposed into a minimisation over the so called complicating variable and a minimisation over the decision variable of an LP problem parameterised in the complicating variables. The LP is dualised and corresponding master problem is obtained. The derived problems are solved using relaxation techniques. In 1972, Geoffrion [29] generalised the Benders decomposition technique for convex functions.

Lazimy [53] exploits the ideas of the Benders decomposition for a class of mixed integer quadratic programming problem. In his approach, assumptions about the rank of the Hessian matrix are made with the consequence that the MIP problems are not included. It is shown by Flippo and Rinnoy Kan [26] that the Lazimy's method is based on inaccurate interpretations of the Benders Decomposition. They then give the correct Benders decomposition for QMIP problems.

Outer approximation

The outer approximation approach proposed first by Duran and Grossmann [23] is similar to the Benders decomposition. The main difference is that instead of searching for the dual representation of the NLP, first order necessary conditions are employed. The obtained optimal solution for a sub-problem provides a point for the generation of the supporting hyper-planes. Fletcher and Leyffer present in [25] an outer approximation routine for mixed integer non-linear programming problems and show results. Difficulties arising due to the non-convexity are overcome using heuristics.

5.3 A Branch&Bound framework for QMIP problems

The B&B method for QMIP problems is a tree search approach. The tree is developed by branching on discrete variables thus partitioning the solution space and bounding the objective function value. Since most of the linear MIP and QMIP problems are such that their integer variables are restricted to take binary values, the description of the B&B method is limited to QMIP problems with binary variables δ_{i_k}.

Every tree has a root node P_0 which gives the entire state space $S = S_0$ and subsequent nodes are referred to as sub-problems P_j and represent smaller partitions $S_j \subset S$. Therefore, every node corresponds to a sub-problem which is linked to its parent sub-problem. Any parent problem P_i is associated with two subproblems P_j and P_{j+1}. These nodes and their associated sub-problems (continuous quadratic programming relaxation) (P_j ,P_{j+1}) are created by setting a variable δ_{i_k} currently taking fractional a value to its lower and upper bounds (0 and 1) respectively.

The branching process can be viewed in the following way:

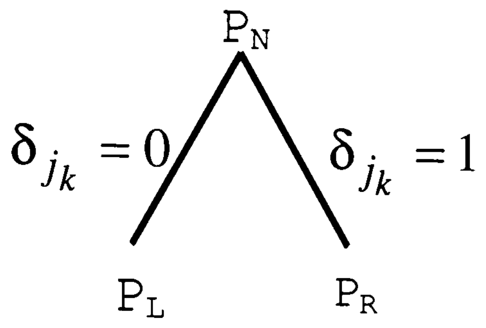

Figure 5.1: Branching process in a QMIP problem

where k is the tree depth, P_N the current problem, P_L the left branch (P_N plus $\delta_{jk} = 0$) and P_R the right branch (P_N plus $\delta_{jk} = 1$). Therefore P_j and P_{j+1} are the finite search spaces of $S_j = \{S_{k_j}\} \cap \{\delta_{k_j} = 0\}$ and $S_{j+1} = \{S_{k_j}\} \cap \{\delta_{k_j} = 1\}$. At any level of the tree, all the unsolved nodes are S. It ensures that the optimal solution over S is equal to the optimal solution in one of the smaller sub-spaces S_i. The optimal solution over S is as well the minimum of all the sub-sets values. The solution process of a node requires usually solving the quadratic programming relaxation dropping all the integrality constraints. If no feasible solution at a node occurs, then the node investigation is terminated. While if the solution of the continuous quadratic programming relaxation of a sub-problem is integer, the value of the objective function of this sub-problem is set as an upper bound for all the remaining sub-problems or is found to be sub-optimal. Furthermore, the optimal solution to the continuous quadratic programming relaxation gives a lower bound on the objective function value of the subsequent sub-problems. The sub-problems with a lower bound on the objective function exceeding the value of the remaining node's objective value are excluded from the branching process. This procedure is pursued until the best integer feasible solution amongst the eligible nodes in the search tree is found.

In the whole process of branch and bound, there are two important choices to be made. Firstly, the choice of the next node which is the next sub-problem to be solved. Secondly, the choice of the fractional variables which are used for partitioning.

An explicit statement of the $B\&B$ algorithm can be found in Nemhauser and Wolsey [79], Bertsekas [10], Nwana [80]. The solution process for branch and bound has six principal steps. Let L denote the list of active subproblems $\{IP^i\}$, where $IP^0 = IP$ denotes the original integer program corresponding to the state S_0. z_i denotes a lower bound on the optimal solution of the value of IP_i (usually the optimal objective value of the relaxation of subproblem i), and z_{IP} denotes the incumbent objective value (representing the current best integer feasible solution). The $B\&B$ is stated below:

1. Initialisation

   Set $L = \emptyset$, $z_0 = -\infty$, $z_{IP} = +\infty$

2. Termination test

   If the list of the uninvestigated nodes is empty, the last stored node is optimal.

3. Node(Subproblem) Selection

   Select and delete active subproblem IP^i from L

   Solve the continuous quadratic programming relaxation of IP^i to obtain the optimal objective value z_i for the optimal solution, x^i (if it exists)

4. Bounding

   If $z_i \geq z_{IP}$ goto step 2

   If $z_i < z_{IP}$ and the solution x_i is integer feasible, then update $z_{IP} = z_i$ and delete from L all problems, IP^j with $z^j \geq z_{IP}$. Go to Step 2

5. Branching (Partitioning)

   Divide the finite search space S_i into smaller search spaces S_j and S_{j+1} such that $S_j \cap S_{j+1} = S_i$. Add the problems IP_j and IP_{j+1} to L. Go to Step 2

6. Exit and report of solution

   If $z_{IP} = +1$ then no feasible solution else Optimum integer solution is z_{IP} end if

The described procedure of the branch and bound algorithm can be extended by more advanced node and variable choice strategies (Nwana [80]).

Analysis of the algorithm: Dual and its importance

An important aspect of the success of the *B&B* algorithm is that by applying basis restart and the dual method a family of models are rapidly investigated. For instance: P_1, P_2, P_3, P_4 are all child problems of P_0 and can be efficiently processed by applying the basis restart from the parent. Further advantage is obtained by applying the dual simplex since every sub-problem after branching is primal infeasible but dual feasible. The importance of this feature is illustrated by the summary information in respect of *MIPLIB* models given by Maros [70], Nwana [80].

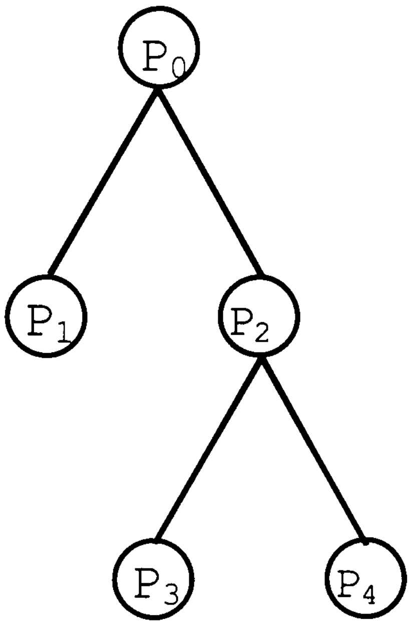

Figure 5.2: Family of a QMIP problem nodes

5.4 Heuristics and investigation of B&B search for QMIP problems

The branch and bound procedure as such performs well in finding an initial (good) solution. For relatively large models a heuristic refinement of the *B&B* search procedure was investigated to get a solution for the industrial portfolio optimisation problem stated in section 3.4. This heuristic is only applicable to the given class of portfolio model.

A three step procedure is followed to solve the problem in less computational time than solving the problem with a conventional solver. The three steps of the refined *B&B* heuristic are described below:

Step 1:

The discrete variables and the corresponding discrete constraints are relaxed (dropped) and the reduced continuous QP model is solved. This yields a limited set of non-zero portfolio weights and is referred to as the "QP" model.

Step 2:

A subset of choices is made by filtering out all portfolio weights less than a certain amount in the QP run. To this reduced model the related binary variables δ_i, threshold constraints and the cardinality constraints are added. A first integer solution is found by the procedure "Priority UP" which means in effect that δ_i's are selected and fixed to 1, one by one, until all have been fixed and the cardinality constraint is satisfied.In general, a loss of feasibility is encountered but since the investigated model is formulated as a goal programm feasibility is kept.

Step 3:

A more general approach is used, fixing δ_i's to zero as well as to 1 but the algorithm starts with the integer solution already known from step 2. Subproblems with optimum solution values above a bound are abandoned and continuing the solution process an improved integer solution is found.

Model Alternatives

The entire procedure is strongly affected by the number of investments chosen by filtering. A smaller filter leading to a larger selection requires longer to solve but the solution found may be better. The filter is chosen by analysing the achieved portfolio weights by solving the continuous portfolio selection problem and depending on the users preference, a filter is applied cutting off relatively small portfolio weights.

In the results tabulated below, three different filters are used and the six corresponding runs are given.

QMIP 1.1 and QMIP 1.2

step 2 and 3 with a filter small enough to cover 1705 investments selected by the QP run.

QMIP 2.1 and QMIP 2.2

steps 2 and 3 with a filter selecting 1267 investments

QMIP 3.1 and QMIP 3.2

steps 2 and 3 with a filter selecting 899 investments.

Computational Platform

Runs were carried out using the following hardware and software:

CPU: Pentium III, 700 MHz,

Memory: 128 MB RAM,

Compiler: Digital Fortran, MS C.

The general statistics of each problem solved are given below in table 5.1:

Model type	Number of variables	Number of constraints	Number of binaries	Number of non-zeros	Filter value
QP	19187	19177	-	90034	
QMIP 1.1	12077	10404	1705	48323	0.0000001
QMIP 1.2	12077	10404	1705	48323	0.0000001
QMIP 2.1	9011	7776	1267	36148	0.00012
QMIP 2.2	9011	7776	1267	36148	0.00012
QMIP 3.1	6435	5568	899	25889	0.00027
QMIP 3.2	6435	5568	899	25889	0.00027

Table 5.1: Model statistics of the discrete portfolio optimisation models

The filter value in table 5.1 is used to truncate every value below from the continuous solution. The lower the filter value is the more assets will be in the discrete portfolio optimisation model.

Model type	Value of objective function	Return relative to benchmark	Risk relative to benchmark	Penalty value	Number of selected stocks	Total time [sec]
QP	5.1825064	0.00086	0.00003	5.18334	1719	319.83
QMIP 1.1	5.1828622	0.00076	0.00028	5.18334	150	1681.88
QMIP 1.2						
1 integer	5.1828603	0.00083	0.00035	5.18334	142	3342.65
3 integer	5.1828514	0.00082	0.00033	5.18334	146	3485.40
QMIP 2.1	5.1828512	0.00077	0.00028	5.18334	150	783.18
QMIP 2.2						
1 integer	5.1828509	0.00081	0.00032	5.18334	127	1265.43
3 integer	5.1825432	0.00081	0.00031	5.18334	129	1362.71
QMIP 3.1	5.1828561	0.00073	0.00025	5.18334	150	382.29
QMIP 3.1						
1 integer	5.1828560	0.00080	0.00032	5.18334	126	530.08
3 integer	5.1828461	0.00081	0.00031	5.18334	128	585.34

Table 5.2: Results using QMIP models and breakdown of objective functions

The chosen, original problem has nearly 24000 variables of which 4571 are discrete. Attempting to solve this problem with branch and bound takes more than 10 hours. Therefore, the described heuristic is a necessary tool to obtain in a reasonable time frame a good solution. Table 5.2 shows the total time taken to solve the three models as well as the different values associated to risk (relative to benchmark), return (relative to benchmark) and penalties of the portfolio. The best risk and return level is achieved by the continuous model with its 1719 stocks in the portfolio. However, one requirement of the model is to have at most 150 stocks in the portfolio.

By choosing different filter values the stock universe is reduced to three levels. The comparison of the three models shows that by taking a small filter value good risk and return values are achieved within an hour. The larger filters values give the same risk and return though gaining less and taking more risks than the portfolio with 1719 stocks.

5.5 Computational study for the discrete portfolio selection model

In this section, two different strategies for solving QMIP problems are compared.

- Strategy 1: The tree is searched until the first integer feasible solution.
- Strategy 2: The tree is searched for 10 integer feasible solutions or 50000 nodes are evaluated.

In the next two tables, the two strategies are applied to the data set.

	Value of discrete obj. function	Number of iterations (nodes)	Branch and Bound time [secs]	Total time [secs]
QMIPFL50	1.2648329	138 (3)	0.03	0.05
QMIPFL150	0.2884729	404 (20)	0.28	0.34
QMIPFL300	0.1988575	846 (30)	1.11	1.34
QMIPFL600	0.1707010	1622 (30)	2.86	3.73
QMIPFL1200	0.1486922	4473 (40)	17.78	25.13
QMIPFL1800	0.1903924	7331 (40)	43.34	62.83
QMIPFL4500	0.2098827	15294 (40)	335.03	585.81
QMIPFR50	118.30733	133 (3)	0.03	0.05
QMIPFR150	32.913636	405 (20)	0.27	0.28
QMIPFR300	32.634988	696 (30)	1.08	0.34

see next page

	Value of discrete obj. function	Number of iterations (nodes)	Branch and Bound time [secs]	Total time [secs]
QMIPFR600	19.831407	1300 (30)	4.91	5.73
QMIPFR1200	18.945271	2533 (40)	16.41	19.47
QMIPFR1800	16.470851	4043 (40)	39.08	49.55
QMIPFR4500	13.268292	8518 (40)	304.83	440.30

Table 5.3: Computational results for discrete factor model using strategy 1

	Value of discrete obj. function	Number of iterations (nodes)	Branch and Bound time [secs]	Total time [secs]
QMIPFL50	0.68604190	24949 (11726)	33.44	33.45
QMIPFL150	0.24580903	603147 (50000)	377.08	377.16
QMIPFL300	0.17313613	1916305 (50000)	1192.69	1192.97
QMIPFL600	0.14288833	133026 (3254)	256.22	258.16
QMIPFL1200	0.090443132	4112809 (50000)	30548.01	30571.92
QMIPFR50	69.852319	22339 (10744)	36.03	36.06
QMIPFR150	28.436833	59569 (29178)	175.06	175.13

see next page

	Value of discrete obj. function	Number of iterations (nodes)	Branch and Bound time [secs]	Total time [secs]
QMIPFR300	20.519690	11548 (2896)	42.09	42.32
QMIPFR600	19.665274	131240 (50000)	1399.72	1400.19
QMIPFR1200	15.831303	1634129 (50000)	19297.97	19306.33

Table 5.4: Computational results for discrete factor model using strategy 2

In the second strategy, the tree search is terminated after 10 integer solutions are found. In the table above, the tree search for each of the problems was not always completed. It stopped after reaching 50000 nodes regardless of the number of feasible integer solutions. The tree of QMIPFL50 and QMIPFR50 was completely searched to optimality without finding 10 feasible integer solutions or exceeding 50000 nodes.

The two tables 5.3 and 5.4 compared to show that the objective value of strategy2 to strategy1 is improved by a factor of approximately 1.2 except for QMIPR50 and QMIPL50 where it is a factor of 1.77.

Chapter 6

Investigation of the continuous and discrete efficient frontiers

The efficient frontier represents that set of portfolios that has the maximum rate of return for every given level of risk, or the minimum risk for every level of return. Following the discussion of section 1.3, the efficient frontier is a Pareto optimal curve of the two objective functions maximising return and minimising risk.

In section 6.1, the background to the efficient frontier is given in respect of the initial definition as well as the underlying discrete or continuous portfolio optimisation problem. Section 6.2 shows different methods for the construction of the continuous efficient frontier. The discrete case of the efficient frontier is discussed in section 6.3.

6.1 Background

Markowitz defines an efficient asset or portfolio of assets to be efficient if no other asset or portfolio of assets offers higher expected return with the same or lower risk with the same or higher expected return. The set of all efficient portfolios are used to plot the efficient frontier. The slope of the efficient frontier curve decreases steadily as one moves upward. This implies that adding equal increments of risk moving up the efficient frontier results in diminishing increments of expected return.

Luenberger [59] describes the set of points of a portfolio as the feasible region or feasible set (see Figure 6.1) which is convex. If the investor allows short selling the feasible region is enlarged though the properties of a convex set do not change.

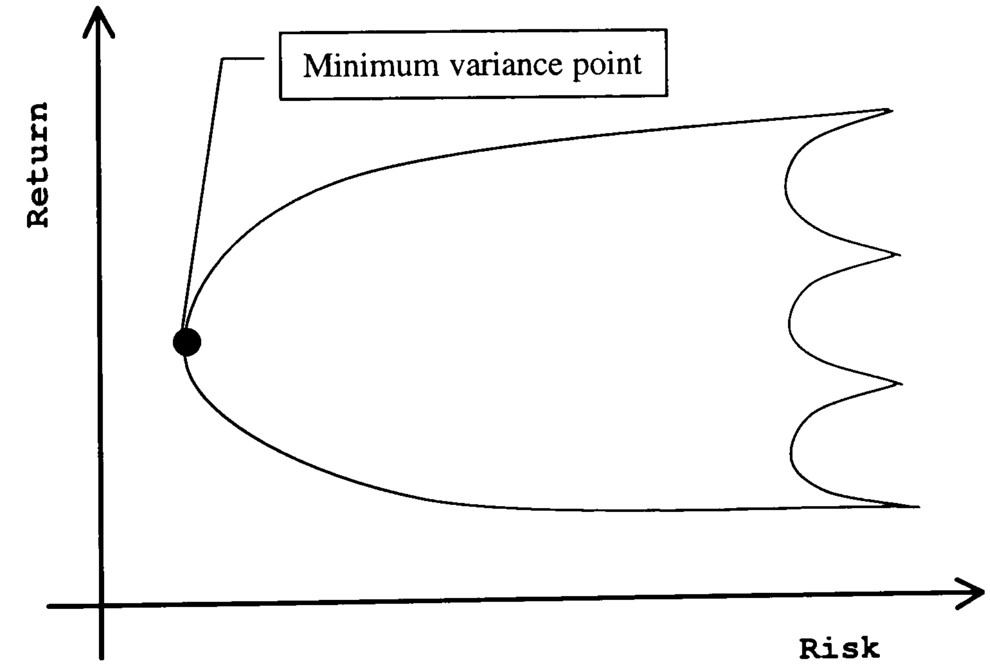

Figure 6.1: Feasible set of all efficient portfolios

The boundary of the feasible set is called the minimum variance set where the minimum variance point marks the turning point from risk averse to risk preferring investors. The upper part of the minimum variance set defines the efficient frontier of the feasible set giving the risk averse investor the best mean-variance portfolios. The efficient frontier marks the minimum variance portfolio on the left side and on the right side the maximum return portfolio.

6.2 The continuous efficient frontier (CEF)

Model statements

In this section of the chapter, the main focus lies in continuous portfolio optimisation and the computation of the continuous efficient frontier (CEF). In chapter 3, a mean-variance (MV) model as well as a factor (F) model are described and in the following study the following two basic models are pursued.

QPMV

$$\begin{aligned} \min \quad & \sum_{i=1}^{N}\sum_{j=1}^{N} \sigma_{ij} x_i x_j && (6.1)\\ \text{s.t.} \quad & \sum_{i=1}^{N} x_i = 1, && (6.2)\\ & \sum_{i=1}^{N} \mu_i x_i = \rho, && (6.3)\\ & x_i \geq 0, \qquad i = 1, ..., N && (6.4) \end{aligned}$$

where $x \in \mathbb{R}^N$ denotes the portfolio weights, $\mu \in \mathbb{R}^N$ the return, $V \in \mathbb{R}^{N\times N}$ the covariance matrix with elements σ_{ij} which is positive definite and ρ the desired level of return.

QPF

$$\min \quad \sum_{i=1}^{M}\sum_{j=1}^{M} \beta_i\beta_j\sigma_{f_{ij}}^2 + \sum_{i=1}^{N} x_i^2\sigma_{e_i}^2 \tag{6.5}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} x_i = 1, \tag{6.6}$$

$$\sum_{i=1}^{N} \mu_i x_i \geq \rho, \tag{6.7}$$

$$\beta_m = \sum_{i=1}^{N} x_i\beta_{im} \qquad m = 1, ..., M \tag{6.8}$$

$$x_i \geq 0, \qquad i = 1, ..., N \tag{6.9}$$

where M is the number of factors, β_{im} the sensitivity of asset i to factor m, β_m the portfolio's factor sensitivities, $\sigma_{f_m}^2$ the factor variances and $\sigma_{e_i}^2$ the random component of the specific variance. Throughout the whole chapter, these are the basic models which are extended later by introducing discrete variables.

Computational algorithm

The theory of the efficient frontier is a straight forward approach in which at every point of the EF a QP or QMIP problem is solved. Traditionally, the desired level of return is varied between its maximum and minimum value and each individual QP problem is solved. The solution is an efficient point and the set of all such points span the efficient frontier. This method is suitable for small problems but for larger problems it is too time consuming.

In order to speed up the procedure, warm-start techniques can be exploited. Originally, each of P_k points on the efficient frontier need to be computed. Obviously, the first few points $P_1, ..., P_{10}$ are computationally cheap since the portfolio optimisation problem is very close to an LP problem. In the case of "real" quadratic problems the computation time by itself is efficient but solving a family of QP problems makes it computationally expensive. The implemented approach solves the initial point P_1 and saves its basis. Then it is used as an initial solution for the subsequent point P_2. This procedure is performed throughout until solving for point P_k.

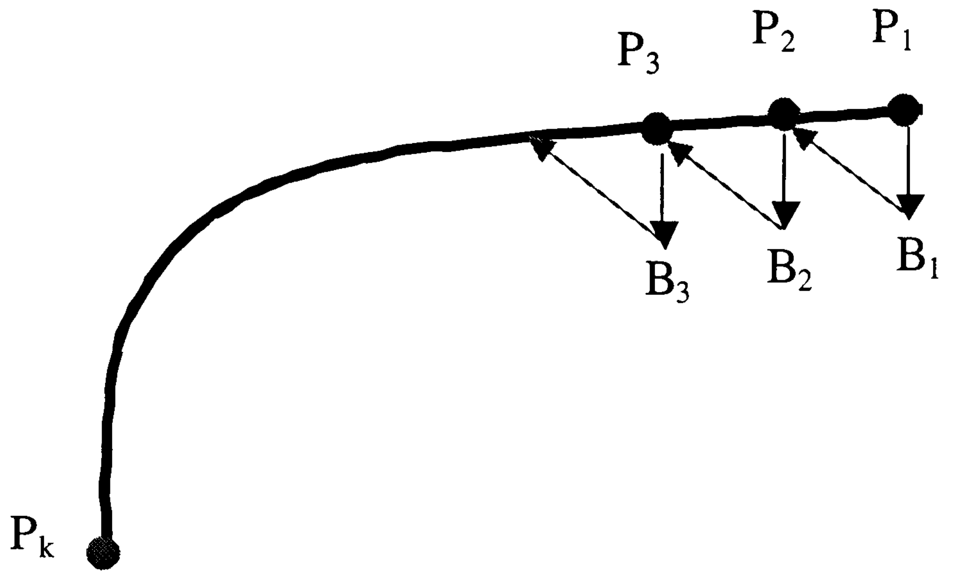

Figure 6.2: Restart techniques for CEF

The continuous efficient frontier (CEF)

The performed study is based on two different portfolio optimisation models. The factor model (QPF) minimises risk subject to a certain level of return and the risk matrix exploits the decomposition of V. As mentioned earlier, with this approach the quadratic terms are reduced and the constraint matrix becomes more sparse. The MV model (QPMV) is the standard Markowitz model.

The following table visualises the universe of stocks used to perform the study.

Number of stocks	10	100	500	1800	4500
QPMV Model	X	X	X	X	-
QPF Model	X	X	X	X	X

The computation of the continuous efficient frontier is done in various ways. Two of them are described in the previous section. The computational study focuses on SSX and its advances using restart. The following table shows a comparison of the different ways of computing the CEF with 100 points.

	CEF (SSX)		CEF (SSX-Restart)		CEF (IPM)	
	Number of	Total	Number of	Total	Number of	Total
NAME	iterations	time	iterations	time	iterations	time
QPF50	2756	1.15	53	0.98	1480	1.63
QPF100	5372	1.77	103	1.53	1361	2.29
QPF500	25486	15.67	519	9.20	1689	17.60
QPF1000	39809	52.05	1041	21.73	2012	63.33
QPF4500	169946	1961.29	5541	375.21	2525	1143.10

Table 6.1: Computation time of CEF for QPF models

From the table above, it is obvious that using the advance basis restart speeds up computation. The advantage of using IPM becomes redundant as soon as the efficient frontier is built with the advanced basis technique. In the case of QPF4500, the computation of one point on average (advanced basis restart) took 3.7 seconds compared with the computation of one singe problem of QPF4500 using IPM took 9.36 seconds.

6.3 The discrete efficient frontier(DEF)

Model statements

The two models used for the computation of the discrete efficient frontier (DEF) originated from the described industrial implementation and are given in its dimension in table 3.3.

The study in the previous section showed that the factor model is desirable for implementation of large scale portfolio selection. Therefore, two different models are given

QMIPFR

$$\begin{aligned}
\min \quad & \sum_{i=1}^{M}\sum_{j=1}^{M}\beta_i\beta_j\sigma_{f_{ij}}^2+\sum_{i=1}^{N}x_i^2\sigma_{e_i}^2 && && (6.10)\\
\text{s.t.} \quad & \sum_{i=1}^{N}x_i=1, && && (6.11)\\
& \sum_{i=1}^{N}\mu_i x_i\geq\rho, && && (6.12)\\
& \beta_m=\sum_{i=1}^{N}x_i\beta_{im} && m=1,...,M && (6.13)\\
& \sum_{i=1}^{N}\delta_i\leq c, && && (6.14)\\
& x_i\leq ut_i\delta_i, && i=1,...,N && (6.15)\\
& x_i\geq lt_i\delta_i, && i=1,...,N && (6.16)\\
& x_i\geq 0, && i=1,...,N && (6.17)
\end{aligned}$$

where c is the cardinality and lt_i, ut_i are the lower and upper threshold for asset i. The variable δ_i is a binary variable indicating if stock i in the portfolio or not which is reflected in the cardinality constraint 6.14. This constraint is associated with the buy-in restrictions 6.16, 6.15 that put lower and upper bounds on the portfolio weights.

In the next model the two objectives are combined in one, minimising the risk on one hand and maximising return on the other.

QMIPFL

$$\begin{aligned}
\min \quad & \lambda\Big(\sum_{i=1}^{M}\sum_{j=1}^{M}\beta_i\beta_j\sigma^2_{f_{ij}} + \sum_{i=1}^{N}x_i^2\sigma^2_{e_i}\Big) - (1-\lambda)\sum_{i=1}^{N}\mu_i x_i && && (6.18)\\
\text{s.t.} \quad & \sum_{i=1}^{N}x_i = 1, && && (6.19)\\
& \beta_k = \sum_{i=1}^{N}x_i\beta_{ik}, && k=1,...,K && (6.20)\\
& \sum_{i=1}^{N}\delta_i \le c, && && (6.21)\\
& x_i \le ut_i\delta_i, && i=1,...,N && (6.22)\\
& x_i \ge lt_i\delta_i, && i=1,...,N && (6.23)\\
& x_i \ge 0, && i=1,...,N && (6.24)
\end{aligned}$$

where λ is a constant in the interval $[0,1]$. By varying the value between zero and one, the points on the efficient set are traced out.

Discontinuities of the DEF

The efficient frontier with discrete variables does not have the same properties as the CEF. The discrete efficient frontier (DEF) exhibits discontinuities in which the selected portfolio would be inefficient. In case of a discontinuity in the efficient frontier, the chosen portfolio depends on the investor's preference.

Example

In order to visualise discontinuities of the DEF, the example with four assets given by Chang [16] is revisited. The given data set was investigated by Horniman [39]. Considered is a portfolio selection model with a universe of 4 stocks. To construct the DEF, the portfolio selection model is built by setting the cardinality $c = 2$.

QMIPC

$$\min \sum_{i=1}^{N}\sum_{j=1}^{N} \sigma_{ij}x_i x_j \tag{6.25}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} x_i = 1, \tag{6.26}$$

$$\sum_{i=1}^{N} \mu_i x_i \geq \rho, \tag{6.27}$$

$$\sum_{i=1}^{N} \delta_i \leq c \tag{6.28}$$

$$x_i \leq ut_i\delta_i \qquad i = 1, ..., N \tag{6.29}$$

$$x_i \geq lt_i\delta_i \qquad i = 1, ..., N \tag{6.30}$$

$$x_i \geq 0, \qquad i = 1, ..., N \tag{6.31}$$

$$\delta_i = 0 \text{ or } 1 \qquad i = 1, ..., N \tag{6.32}$$

The computation of the DEF is processed by gradually changing ρ and it is shown in Figure 6.3, 6.4. The DEF begins at curve MEF 1-3 and continues until the tangency with curve MEF 2-4 is found. At the tangency to the lower curve the smoothness in the increasing gradient is maintained but some efficient points are missing.

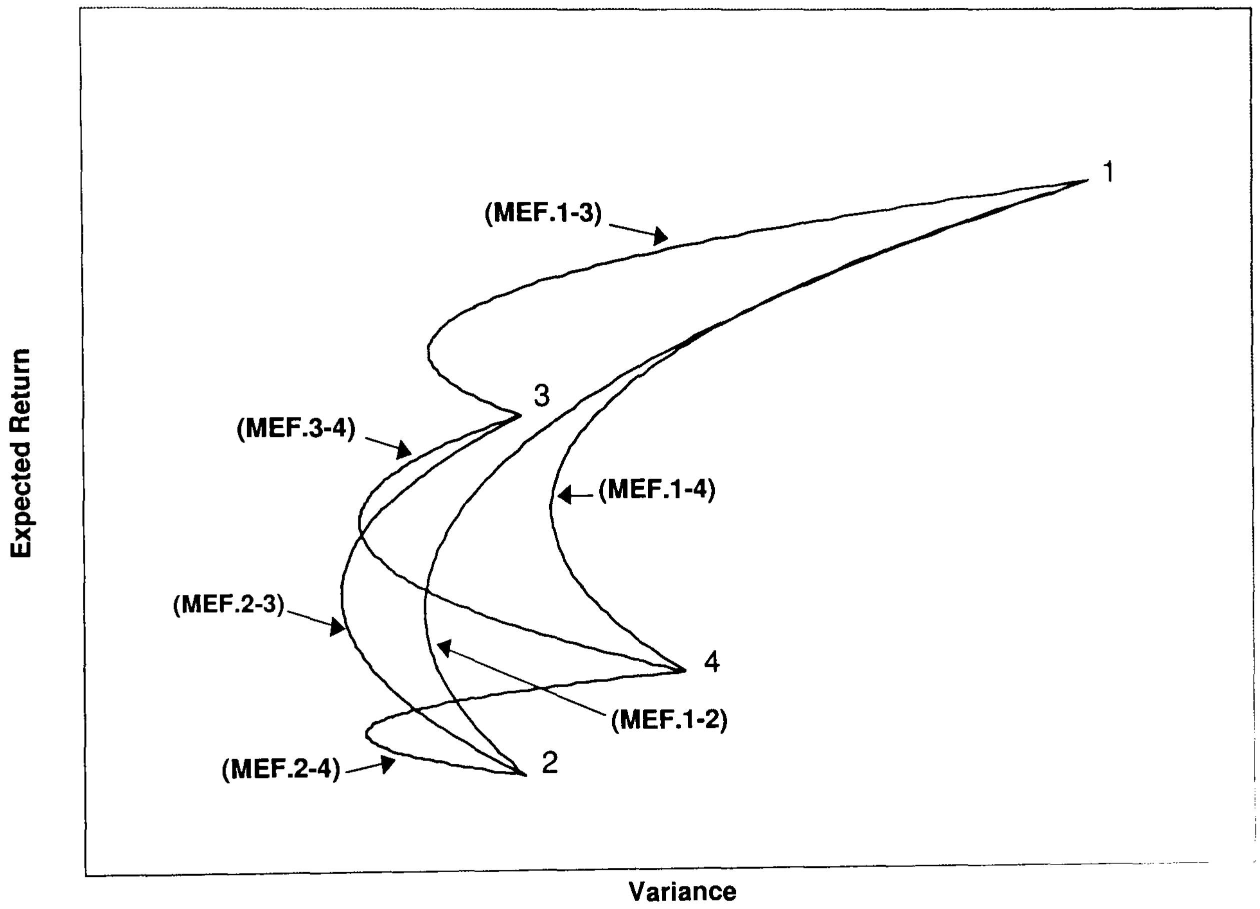

Figure 6.3: DEF: A four stock example

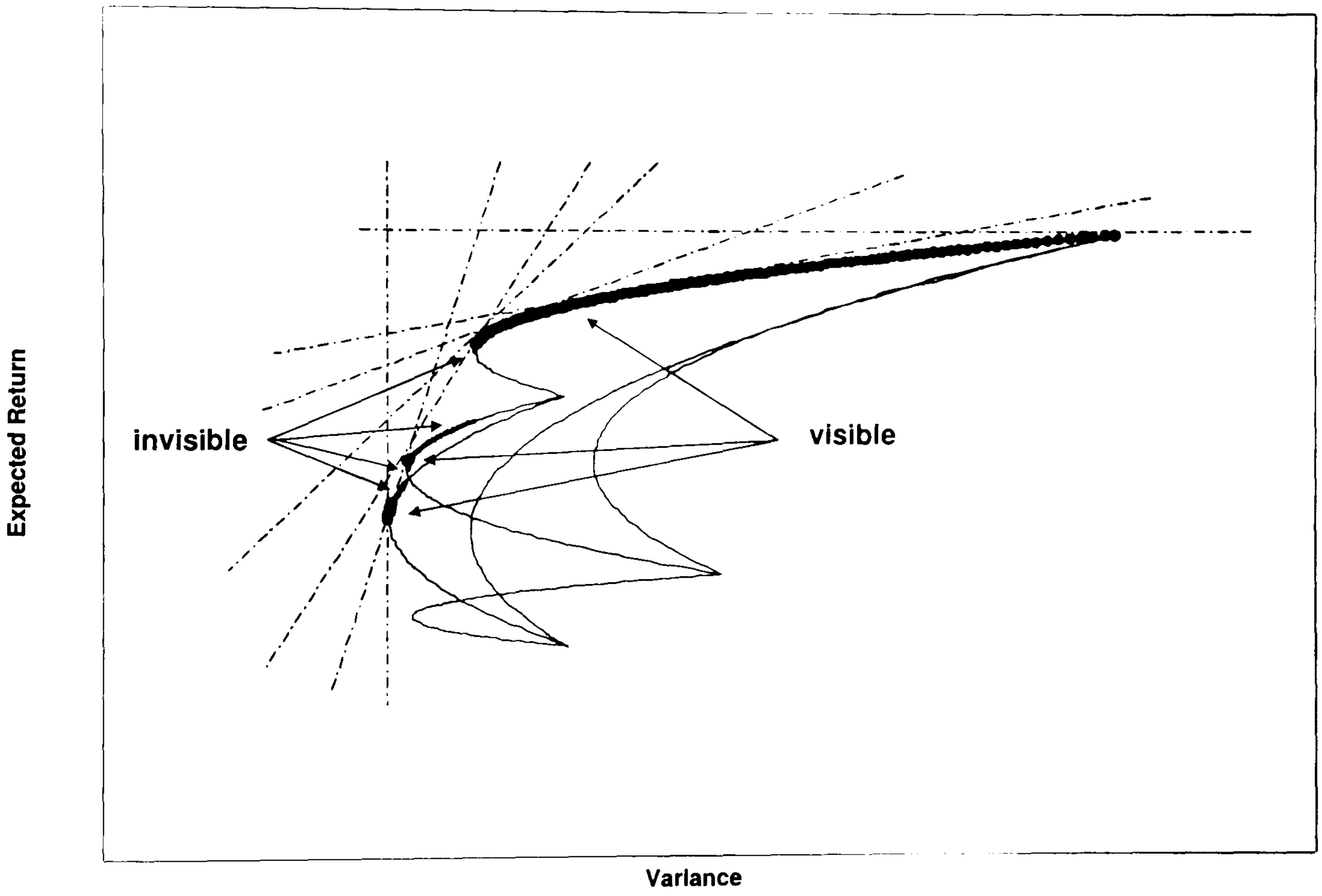

Figure 6.4: DEF: A four stock example (expanded)

Computational algorithm

The strategy for the computation of the discrete efficient frontier is similarly performed to the CEF. During the first computation of QMIP problem both, the binary and the 'continuous' basis is reported. In the subsequent stages either both are used or just one of them. Naturally, using both the binary and the continuous basis to warm start the following QMIP problem should achieve the same solution though faster.

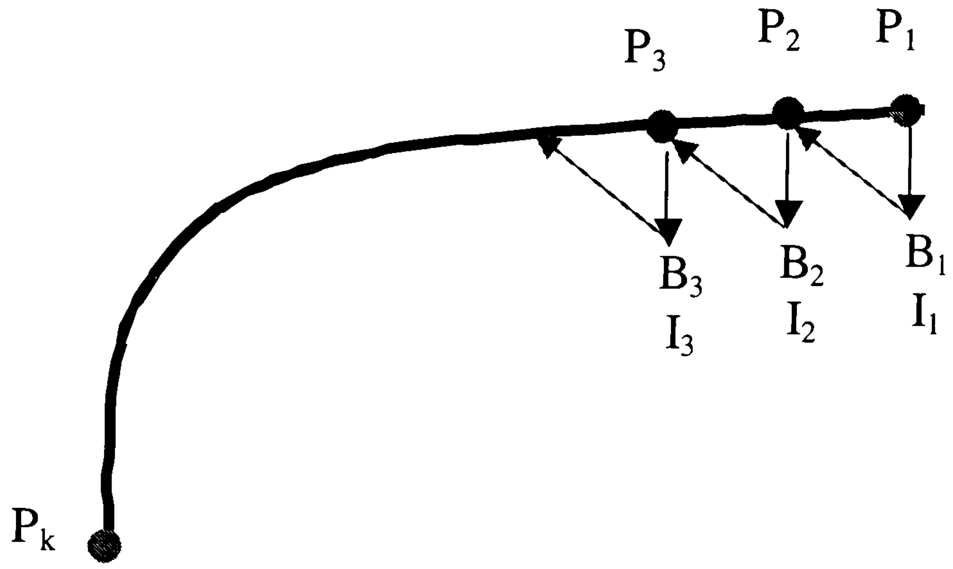

Figure 6.5: Restart techniques for DEF

The discrete efficient frontier (DEF)

The discrete factor portfolio planning model can be written in two different forms. There is on the one hand the factor model (QPL)using the lambda formulation combining the two objectives of a portfolio optimisation in one objective minimising risk and maximising return. While on the other hand, there is the factor model (QPR) which minimises risk subject to a certain level of return. Since this section is dedicated to the discrete efficient frontier, in the two continuous factor models (QPL and QPR) cardinality as well as buy-in constraints are added. Thus, the given factor models become the two quadratic mixed integer programming (QMIPFL and QMIPFR) problems.

The following table gives the universe of stocks with the chosen cardinalities for the various factor models:

Number of stocks	50	150	300
Cardinality	3	20	30

From the table below, the computation of the DEF using either with or without restart is shown.

	DEF without restart		DEF with restart	
	Number of iterations	Total time	Number of iterations	Total time
QMIPFR50	3588943	3475.27	355396	2580.13
QMIPFR150	59883797	37336.11	8075805	23373.97
QMIPFR300	127933358	110560.8	18267631	71671.02

Table 6.2: Computation time of DEF for QMIPF models

Comparing the two different methods, one can clearly see that computing the DEF with restart does better.

Chapter 7

Discussion and conclusions

This thesis is concerned with the modelling of portfolio optimisation problems presented in the Markowitz MV framework. Therefore its main focus is the solution of the quadratic and quadratic mixed integer programming problems.

7.1 Summary of contributions

Portfolio optimisation models

In chapter 3, a family of real world portfolio optimisation problems are introduced; these models include buying and selling of assets for portfolio re-balancing and also (market index) benchmark tracking. Other features of the model are the sector and industry groupings which are used to create realistic restrictions by such categories. This approach provides exogenous control in respect of diversification. The constraints are presented in an elastic (goal) programming form which makes the model fairly amiable to tuning.

Solution method for QP models

Out of the many alternative algorithms for solving QP problems, sparse simplex and the interior point method are investigated (see chapter 4). The computational study shows that the implementation of a crossover between IPM and SSX could be of advantage especially taking into consideration the computationally encouraging results reported by Levkovitz et.al. [55] and others in respect to crossover for LP problems. A crossover routine is implemented and negative results are reported which show that there is little advantage of IPM to SSX crossover compared to straight application of SSX.

The three approaches SSX, IPM and XO are applied to the test models which include Maros et.al [69] and our continuous portfolio planning models (QPF and QPMV). The study shows that the computational time for the factor based portfolio planning model (QPF) is superior to the full variance/covariance matrix (original) Markowitz model (QPMV).

Solution methods for QMIP models

The actual industrial portfolio model investigated in this thesis is a discrete portfolio planning model in which the discrete constraints, that is cardinality and thresholds are considered. The branch and bound based algorithm for QMIP problems is developed in depth. It is shown that (a) use of basis restart to process a family of models and (b) use of dual algorithm enhance the processing speed. The scale up property of this discrete programming problem is reported as a heuristic. For these large scale NP-hard models under consideration it is not possible to complete the full search for the discrete optimum. However. the results presented show encouraging scale up property in respect of good solutions which are obtained rapidly (see chapter 5.

Computing the efficient frontier

The computation of the continuous efficient frontier can be easily achieved by simply reapplying the QP solver. However, for large scale models using the basis restart technique allows the processing of the family of models efficiently again with good scale up property (see chapter 6).

The consideration of discrete constraints allowing a more realistic model change the appearance of the efficient frontier. For the discrete constrained efficient frontier problem, there are two contributions. First, it is identified how the lambda formulation leads to a missing portion of the frontier. Then by using integer restart, dominated solutions are eliminated. Thus relative efficiency of the discrete feasible solutions is achieved. The computation of the discrete efficient frontier are performed using two QMIP methods - standard *B&B* and *B&B* with restart. These are contributions to current research in portfolio selection.

7.2 Research challenges and future directions

The discussed portfolio planning model uses symmetric risk measures. Recently, semi-variance, conditional Value-at-Risk are brought forward as risk measures for a portfolio. Attempting to solve the portfolio planning model with a universe of 5000 stocks using asymmetric risk measures implies the solution of a non-linear model. The investigation of these problems would be challenging. Furthermore, the mean-variance approach can be generalised introducing multiple chance constraints. The formulation of the uncertain return in form of chance constraints results in a model with linear or quadratic objective functions and quadratic (or more general non-linear) constraints. The solution of such models provide ambitious research problems.

References

[1] E.D. Andersen. On exploiting problem structure in a basis identifications procedure for linear programming. Technical Report, 1996.

[2] S. Balas, E. Ceria, G. Cornuejols, and N. Natraj. Gomory cuts revisited. *Operations Research Letters*, 19:1–9, 1996.

[3] M. Balinski. Integer programming: Methods, uses, computation. *Management Science*, 12:253–313, 1965.

[4] E.M.L. Beale. On quadratic programming. *Naval Research Logistics Quarterly*, 6:227–243, 1959.

[5] E.M.L. Beale. Branch and bound methods for mathematical programming systems. *Annals of Discrete Mathematics*, 5:201–219, 1969.

[6] J.E. Beasley. *Advances in linear and integer programming.* Oxford University Press, 1996.

[7] J.E. Beasley, N. Meade, and T.-E. Chang. An evolutionary heuristic for the index tracking problem. Report (Revised October 2001),Imperial College, U.K., 1999.

[8] J.F. Benders. Partitioning procedures for solving mixed variable programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[9] A.B. Berkelaar, B. Jansen, K. Roos, and T. Terlaky. Basis- and partition identification for quadratic programming and linear complementarity problems. *Mathematical Programming*, 86(2):261–282, 1999.

[10] D. P. Bertsekas. *Nonlinear Programming: 2nd Edition.* Athena Scientific, 1999.

[11] D. Bienstock. Computational study of a family of mixed-integer quadratic programming problems. *Mathematical Programming*, 74:121–140, 1996.

[12] R.E. Bixby, J.W. Gregory, I.J. Lustig, R.E. Marsten, and D.F. Shanno. Very large-scale linear programming: A case study in combining interior point and simplex method. *Operations Research*, 40(5):885–897, 1992.

[13] L. Camarinopoulos and G. Zioutas. Formulating robust regression estimation as an optimum allocation problem. *Journal of Statistical Computation and Simulation*, 9(72):1–19, 2002.

[14] T.J. Carpenter, I.J. Lustig, J.M. Mulvey, and D.F. Shanno. Separable quadratic programming via a primal-dual interior point method and its use in a sequential procedure. *ORSA Journal on Computing*, 5(2):182–191, 1993.

[15] G. Champleboux, S. Lavallee, R. Szeliski, and L. Brunie. From accurate range imaging sensor calibration to accurate model-based 3-d object localization. *In Proc. CVPR*, pages 83–89, 1992.

[16] T.-J. Chang, N. Meade, Y.M. Sharaiha, and J.E. Beasley. Heuristics for cardinality constrained portfolio optimisation. *Computers and Operations Research*, 27:1271–1302, 2000.

[17] A. Charnes, W.W. Cooper, and R. Ferguson. Optimal estimation of executive compensation by linear programming. *Management Science*, 1:138–151, 1955.

[18] R.W. Cottle and G.B. Dantzig. Complementary pivot theory of mathematical programming. *Linear Algebra and Its Applications*, 1:103–125, 1968.

[19] H. Crowder, E. L. Johnson, and M. W. Padberg. Solving large-scale zero-one linear integer problems. *Operations Research*, 31:803–834, 1983.

[20] B. Dantzig, D.R. Fulkerson, and S. M. Johnson. Solution of a large scale travelling-salesman problem. *Operations Research*, 2:393–410, 1954.

[21] G.B. Dantzig. *Linear programming and extensions.* Princeton University Press, Princeton, 1963.

[22] R. Dorfman. *Application of linear programming to the theory of the firm.* University of California Press, Berkeley, California, 1951.

[23] M.A. Duran and I.E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307–339, 1986.

[24] M. Ehrgott. Multicriteria optimization. Vorlesungsskript, University of Kaiserslautern, Germany, 1999.

[25] R. Fletcher and S. Leyffer. Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming*, 8(2):327–349, 1994.

[26] O.E. Flippo and A.H.G. Rinnoy Kan. A note on benders decomposition in mixed integer quadratic programming. *Operations Research Letters*, 9:81–83, 1990.

[27] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of* NP-*Completeness*. .H. Freeman and Co, 1979.

[28] R.S. Garfinkel and G.L. Nemhauser. *Integer Programming*. John Wiley, NewYork, 1972.

[29] A.M. Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.

[30] G.H. Golub and M.A. Saunders. Linear least squares and quadratic programming. In J. Abadie, editor, *Integer and nonlinear programming*, pages 229–256. North Holland Publishing Company, 1970.

[31] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 1983.

[32] R.E. Gomory. An algorithm for integer solutions to linear programs. In R.L. Graves and P. Wolfe, editors, *Recent Advances in Mathematical Programming*, pages 269–302. McGraw Hill, New York, 1963.

[33] J. Gondzio. Presolve analysis of linear programs prior to applying an interior point method. *INFORMS Journal on Computing*, 9(1):73–91, 1997.

[34] M. Gürtler, G. Mitra, and E.F.D. Ellison. Solution of quadratic problems by the interior point method and sparse simplex: A computational study. Technical Report, Department of Mathematical Sciences, Brunel University, 2003.

[35] M. Gürtler, G. Mitra, and E.F.D. Ellison. Solving large scale QP and QMIP portfolio models: Discussion of algorithms. Technical Report, Department of Mathematical Sciences, Brunel University, 2003.

[36] O. Gueler and Y. Ye. Convergence behaviour of interior-point algorithms. *Mathematical Programming*, 60(2):215–228, 1993.

[37] O. K. Gupta and A. Ravindran. Nonlinear integer programming algorithms: A survey. *OPSEARCH*, 20:189–206, 1983.

[38] G. Hanoch and H. Levy. The efficiency analysis of choices involving risk. *Review of Economic Studies*, 36:335–346, 1969.

[39] M.D. Horniman. *Constructing Efficient Portfolios: Alternative Models and Discrete Constraints - A Computational Study.* PhD thesis, Department of Mathematical Sciences, Brunel University, 2001.

[40] E. Horowitz and S. Sahni. *Fundamentals of computer algorithms.* Pitman, London, 1978.

[41] J.P. Ignizio. *Linear programming in single & multiple-objective systems.* Prentice-Hall International, London, 1982.

[42] Northfield Inc. *Open optimizer for windows.* Northfield publications.

[43] N.J. Jobst, M.D. Horniman, C.A. Lucas, and G. Mitra. Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. *Quantitative Finance*, 1:1–13, 2001.

[44] H. Jones and G. Mitra. Solution of the convex quadratic programming problem using the interior point method. Technical Report, TR/03/97, Department of Mathematics and Statistics, Brunel University, U.K., 1997.

[45] P. Jorion. Bayes-stein estimation for portfolio analysis. *Journal of Financial and Quantitative Analysis*, 21(3):279–292, 1986.

[46] J.B. Kallberg and W.T. Ziemba. Comparison of alternative utility functions in portfolio selection problems. *Management Science*, 29:1257–1276, 1983.

[47] N.K. Karmarkar. A new polynomial time algorithm for linear programming. Technical Report, AT&T Bell Laboratories, Murray Hill New Jersey, 1984.

[48] J.S.H. Kornbluth. Duality, indifference and sensitivity analysis in multiple objective linear programming. *Operational Research Quarterly*, 25:599–614, 1974.

[49] F. Körner. Integer quadratic optimization: A new branching rule for the branch and bound algorithm for solving nonlinear integer programming problems. *BIT*, 28:701–708, 1988.

[50] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492. Berkeley, California, 1951.

[51] A. Land and A. Doig. An automatic method for solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.

[52] E. Lawler and D. Wood. Branch-and-bound methods: A survey. *Operations Research*, 1:699–719, 1966.

[53] R. Lazimy. Improved algorithm for mixed-integer quadratic programs and a computational study. *Mathematical Programming*, 32:100–113, 1985.

[54] C.E. Lemke. On complementary pivot theory. In G.B. Dantzig and A.F. Veinott, editors, *Mathematics of the Decision Sciences Part I*, pages 95–114. Lect. Appl. Math. 11, 1968.

[55] R. Levkovitz and G. Mitra. Solution of large-scale linear programs: A review of hardware, software and algorithm issues. In T.A. Ciriani and R.C. Leachman, editors, *Optimisation in Industry*, pages 139–172. John Wiley & Sons, Dordrecht, The Netherlands, 1993.

[56] R. Levkovitz and G. Mitra. Experimental investigations in combining primal dual interior point method and simplex based LP solvers. *Annals of Operations Research*, 58:19–38, 1995.

[57] J.D.C Little, K.G. Murty, D.W. Sweeney, and C. Karel. An algorithm for the travelling salesman problem. *Operations Research*, 14:972–989, 1963.

[58] D.G. Luenberger. *Linear and nonlinear programming, 2nd edition*. Addison-Wesley, 1984.

[59] D.G. Luenberger. *Investment science*. Oxford University Press, 1998.

[60] I.J. Lustig, R.E. Marsten, and D.F. Shanno. Interior point methods for linear programming: Computational state of the art. *ORSA Journal on Computing*, 6(1):1–14, 1994.

[61] C. L. Mallows. Some comments on cp. *Technometrics*, 15(4):661–675, 1973.

[62] R. Mansini and M.G. Speranza. Heuristic algorithms for the portfolio selection problem with minimum transaction lots. *European Journal of Operational Research*, 114:219–233, 1999.

[63] H.M. Markowitz. Portfolio selection. *Journal of Finance*, 7:7–91, 1952.

[64] H.M. Markowitz. The optimization of a quadratic function subject to linear constraints. *Naval Research Logistics Quarterly*, 3:111–133, 1956.

[65] H.M. Markowitz. The elimination form of the inverse and its application to linear programming. *Management Science*, 3:255–269, 1957.

[66] H.M. Markowitz. *Portfolio Selection: Efficient diversification of investments.* John Wiley, Yale University Press, 1970, 1959.

[67] H.M. Markowitz and G.P. Todd. *Mean-variance analysis in portfolio choice and capital markets.* Frank J. Fabozzi Associates, New Hope, Pennsylvania, 2000.

[68] I. Maros. *Computational Techniques of the Simplex Method.* Kluwer Academic Publishers, Boston, 2003.

[69] I. Maros and Cs. Mészáros. A repository of convex quadratic programming problems. Department of Computing, Imperial College, U.K.

[70] I. Maros and G. Mitra. Investigating the sparse simplex algorithm on a distributed memory multiprocessor. *Parallel Computing*, 26:151–170, 2000.

[71] L. McLinden. The analogue of moreau's proximation theorem, with applications to the nonlinear complementarity problem. *Pacific Journal of Mathematics*, 88:101–161, 1980.

[72] N. Megiddo. On finding primal- and dual-optimal bases. *ORSA Journal of Computing*, 3:63–65, 1991.

[73] S. Mehrotra. On the implementation of a (primal-dual) interior point method. Technical Report 90-03, Department of Industrial Engineering and Management Sciences, Northwestern University, Evaston, IL, 1990.

[74] Cs. Meszaros. The BPMPD interior point solver for convex quadratic problems. Working paper WP 98-8, Computer and Automation Institute, Budapest, 1998.

[75] J.E. Mitchel. *Branch-and-cut methods for combinatorial optimization problems. In Handbook of Applied Optimization.* Oxford University Press, 2000.

[76] G. Mitra. Investigation of some branch and bound strategies for the solution of mixed integer linear programs. *Mathematical Programming*, 4:155–170, 1973.

[77] G. Mitra. *Theory and application of mathematical programming.* Academic Press, 1976.

[78] L.G. Mitten. Branch and bound methods: General formulations and properties. *Operations Research*, 18:24–34, 1970.

[79] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimisation.* Wiley, 1989.

[80] V.L. Nwana. *Parallel algorithms for solving mixed integer linear programs.* PhD thesis, Department of Mathematical Sciences, Brunel University, 2001.

[81] C.H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: Algorithms and complexity.* Dover Publications, Inc., New York, 1998.

[82] I. Quesada and I.E. Grossmann. An LP/NLP based branch-and-bound algorithm for convex MINLP optimization problems. *Computers and Chemical Engineering*, 16:937–947, 1992.

[83] R.T. Rockafellar. *Convex analysis.* Princeton University Press (c 1970), 1972.

[84] C. Roos, T. Terkaly, and J.-Ph. Vial. *Theory and algorithms for linear optimization: An interior point approach.* John Wiley & Sons, 1997.

[85] B. Rosenberg. Extra-market components of covariance in security returns. *Journal of Financial and Quantitative Analysis*, 9:263–273, 1974.

[86] W.F. Sharpe. A linear programming approximation for the general portfolio analysis problem. *Journal of Financial and Quantitative Analysis*, 6:1263–1275, 1971.

[87] C. Van-de Panne and A. Whinston. The simplex and the dual method for quadratic programming. *Operational Research Quarterly*, 15:355–388. 1964.

[88] C. Van-de Panne and A. Whinston. The symmetric formulation of the simplex method for quadratic programming. *Econometrica*, 37(3):507–527, 1969.

[89] R.J. Vanderbei. LOQO: An interior point method for quadratic programming. Technical Report, Princeton University, February 1995.

[90] R.J. Vanderbei. *Linear Programming: Foundations and Extensions.* Kluwer Academic Press, 1996.

[91] W.B. White, S.M. Johnson, and G.B. Dantzig. Chemical equilibrium in complex mixtures. *Journal of Chemical Physics*, 28(5):751–755, 1958.

[92] P. Wolfe. The simplex method for quadratic programming. *Econometrics*, 27:382–398, 1959.

[93] P. Wolfe. A duality theorem for non-linear programming. *Quart. Appl. Math.*, 19:239–244, 1961.

[94] P. Wolfe. The composite simplex algorithm. *SIAM Review*, 7(1):42–54, 1965.

[95] S.J. Wright. *Primal-Dual Interior-Point Methods.* SIAM, 1997.

[96] S.A. Zenios. *Practical Financial Optimization: Decision making for financial engineers.* Manuscript, *HERMES* Center on Computational Finance and Economics, Cyprus, 2001.