ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

DEPARTMENT OF ECONOMICS

ACTIVE PORTFOLIO MANAGEMENT WITH APPLICATIONS

By
Papadakis Dimitrios (S.N.: 865)

Diploma thesis
Submitted in part fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Advisor: Assoc. Prof. Gregory Kordas
Co-advisor: Prof. E. Tzavalis

Athens
December 2007

We approve the master thesis of Papadaki Dimitrios

Signature

Γρηγόρης Κόρδας

Gregory Kordas
Athens University of Economics and Business

Elias Tzavalis
Athens University of Economics and Business

Date 19/12/2007

To my family
Manolis, Georgia, Maria, Katerina

ACKNOWLEDGEMENTS

I would want firstly to thank my professor and supervisor of this work, Gregory Korda for the guidance, the precious advices and the interest that he showed at all the duration of this thesis. Furthermore, I would like to thank professor and co-advisor of this thesis Elia Tzavali for his time to read my master thesis.

At this point I also want to thank my parents for the moral and material support at all the duration of my studies at Athens University of Economic and Business. Special thanks to my sister Katerina for her supporting at many difficulties during this thesis.

Contents

List of Figures

List of tables

ABSTRACT

In this thesis we investigate the application of the heuristic method Simulated Annealing (SA) in order to optimize realistic portfolios. The model is based on the classical mean-variance approach but we improved it with the addition of many constraints that are playing a vital role in portfolio selection.

It is shown that Simulated Annealing (SA) can optimize these portfolios effectively and within reasonable time because the complexity of this problem makes the time of computation very important. This approach is also flexible because it can easily cope with extensive modifications such as the addition of complex new constraints, discontinuous variables and changes in the objective function.

Master thesis is organized in seven chapters. Chapter 1 focuses on the introduction of problem of portfolio optimization and it gives a general concept abut this work. Chapter 2 covers the general theories of Portfolio Optimization and its vital role in Financial Science. Chapter 3 includes theory and applications of Mean Variance Analysis from risky assets only and we give some extensions of the basic model. Chapter 4 covers the important points of Heuristic Optimization and in the next chapter 5 we mention the heuristic optimization technique of Simulated Annealing for portfolio selection. In chapter 6 we introduce our results which are came for the algorithm of Simulate Annealing. Final, in chapter 7 we give the synopsis of master thesis and we discuss for future improvements in some relative topics with our master thesis.

Chapter 1

INTRODUCTION

Markowitz' mean-variance model of portfolio selection is one of the best known models in finance that gives the necessary economic background to determine the composition of a portfolio of assets that minimize risk for a given level of expected return. While the basis for portfolio optimization was established by Markowitz, it is often difficult to incorporate real-world constraints and dilemmas into the classical theory, which can limit its use. To be more specific, Markowitz' model ignores practical subjects such as transaction costs, liquidity constraints (and the resulting nonlinearities in transaction costs which result from this), minimum lot sizes and cardinality constraints, i.e. the restriction of a portfolio to a certain number of assets.

A fundamental reason for the "hardness" of the portfolio selection problem is the number of possible portfolios, making solution by enumeration a very difficult task. The horrors of enumeration can be illustrated as follows.

Suppose we have a universe of N assets and we want to select a number of K assets (K<N) in order to create optimal portfolios. The number of possible combinations is

$$C_K^N = \binom{N}{K} = \frac{N!}{K! \cdot (N-K)!}$$

Suppose now that for each K-asset portfolio the asset weights are defined with a resolution of r, so for example if $r = 1$ the asset's weighting is 100% (or 0%), if $r = 2$ its weighting is either 50% or 100% (or 0%). (The number of weighting possibilities is given by $r+1$ and the percentage resolution is given by $p = \frac{100}{r}$, so a weighting with a percentage resolution of 1% will require $r = 100$). Clearly, the total number of possible portfolios with different combinations of asset weights is given by K^{r+1}.

But only a subset of previous combinations satisfy the budget constraint (asset weights sum to 100%). This is known as $C'(n,k)$ a *k-composition of n*, which is a partition of *n* into xactly *k* parts, *with regard to order,* where each part is an integer greater than or equal to zero. The number of compositions is given by $C'(n,k) = C_{k-1}^{n+k-1}$.

The total number of enumeration possibilities *E* is therefore given by

$$E = C_K^N \cdot C'(r+K-1, K-1) = C_K^N \cdot C_{K-1}^{r+K-1} = \frac{N!}{K! \cdot (N-K)!} \cdot \frac{(r+K-1)!}{(K-1)! \cdot (r)!}$$

For example, searching for the optimal 30-stock portfolio selected from a universe of 50 stocks and wish weightings to be defined within 1%. Therefore $N = 50$, $K = 30$ and $p = 1$, giving $r = 100$.

So $E = C_{30}^{50} \cdot C_{29}^{79} = \left(\frac{50!}{30!20!}\right) \cdot \left(\frac{79!}{29!50!}\right) = 1{,}53 \times 10^{14}$ *portfolios*

For example the Cray T3E supercomputer operates at 2.4 teraflops. Assume that the evaluation of each portfolio will require around 300 floating-point operations. Therefore to evaluate each portfolio the Cray will take

$\frac{300\, flop/portfolio}{2.4 \times 10^{12}\, flop/\sec ond} = 1.25 \times 10^{-10}$ sec*onds* / *portfolio* or

will process 8×10^9 portfolios/second)

The time required to evaluate all the possible portfolios is therefore

$$(1.25 \times 10^{-10} \text{ sec}/\,portfolio) \times 1.53 \times 10^{14}\, portfolios = 19130 \text{ sec}onds$$
$$or\ 5\ hours\ and\ 20\ \text{min}utes$$

Optimization by enumeration could be tedious as the number of assets in our universe increases on the number of 100.

Portfolio selection issues

In order to handle portfolio selection problems in a formal way, we must answer at least three questions:

1. Data modeling, in particular the behavior of asset returns.

2. The choice of the optimization model, including
• the nature of the objective function;
• the constraints faced by the investor.

3. The choice of the optimization technique.

The first step is to understand the nature of our data and to be able to correctly represent them. The Markowitz model assumes that the asset returns follow a multivariate normal distribution. In particular, the first two moments of the distribution suffice to describe completely the distribution of the asset returns and the characteristics of the different portfolios. But in the real markets this assumption is not completely true because in the real markets we can exhibit more intricacies, with distributions of returns depending on moments of higher-order (skewness, kurtosis, etc.), and distribution parameters varying over time. Analyzing and modeling such complex financial data is a whole subject in itself, which we do not tackle in this thesis explicitly. We adopt the classical assumptions of the Mean-Variance analysis, where the expected returns and the variance–covariance matrix are supposed to provide a satisfactory description of the asset returns.

When implementing an optimization model of portfolio selection, a second question consists in identifying the objective of the investor and the constraints that he is facing. As far as the objective goes, the quality of the portfolio could be measured using a wide variety of utility functions. Following again Markowitz model, we assume here that the investor is risk averse and wants to minimize the variance of the investment portfolio subject to the expected level of final wealth. It should be noted, however, that this assumption does not play a crucial role in our algorithmic developments, and that the objective could be replaced by a more general utility function without much impact on the optimization techniques that we propose.

In our model, we are interested in two types of complex constraints

- limiting the number of assets included in our portfolio (thus reflecting some behavioral or institutional restrictions faced by the investor),
- the minimal quantities which can be traded when rebalancing an existing portfolio (thus reflecting individual or market restrictions).

In the final question we must choose what portfolio selection method we use for the optimization of the chosen model. Because of the complexity of our model, we have chosen to work with a simulated annealing metaheuristic.

Objectives

The purpose of our research is to investigate the ability of the Simulated Annealing (SA) which is a metaheuristic method to create high quality solutions for the mean-variance model when enriched by practical constraints.

The Markowitz model is extended by

- floor and ceiling constraints
- cardinality constraints
- turnover (purchase) constraints
- turnover (sale) constraints
- trading constraints

Heuristic approaches are very attractive because they are independent of the objective function and the structure of the models and constraints. These advantages make the method general and robust.

Chapter 2

Portfolio Optimization

A common property of investment opportunities is that their actual returns might differ from what has been expected, or in short: they are *risky*. This notion of financial risk, defined by the (potential) deviation from the expected outcome, includes not only a lower than expected outcome (*downside risk*) but also that the actual return is better than initially expected (*upside risk*) because of positive surprises or non-occurrences of apprehended negative events.

When all available information and expectations on future prices are contained in current prices, then the future payoffs and returns can be regarded and treated as random numbers. In the simplest case, the returns of an asset *i* can be described with the normal distribution: the *expected value* (*mean*) of the returns, $E(r_i)$ and their *variance*, σ_i^2 (or its square root, σ_i in the finance literature usually referred to as *volatility*) capture all the information about the expected outcome and the likelihood and range of deviations from it.

When comparing investment opportunities and combining them into portfolios, another important aspect is how strong their returns are "linked", i.e., whether positive deviations in the one asset tend to come with positive or negative deviations in the other assets or whether they are independent. If the assets are not perfectly positively correlated, then there will be situations where one asset's return will be above and another asset's return below expectance. Hence, positive and negative deviations from the respective expected values will tend to partly offset each other. As a result, the risk of the combination of assets, the *portfolio*, is lower than the weighted average of the risks of the individual assets. This effect will be the more distinct the more diverse the assets are. The intuition is that similar firms (and hence their stocks) do similarly poorly at the same time whereas in heterogeneous stocks, some will do better than expected while others do worse than expected. The positive and negative deviations from the expected values will then (to some degree) balance, and the actual deviation from the portfolio's expected return will be smaller than would be the deviation from an asset's expected return even when both have the same expected return.

Technically speaking, the risk and return of a portfolio P consisting of N risky assets can be treated as a convolution of the individual assets' returns and co-variances when the included assets can be described by the distributions of their returns.

Harry M. Markowitz was the first to come up with a parametric optimization model to this problem which meanwhile has become the foundation for *Modern Portfolio Theory* (*MPT*).

The model of Markowitz for portfolio analysis can be summarized as follows. First, the two relevant characteristics of a portfolio are its expected return and the variance. Second, rational investors will choose to hold efficient portfolios-those that maximize expected returns for a given degree of risk or, alternatively and equivalently, minimize risk for a given expected return. Third, it is possible to identify efficient portfolios by the proper analysis of information for each security on expected return, variance of return, and the interrelationship between the return for each security and that for every other security as measured by the covariance. Finally, a computer program can utilize these inputs to calculate the set of efficient portfolios. The program indicates the proportion of an investor's fund that should be allocated to each security in order to achieve efficiency-that is, the maximization of return for a given degree of risk or the minimization of risk for a given expected return.

More analytically, in his seminal paper, Markowitz (1952) considers rational investors who want to maximize the expected utility of their terminal wealth at time T, $E(U(w_T))$. Investors are price takers and make their sole investment decision at time 0. If an investor prefers more terminal wealth to less and is risk averse, then her utility function U with respect to terminal wealth w_T has the properties

$$\frac{\partial U}{\partial w_T} > 0 \text{ and } \frac{\partial^2 U}{\partial w_T^2} < 0$$

At this point, we will give an analytical prove for the consistency for mean variance analysis.

We say that mean-variance analysis is **consistent** with expected utility maximization if there exists a **derived utility function** $V()$ that depends only on μ_R and σ_R^2 such that $V(\mu_R,\sigma_R^2) = E[U(R)]$.

Theorem 1. Mean-variance analysis is consistent with expected utility maximization, i.e. there exists $V()$ such that $V(\mu_R,\sigma_R^2) = E[U(R)]$, if and only if either of the following conditions is satisfied:

(a) R is normally distributed (or, more generally, elliptically distributed).

(b) U takes the quadratic form, i.e. $U(R) = a \cdot R - b \cdot R^2,\ \ a > 0, b > 0, R < \dfrac{a}{2b}$

The deep meaning of the theorem 1 is that the expected returns and (co-)variances contain all the necessary information not only when the returns are normally distributed (and, hence, are perfectly described with mean and variance), but also for arbitrary distributions when the investor has a quadratic utility function. More generally, it can be shown that the mean-variance framework is approximately exact for any utility function that captures the aspects *non-satiation* and *risk aversion.*

The classical Markowitz model also assumes a perfect market without taxes or transaction costs where short sales are disallowed, but securities are infinitely divisible and can therefore be traded in any (non-negative) fraction.

At this point and before we are continued with the Markowitz problem formulation, it very important to point out the definition of risk and volatility.

The term ***risk*** has been used to characterize a situation where the exact outcome is not known and where the employed risk measure indicates the magnitude of deviations from the expected value. “Risk” therefore reflects not only the “dangers” associated with an investment, but also the chances; in this sense, a risky situation is one in which surprises and unexpected developments might occur. A typical representative of this type of risk measures is the volatility of returns which not only is one of the foundations of portfolio theory as introduced so far, but will also be the prime measure of risk for most of the remainder of this contribution.

The term ***volatility*** is the most widespread risk measure in financial management and is often even used synonymously for the term risk. The reasons for this are various, most notably among them perhaps the fact that it reflects the assumption of normally distributed returns or of price processes with normally distributed price movements. Also, it measures not just potential losses but the whole range of uncertain outcomes and therefore serves the demands on a "general purpose" risk measure in finance. Not least, it also has a number of favorable (technical and conceptual) properties. Hence, this contribution, too, will use volatility as preferred measure of risk.

2.1 Risk-Return Trade-off

In selecting asset classes for portfolio allocation, investors need to consider both the return potential and the riskiness of the asset class. It is clear from empirical estimates that there is a high correlation between risk and return measured over longer periods of time. Capital market theory posits that there should be a systematic relationship between risk and return. This theory indicates that securities are priced in the market so that high risk can be rewarded with high return, and conversely, low risk should be accompanied by correspondingly lower return.

Figure 1 is a capital market line showing an expected relationship between risk and return for representative asset classes arrayed over a range of risk. Note that the line is upward-sloping, indicating that higher risk should be accompanied by higher return.

RISK – RETURN TRADE-OFF

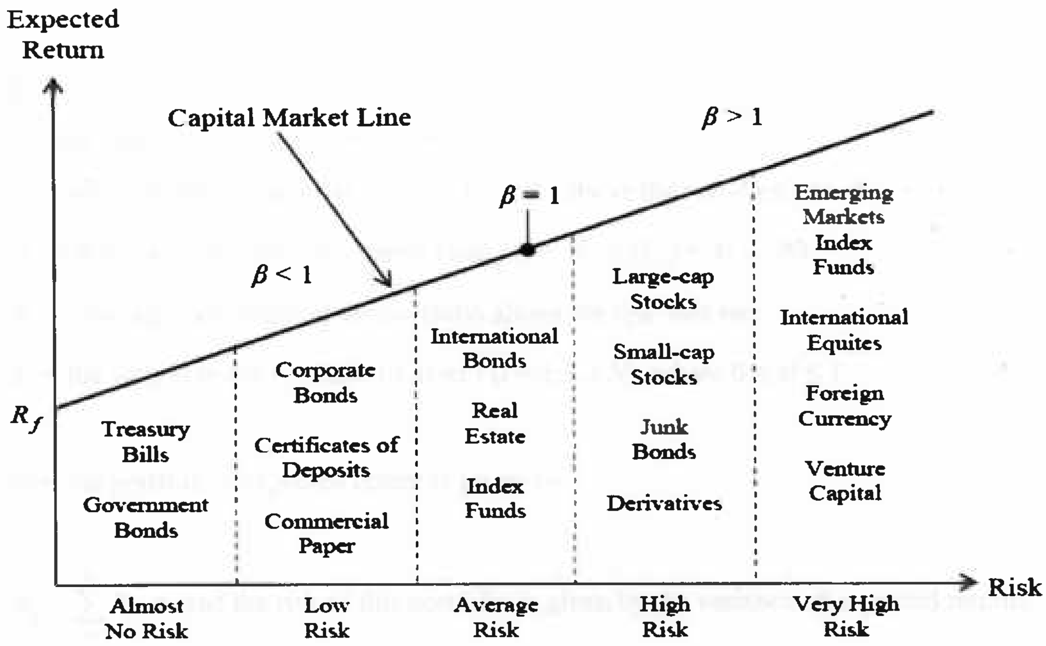


Figure 1 Relationship between risk and return

2.2 Theory and problem formulation

Unconstrained Markowitz model

If

N = the number of assets in the universe

R_i = the expected return of asset i (i = 1; ...; N) above the risk-free rate rf

σ_{ij} = the covariance between assets i and j (i = 1; ...; N, j = 1; ...; N)

R_p = the expected return of the portfolio above the risk-free rate

x_i = the weight in the portfolio of asset i (i = 1; ...; N), where $0 \le xi \le 1$

then the portfolio's expected return is given by

$R_p = \sum_{i=1}^{N} R_i \cdot x_i$ and the risk of this portfolio is given by the variance of expected returns

$$\sigma_p^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} \sigma_{ij} \cdot x_i \cdot x_j$$

The unconstrained portfolio optimization problem is therefore to

$$\text{minimize } \sigma_p^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} \sigma_{ij} \cdot x_i \cdot x_j$$

At this point we can use the fact that $\sigma_{ij} = \rho_{ij} \cdot \sigma_i \cdot \sigma_j$, where ρ_{ij} is the correlation between i and j asset and s_i, s_j represents the standard deviation of the returns. Therefore,

$$\text{minimize } \sigma_p^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} \rho_{ij} \cdot \sigma_i \cdot \sigma_j \cdot x_i \cdot x_j$$

$$\text{subject to } R_p = \sum_{i=1}^{N} R_i \cdot x_i$$

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

$$0 \le x_i \le 1 \quad i = 1, \ldots N$$

The portfolio's variance or risk is therefore minimized for a required rate of return *Rp*, while all asset weights sum to one (budget constraint). In this model you can use the excess return of the assets. This is a simple nonlinear (quadratic) programming problem which is easily solved using standard techniques.

In this form the model requires $\frac{(n^2+3n)}{2}$ items of data for an n-asset portfolio, comprising n estimates of expected returns, n estimates of variances and $\frac{n^2-n}{2}$ estimates of correlations (since the correlation matrix's diagonal elements are all one and $\rho_{ij} = \rho_{ji}$. Therefore a portfolio consisting of only 50 assets requires 1325 separate items of data while a 100-asset portfolio would require 5150 data items.

Chapter 3

Mean Variance portfolio from risky assets only

As a first step, we must determine the initial problem. Let $R = (R_1, R_2, ..., R_N)'$ be the vector of risky (random) returns of N assets that are normally distributed with **mean** vector $\mu = (\mu_1, \mu_2, ..., \mu_N)'$ and **variance-covariance** (symmetric and strictly positive-definite) matrix

$$\Sigma = \begin{bmatrix} \sigma_1^{\ 2} & \sigma_{12} & ... & \sigma_{1N} \\ \sigma_{21} & \sigma_2^{\ 2} & ... & \sigma_{2N} \\ ... \\ \sigma_{N1} & \sigma_{N2} & ... & \sigma_N^{\ 2} \end{bmatrix} = \begin{bmatrix} \sigma_1^{\ 2} & \rho_{12} \cdot \sigma_1 \cdot \sigma_2 & ... & \rho_{1N} \cdot \sigma_1 \cdot \sigma_N \\ \rho_{21} \cdot \sigma_2 \cdot \sigma_1 & \sigma_2^{\ 2} & ... & \rho_{2N} \cdot \sigma_2 \cdot \sigma_N \\ ... \\ \rho_{N1} \cdot \sigma_N \cdot \sigma_1 & \rho_{N2} \cdot \sigma_N \cdot \sigma_2 & ... & \sigma_N^{\ 2} \end{bmatrix}$$

We assume that $R \sim Normal_N(\mu, \Sigma)$

where $$f(R) = \frac{1}{(2 \cdot \pi)^{N/2} \cdot Det(\Sigma)^{1/2}} \cdot e^{\{-\frac{1}{2} \cdot (R-\mu)' \cdot \Sigma^{-1} \cdot (R-\mu)\}}$$

Let also $w = (w_1, w_2, ..., w_N)'$ be a N-vector of **weights** that sum to one, that is

$$x'1 = \sum_{i=1}^{N} x_i = 1$$

where **1** = (1, 1, 1, …, 1)' is a N-vector of ones (also called the **sum vector**). Any collection of individual assets is a **portfolio** of these assets. For simplicity, in what follows we will denote a portfolio by the weight vector **w** that produces it.

A portfolio with weights **w** has a scalar random return $x' \cdot R$ that is distributed as

$R_P = \mathbf{x}' \times \mathbf{R} \sim Normal_1(\mu_P,\ \sigma_P{}^2)$

where $\mu_P \equiv \mathbf{x}' \times \boldsymbol{\mu} = \sum_{i=1}^{N}\sum_{j=1}^{N} x_i \cdot x_j \cdot \sigma_{ij} = \sum_{i=1}^{N}\sum_{j=1}^{N} x_i \cdot x_j \cdot \rho_{ij} \cdot \sigma_i \cdot \sigma_j$

The minimum variance portfolio with expected return μ_P is the solution of the problem

$$\begin{aligned} \min_{w} \quad & \frac{1}{2}\sigma_P{}^2 = \frac{1}{2}\mathbf{x}' \times \boldsymbol{\Sigma} \times \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x}'\mathbf{1} = 1 \\ & \mathbf{x}'\boldsymbol{\mu} = \mu_P \end{aligned} \qquad (1)$$

A portfolio w* is **Mean-Variance Efficient** if there exists no portfolio **w** that has an equal or higher return and a lower variance, i.e., if there exists no w such that

$$\mathbf{x}' \cdot \boldsymbol{\mu} \geq \mathbf{x}^{*\prime} \cdot \boldsymbol{\mu} \ \ and \ \ \mathbf{x}' \cdot \boldsymbol{\Sigma} \cdot \mathbf{x} < \mathbf{x}^{*\prime} \cdot \boldsymbol{\Sigma} \cdot \mathbf{x}^*$$

3.1 The MV optimal solutions

The problem in (1) minimizes a quadratic function subject to linear constraints and under the condition that the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{R}}$ is strictly positive-definite, this minimization problem has a unique solution that may be computed by an analytical way. To obtain this solution we form the Lagrangian of the problem

$$\min_{x,\lambda_1,\lambda_2} \quad L = \frac{1}{2}\mathbf{x}'\boldsymbol{\Sigma}\mathbf{x} + \lambda_1 \cdot (1 - \mathbf{x}'\mathbf{1}) + \lambda_2 \cdot (\mu_P - \mathbf{x}'\boldsymbol{\mu})$$

where λ_1 and λ_2 are Lagrange multipliers. The first-order conditions are

$$\frac{\partial L}{\partial \mathbf{x}} = \mathbf{\Sigma x} - \lambda_1 \cdot \mathbf{1} - \lambda_2 \cdot \boldsymbol{\mu} = 0 \qquad (2)$$

$$\frac{\partial L}{\partial \lambda_1} = 1 - \mathbf{x'1} = 0 \qquad (3)$$

$$\frac{\partial L}{\partial \lambda_2} = \mu_P - \mathbf{x'}\boldsymbol{\mu} = 0 \qquad (4)$$

From the equation (2) we take,

$$\mathbf{x}^*(\mu_P) = \lambda_1 \mathbf{\Sigma}^{-1}\mathbf{1} + \lambda_2 \mathbf{\Sigma}^{-1}\boldsymbol{\mu} \qquad \mathbf{(5)}$$

and substituting into (3) and (4) we get

$$\lambda_1 \mathbf{1'\Sigma}^{-1}\mathbf{1} + \lambda_2 \boldsymbol{\mu}'\mathbf{\Sigma}^{-1}\mathbf{1} = 1$$

$$\lambda_1 \mathbf{1'\Sigma}^{-1}\boldsymbol{\mu} + \lambda_2 \boldsymbol{\mu}'\mathbf{\Sigma}^{-1}\boldsymbol{\mu} = \mu_P$$

Solving now this system for λ_1 and λ_2 we get

$$\lambda_1 = \frac{C - B\cdot\mu_P}{D} \quad \text{and } \lambda_2 = \frac{A\cdot\mu_P - B}{D}$$

where

$$A = \mathbf{1'\Sigma}^{-1}\mathbf{1} > 0$$

$$B = \mathbf{1'\Sigma}^{-1}\boldsymbol{\mu} > 0$$

$$C = \boldsymbol{\mu}'\mathbf{\Sigma}^{-1}\boldsymbol{\mu} > 0$$

$$D = AC - B^2 > 0$$

The variance of the portfolio $x^*(\mu_P)$ in the equation (5) is:

$$
\begin{aligned}
\sigma_P^2(\mu_P) &= \mathbf{x}^*(\mu_P)' \cdot \mathbf{\Sigma} \cdot \mathbf{x}^*(\mu_P) \\
&= \mathbf{x}^*(\mu_P)' \cdot \mathbf{\Sigma} \cdot (\lambda_1 \cdot \mathbf{\Sigma}^{-1} \cdot 1 + \lambda_2 \cdot \mathbf{\Sigma}^{-1} \cdot \boldsymbol{\mu}) \\
&= \lambda_1 \cdot \mathbf{x}^*(\mu_P)' \cdot 1 + \lambda_2 \cdot \mathbf{x}^*(\mu_P)' \cdot \boldsymbol{\mu} \\
&= \lambda_1 + \lambda_2 \cdot \mu_P^2 \\
&= \frac{A \cdot \mu_P^2 - 2 \cdot B \cdot \mu_P + C}{D}
\end{aligned}
\qquad (6)
$$

where, to obtain the next to last line of the display above, we have used the constraints that the portfolio weights sum to one and the return of the portfolio is μ_P. Equation (6) is recognized as the equation of a **parabola**, so the set of Mean-Variance Efficient Portfolios is a parabola in Mean-Variance space. If we take the first and second derivatives of (6) with respect to μ_P,

$\dfrac{d\sigma_P^2(\mu_P)}{d\mu_P} = \dfrac{2 \cdot (A \cdot \mu_P - B)}{D}$ that can take possitive, zero or negative

$$\frac{d^2\sigma_P^2(\mu_P)}{d{\mu_P}^2} = \frac{2 \cdot A}{D} > 0$$

shows that $\sigma_P^2(\mu_P)$ is a strictly convex function of μ_P, with a unique minimum at

$$\frac{d\sigma_P^2(\mu_P)}{d\mu_P} = 0 \Leftrightarrow \frac{2 \cdot (A \cdot \mu_P - B)}{D} = 0 \Leftrightarrow \mu_P = \frac{B}{A}$$

It is more usual to represent the frontier in Mean-Standard Deviation space instead of Mean-Variance space. After this, we have:

$$\sigma_P(\mu_P) = \sqrt{\frac{A \cdot \mu_P^2 - 2 \cdot B \cdot \mu_P + C}{D}} \qquad (7)$$

so that,

$$\frac{d\sigma_P(\mu_P)}{d\mu_P} = \frac{A \cdot \mu_P - B}{D \cdot \sigma_P} \qquad \textbf{(8)}$$

$$\frac{d^2\sigma_P(\mu_P)}{d\mu_P^2} = \frac{1}{D \cdot \sigma_P^3} > 0$$

We see from equation (8) that σ is also a strictly convex function of μ_P and that the Minimum-Standard Deviation portfolio is the same as the Minimum-Variance portfolio (which is attained at $\mu_P = \frac{B}{A}$). Equation (7) has the form of the equation of a **hyperbola**, so the set of Mean-Variance Efficient Portfolios describe a hyperbola in Mean-Standard Deviation space.

Figure 2 graphs this frontier. For any given σ_P there are two feasible portfolios, one of which has a higher return. Since no rational investor would prefer the portfolio with the low return when, with the same risk, another portfolio with a higher return is feasible, the **Efficient Portfolio Frontier** is the upper part of the hyperbola. The equation for μ_P as a function of σ_P is

$$\mu_P = \frac{B}{A} \pm \frac{1}{A}\sqrt{D(A \cdot \sigma_P^2 - 1)} \quad (9)$$

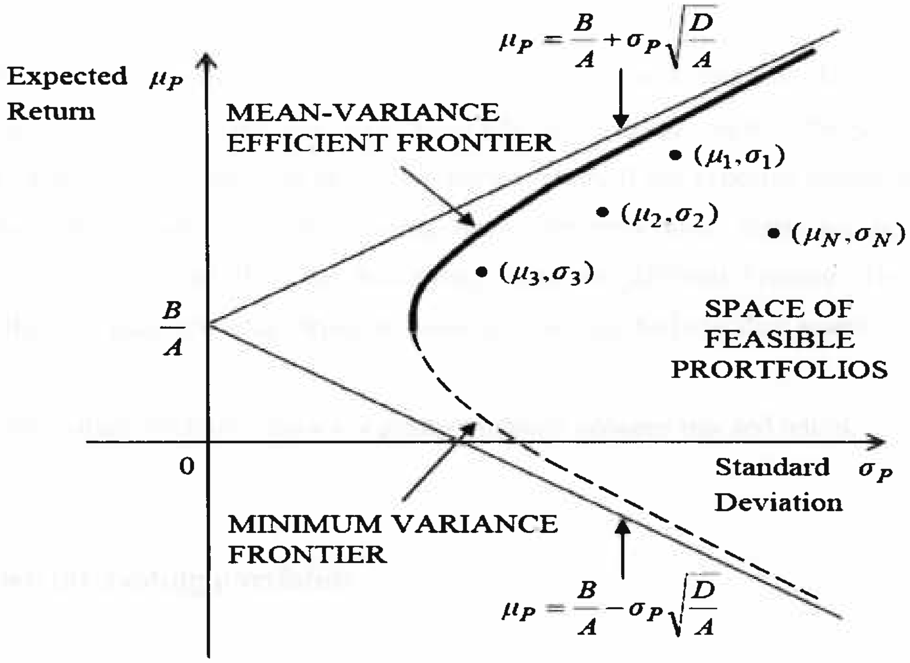


Figure 2 Mean-Standard Deviation Frontier: Risky Assets Only

Figure 2 also graphs the asymptotes of the Efficient Frontier which are given by

$$\mu_P = \frac{B}{A} \pm \sigma_P \cdot \sqrt{\frac{D}{A}}$$

The returns and standard deviations of individual assets $(\mu_{R_i}, \sigma_{R_i})$, $i = 1, \ldots, N$ belongs to the **Space of Feasible Portfolios.**

Two important things are:

1) The meaning of Diversification in portfolio assets

Nobody rational investor do not want to have only one individual asset because if he has a collection of assets (portfolio) can obtain a better expected return for the same level of risk, or less risk for the same level of expected return. If the expected returns for the individual assets have covariance among them, the individuals assets are no more Minimum Variance and they are positioning inner the Efficient Frontier. Thus, the Diversification takes advantage from the covariance among the individual assets.

2) For diversified portfolios, there is a positive relation between risk and return.

3.2 Two interesting portfolios

Every point on the Efficient Frontier (the upper part of the hyperbola) corresponds to a portfolio that is optimal in the sense that no other portfolio can yield a higher return for a given risk, or a lower risk for a given return. In this section we wish to describe the properties of the Efficient Frontier by studying two very interesting portfolios on it: *the minimum-variance portfolio* and the *tangency portfolio*.

The **minimum-variance portfolio** is the portfolio that solves the equation (1) without a mean constraint. This portfolio is given by

$$x_{\min} = \frac{\mathbf{\Sigma}^{-1} \times \mathbf{1}}{\mathbf{1}' \times \mathbf{\Sigma}^{-1} \times \mathbf{1}}$$

Now consider rays through the origin, like the line OS_1. The slope of such a ray may be thought as the tradeoff between return and risk. For example, the ray OS_1 has a small slope and therefore the portfolios along this ray will have large risk relative to their return. On the other hand, a ray like OS_3 is much steeper and therefore the tradeoff between risk and return is much more attractive. Continuing like this, we see that the 'best' such ray is the vertical axis itself, since along the vertical axis one has a positive return with no risk at all. The vertical axis, however, is not feasible, since the only feasible portfolios are points inside the hyperbola. This analysis leads us to choose the portfolio that is feasible and corresponds to a line with the maximum possible slope. The ray with the maximum possible slope that is feasible is OS_2, and since this line is tangent to the Efficient Frontier, the corresponding portfolio is called the **tangency portfolio**.

The tangency portfolio is given by

$$x_{\tan} = \frac{\mathbf{\Sigma}^{-1} \times \boldsymbol{\mu}}{\mathbf{1}' \times \mathbf{\Sigma}^{-1} \times \boldsymbol{\mu}}$$

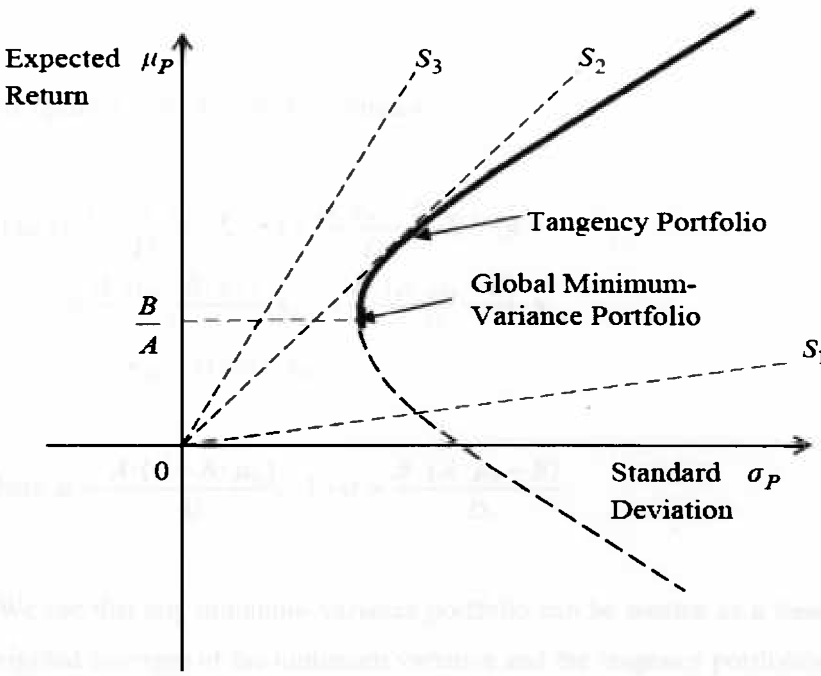


Figure 3 The Minimum Variance and Tangency Portfolios.

At this point is very important to define the Sharpe Ratio and to give an alternative prospective of the tangency portfolio.

$SR = Sharpe\ Ratio = \frac{\mu_P}{\sigma_P}$ where μ_P is the mean return of the portfolio P for the specific time period and σ_P is the standard deviation of returns of the portfolio P for the specific time period.

This measure of risk calculates the tradeoff between the return of the specific portfolio for each unit of the total risk of the portfolio. If our portfolio has big value of Sharpe Ratio means that it has good enough return for this specific time period.

The tangency portfolio can view as an outcome of solving the following problem:

$$\max_x SR = \max_x \frac{\mu_P}{\sigma_P}$$

Any optimal portfolio can be written as

$$\begin{aligned} x_*(\mu_P) &= \frac{C - B\cdot\mu_P}{D}\cdot\boldsymbol{\Sigma}^{-1}\times\mathbf{I} + \frac{A\cdot\mu_P - B}{D}\cdot\boldsymbol{\Sigma}^{-1}\times\boldsymbol{\mu} \\ &= \frac{A\cdot(C - B\cdot\mu_P)}{D}\cdot\mathbf{x}_{\mathbf{min}} + \frac{B\cdot(A\cdot\mu_P - B)}{D}\cdot\mathbf{x}_{\mathbf{tan}} \\ &= \alpha\cdot\mathbf{x}_{\mathbf{min}} + (1-\alpha)\cdot\mathbf{x}_{\mathbf{tan}}, \end{aligned}$$

where $\alpha = \frac{A\cdot(C - B\cdot\mu_P)}{D}$, $1-\alpha = \frac{B\cdot(A\cdot\mu_P - B)}{D}$.

We see that any minimum-variance portfolio can be written as a linear combination (a weighted average) of the minimum variance and the tangency portfolios. In fact, any two distinct minimum-variance portfolios will serve in place of $x_{\min}$ and $x_{\tan}$.

3.3 Utility Maximization and Portfolio Choice

We consider derived utility functions of the form

$$V(\mu,\sigma^2) = \mu - \theta\cdot(\sigma^2 + \mu^2),$$

where θ is a constant.

The above derived utility function is derived from a quadratic utility function U(W), without the returns R of our assets must be distributed normal.

For risk averse investors, this function should be increasing in μ and decreasing in σ^2, so that

$$\frac{dV}{d\mu} > 0 \Leftrightarrow 1-2\cdot\theta\cdot\mu > 0 \Leftrightarrow \theta < \frac{1}{2\cdot\mu},$$

and

$$\frac{d^2V}{d\sigma^2} < 0 \Leftrightarrow -\theta < 0 \Leftrightarrow \theta > 0.$$

For θ in the risk averse range ($0 < \theta < \frac{1}{2\cdot\mu}$), the indifference curves in mean-variance space are lines with upward slope

$$\frac{d\mu}{d\sigma^2} = \frac{\theta}{1-2\cdot\theta\cdot\mu} > 0. \quad (10)$$

In order to obtain this slope, we take the differential of the utility function

$dV = d\mu - \theta\cdot d\sigma^2 - 2\cdot\theta\cdot\mu\cdot d\mu$ and we set it equal to zero and solve for $\frac{d\mu}{d\sigma^2}$.

The steepness of the indifference curves depends on θ, with individuals that are more risk averse possessing higher θ and steeper indifference curves.

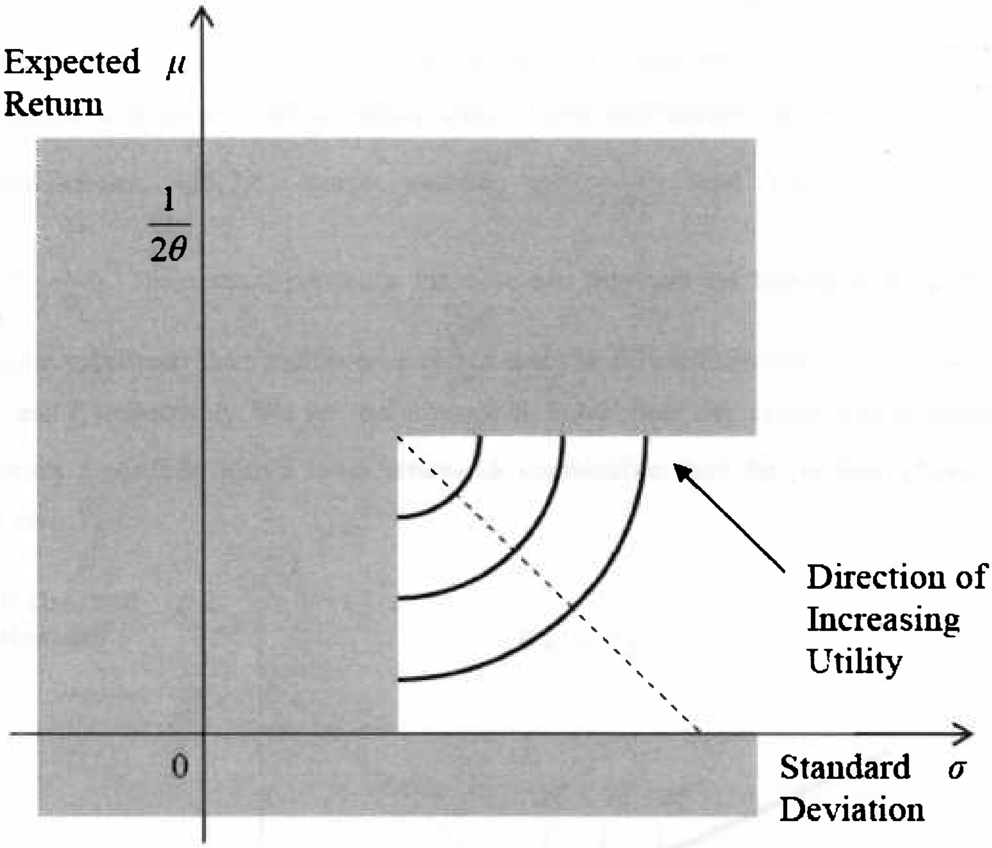


Figure 4 The indifference curves in *mean-variance space*

The indifference curves in mean-standard deviation space are concentric quarter-circles with center at the point (0, 1/2θ) and upward slope

$$\frac{d\mu}{d\sigma} = \frac{2\cdot\theta\cdot\sigma}{1-2\cdot\theta\cdot\mu} > 0.$$

The total differential in terms of μ and σ, instead of μ and σ^2, is

$dV = d\mu - 2\cdot\theta\cdot\sigma\cdot d\sigma - 2\cdot\theta\cdot\mu\cdot d\mu$

Setting this equal to zero and solving for $\frac{d\mu}{d\sigma}$ we obtain the above expression.

At this moment, consider two investors X and Y, and assume that X is more risk averse than Y, thus $\theta_X > \theta_Y$. In Figure 5 we can see the indifference curves of these two investors in mean-standard deviation space. These indifference curves are concentric quarter-circles, with X's curves centered at $(0, \frac{1}{2\cdot\theta_X})$, and Y's curves centered at $(0, \frac{1}{2\cdot\theta_Y})$. The optimal portfolios for these two investors are located at the point of tangency between their indifference curves and the Efficient Frontier, and are labeled P_X and P_Y respectively. We see that investor X, being more risk averse than investor Y, chooses a portfolio with a lower return-risk combination than the portfolio chosen by investor Y.

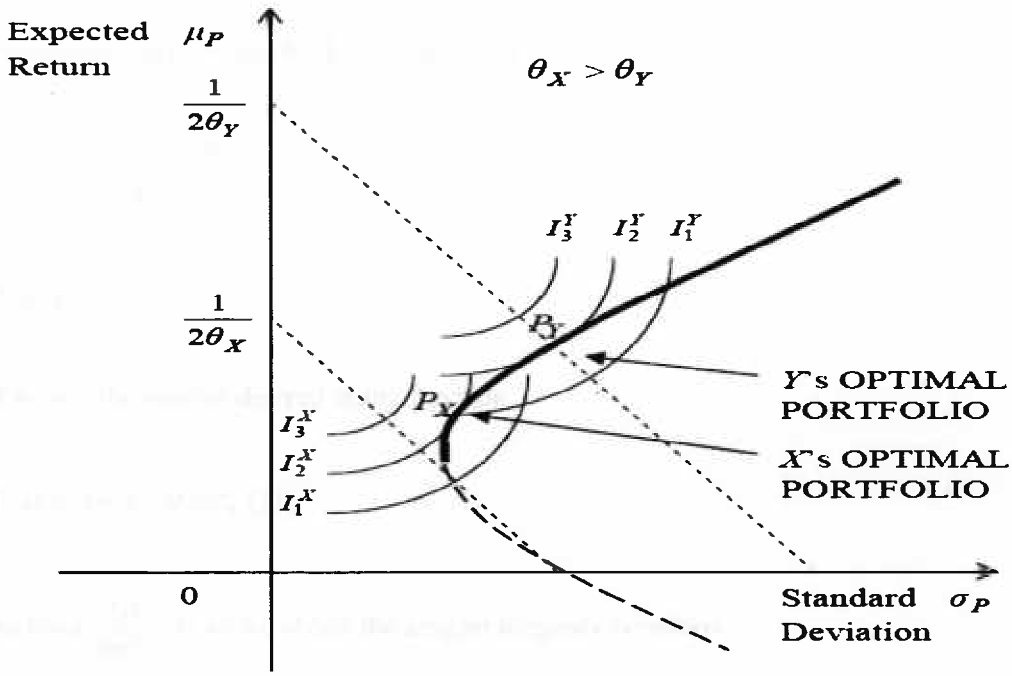


Figure 5 Optimal portfolios for investors X, Y

It is clear from figure – that there is a one-to-one mapping between points (μ, σ) on the efficient frontier and the risk aversion parameter θ. To derive this mapping, we equate the slope of the efficient frontier

$$\frac{d\mu}{d\sigma^2}=\frac{D}{2\cdot\sqrt{D\cdot(A\cdot\sigma^2-1)}}$$

with the slope of the indifference curves in the relation (10) to get

$$\frac{\theta}{1-2\cdot\theta\cdot\mu(\sigma^2)}=\frac{D}{2\cdot\sqrt{D\cdot(A\cdot\sigma^2-1)}}.$$

Substituting (9) into the above equation and solving for σ^2 we get

$$\sigma_*^2(\theta)=\frac{4\cdot A\cdot(A+2\cdot D+C\cdot D)\cdot\theta^2-4\cdot A\cdot B\cdot D\cdot\theta+A^2\cdot D}{4\cdot A\cdot(A+D)^2\cdot\theta^2}$$

Substituting this solution back into (9) we get

$$\mu_*(\theta)=\frac{D+2\cdot B\cdot\theta}{2\cdot(A+D)\cdot\theta}$$

Note 1:

If we use the simpler derived utility function

$V(\mu,\sigma^2)=\mu-\theta\cdot\sigma^2$, (11)

we have $\frac{d\mu}{d\sigma^2}=\theta$, so we obtain the simpler tangency condition

$$\theta = \frac{D}{2 \cdot \sqrt{D \cdot (A \cdot \sigma^2 - 1)}}$$

Solving for σ^2 we obtain

$\sigma_*^2(\theta) = \frac{1}{A} + \frac{D}{4 \cdot A \cdot \theta^2}$, and substituting into (9) we get $\mu_*(\theta) = \frac{B}{A} + \frac{D}{2 \cdot A \cdot \theta}$

The utility function in (11) may be derived from an exponential von-Neumann-Morgenstern utility function under normality of the asset returns.

Note 2:

The definition of the Arrow-Pratt absolute risk aversion coefficient A(W) is:

$$A(W) = -\frac{U''(W)}{U'(W)}$$

which is a measure of risk aversion. Since U(.) is concave, this coefficient is a function of wealth level and implies that risk aversion decreases as the level of wealth increases. This may reflect individuals' attitude to take more risk when they are more financially secure. The stronger the concaveness the greater the risk aversion is. The term in the denominator is to normalize the coefficient. With this normalization *A(W)* is the same for all equivalent utility functions. The absolute coefficient of risk aversion determines how much wealth an investor will put in a risky investment in absolute terms.

3.4 Example

Consider four risky assets, the returns of which are distributed as

$$\begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \end{bmatrix} \sim N_4\left(\begin{bmatrix} 0.01 \\ 0.03 \\ 0.07 \\ 0.12 \end{bmatrix}, \begin{bmatrix} 0.0016 & 0.0017 & 0.0006 & 0.0004 \\ 0.0017 & 0.0049 & 0.0026 & 0.0021 \\ 0.0006 & 0.0026 & 0.0225 & 0.0090 \\ 0.0004 & 0.0021 & 0.0090 & 0.0400 \end{bmatrix}\right).$$

To be more specific, consider the situation in which the four risky assets are Treasury Bills, Bonds, Large Cap Shares and Small Cap Shares with expected returns 1%, 3%, 7%, 12% and Standard Deviations of 4%, 7%, 15% and 20% respectively.

Asset Data	Expected Return	Standard Deviation
Tbills	1,0%	4,0%
Bonds	3,0%	7,0%
LCShares	7,0%	15,0%
SCShares	12,0%	20,0%

Table 1 Expected returns and Risk for the four asset classes

In this example, I will try to give an analytical solution for the Mean-Variance optimal portfolios.

As a first step, we must calculate the inverse matrix of Σ, for this calculation we use the MATLAB program and we take:

$$\Sigma^{-1} = \begin{bmatrix} 995.4244 & -354.2741 & 12.0172 & 5.9413 \\ -354.2741 & 344.9683 & -27.0203 & -8.4885 \\ 12.0172 & -27.0203 & 51.3483 & -10.2550 \\ 5.9413 & -8.4885 & -10.2550 & 27.6936 \end{bmatrix}$$

After we compute the qualities:

$$A = \mathbf{1}' \times \mathbf{\Sigma}^{-1} \times \mathbf{1} = 655.2758$$
$$B = \mathbf{1}' \times \mathbf{\Sigma}^{-1} \times \boldsymbol{\mu} = 8.8599$$
$$C = \boldsymbol{\mu}' \times \mathbf{\Sigma}^{-1} \times \boldsymbol{\mu} = 0.5320$$
$$D = A \cdot C - B^2 = 270.1352$$

Therefore the equation of the Efficient Frontier in Mean-Standard Deviation space is

$$\begin{aligned} \mu_P &= \frac{B}{A} \pm \frac{1}{A}\sqrt{D(A \cdot \sigma_P{}^2 - 1)} \\ &= \frac{8.8599}{655.2758} \pm \frac{1}{655.2758} \cdot \sqrt{270.1352 \cdot (655.2758 \cdot \sigma_P{}^2 - 1)} \\ &= 0.0135 \pm \frac{1}{655.2758} \cdot \sqrt{177013.0593 \cdot \sigma_P{}^2 - 270.1352} \end{aligned}$$

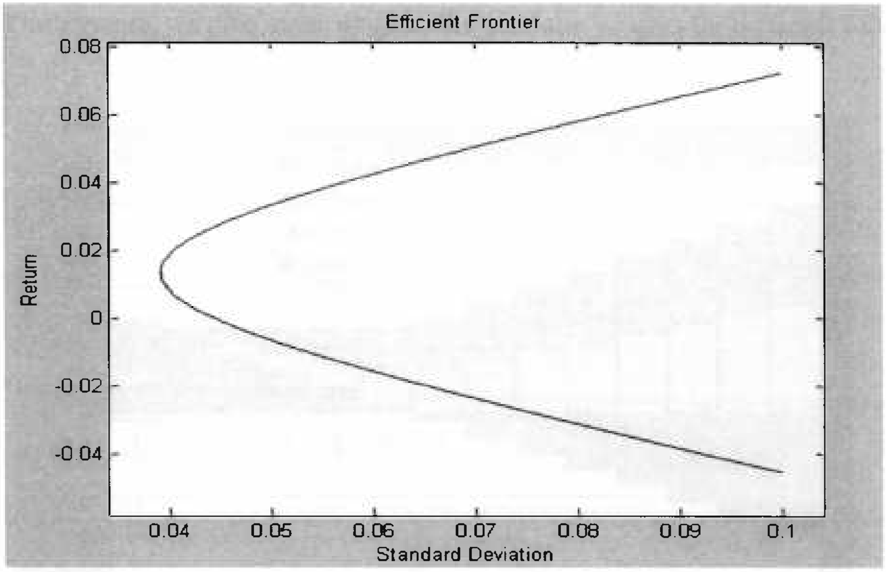


Figure 6 Efficient Frontier for the four risky assets

The minimum standard-deviation portfolio is attained at

$$\mu_P = \frac{B}{A} = \frac{8.8599}{655.2758} = 0.01352 = 1.352\%$$

which yields a minimum standard deviation of

$$\sigma_P = \sqrt{\frac{C - \frac{B^2}{A}}{D}} = \sqrt{\frac{0.532 - \frac{8.8599^2}{655.2758}}{270.1352}} = 0.03906$$

Finally, the asymptotes of the hyperbola are

$$\mu_P = \frac{B}{A} \pm \sigma_P \cdot \sqrt{\frac{D}{A}} = \frac{8.8599}{655.2758} \pm \sigma_P \cdot \sqrt{\frac{270.1352}{655.2758}} = 0.0135 \pm \sigma_P \cdot 0.6421$$

Furthermore, we give in the diagram the portfolio weights for different values of the $R_{\exp}$:

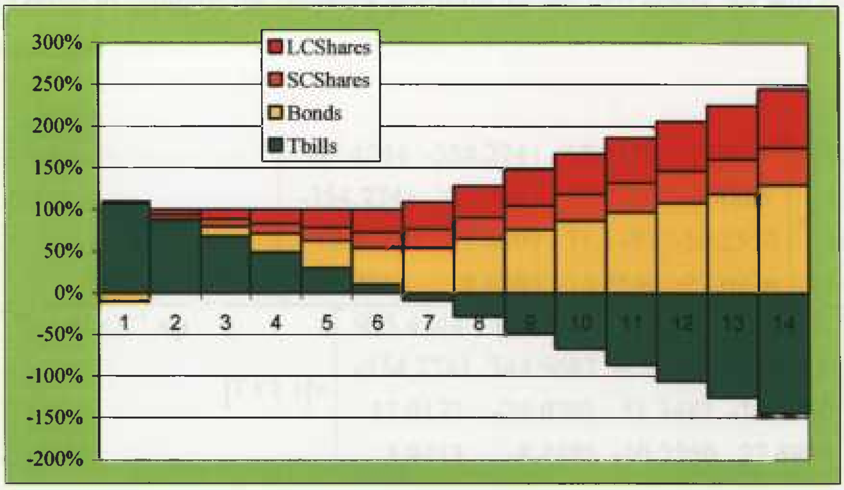


Figure 7 Portfolio weights for different values of $R_{\exp}$

At this point, we will calculate the above two interesting portfolios, *the minimum-variance portfolio* and the *tangency portfolio*.

The minimum-variance portfolio

In order to compute the minimum-variance portfolio for our four assets, we will use the following equation:

$$x_{\min} = \frac{\Sigma^{-1} \times \mathbf{1}}{\mathbf{1}' \times \Sigma^{-1} \times \mathbf{1}} = \frac{\begin{bmatrix} 995.4244 & -354.2741 & 12.0172 & 5.9413 \\ -354.2741 & 344.9683 & -27.0203 & -8.4885 \\ 12.0172 & -27.0203 & 51.3483 & -10.2550 \\ 5.9413 & -8.4885 & -10.2550 & 27.6936 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}}{[1\,1\,1\,1] \times \begin{bmatrix} 995.4244 & -354.2741 & 12.0172 & 5.9413 \\ -354.2741 & 344.9683 & -27.0203 & -8.4885 \\ 12.0172 & -27.0203 & 51.3483 & -10.2550 \\ 5.9413 & -8.4885 & -10.2550 & 27.6936 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}} = \begin{bmatrix} 1.0058 \\ -0.0684 \\ 0.0398 \\ 0.0227 \end{bmatrix}$$

The tangency portfolio

In order to compute the tangency portfolio for our four assets, we will use the following equation:

$$x_{\tan} = \frac{\boldsymbol{\Sigma}^{-1} \times \boldsymbol{\mu}}{\mathbf{1}' \times \boldsymbol{\Sigma}^{-1} \times \boldsymbol{\mu}} = \frac{\begin{bmatrix} 995.4244 & -354.2741 & 12.0172 & 5.9413 \\ -354.2741 & 344.9683 & -27.0203 & -8.4885 \\ 12.0172 & -27.0203 & 51.3483 & -10.2550 \\ 5.9413 & -8.4885 & -10.2550 & 27.6936 \end{bmatrix} \times \begin{bmatrix} 0.01 \\ 0.03 \\ 0.07 \\ 0.12 \end{bmatrix}}{\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 995.4244 & -354.2741 & 12.0172 & 5.9413 \\ -354.2741 & 344.9683 & -27.0203 & -8.4885 \\ 12.0172 & -27.0203 & 51.3483 & -10.2550 \\ 5.9413 & -8.4885 & -10.2550 & 27.6936 \end{bmatrix} \times \begin{bmatrix} 0.01 \\ 0.03 \\ 0.07 \\ 0.12 \end{bmatrix}} = \begin{bmatrix} 0.0993 \\ 0.4398 \\ 0.1889 \\ 0.2720 \end{bmatrix}$$

The expected return of the tangency portfolio is:

$$\mu_{\tan} = x_{\tan}' \times \mu = \begin{bmatrix} 0.0993 & 0.4398 & 0.1889 & 0.2720 \end{bmatrix} \times \begin{bmatrix} 0.01 \\ 0.03 \\ 0.07 \\ 0.12 \end{bmatrix} = 0.0601 = 6.01\%$$

and its standard deviation is:

$$\sigma_{\tan} = \sqrt{x_{\tan}' \times \Sigma \times x_{\tan}} = 0.0823$$

Continuing, we will find the optimal mean standard deviation combination for two values of risk aversion parameter θ (θ = 4 and θ = 1) for two investors.

We use the following two equations, which consider the following derived utility function $V(\mu,\sigma^2) = \mu - \theta \cdot (\sigma^2 + \mu^2)$:

$$\mu_*(\theta)=\frac{D+2\cdot B\cdot\theta}{2\cdot(A+D)\cdot\theta} \text{ and } \sigma_*^2(\theta)=\frac{4\cdot A\cdot(A+2\cdot D+C\cdot D)\cdot\theta^2-4\cdot A\cdot B\cdot D\cdot\theta+A^2\cdot D}{4\cdot A\cdot(A+D)^2\cdot\theta^2}$$

- For θ=4:

the optimal mean-standard deviation combination is

$$(\mu,\sigma)=(0.0461,\ 0.0640)$$

As next step, we try to compute the portfolio that yields this optimal combination for our investor.

The weights of this portfolio are:

$$x^*(0.0461)=\lambda_1\cdot\mathbf{\Sigma}^{-1}\times\mathbf{I}+\lambda_2\cdot\mathbf{\Sigma}^{-1}\times\boldsymbol{\mu}$$

We must compute the λ_1 and λ_2:

$$\lambda_1=\frac{C-B\cdot\mu_P}{D}=\frac{0.5320-8.8599\cdot0.0461}{270.1352}=0.000457$$

$$\lambda_2=\frac{A\cdot\mu_P-B}{D}=\frac{655.2758\cdot0.0461-8.8599}{270.1352}=0.079$$

Thus

$$\begin{aligned}x^*(0.0461)&=\lambda_1\cdot\mathbf{\Sigma}^{-1}\times\mathbf{I}+\lambda_2\cdot\mathbf{\Sigma}^{-1}\times\boldsymbol{\mu}\\
&=0.000457\cdot\begin{bmatrix}995.4244 & -354.2741 & 12.0172 & 5.9413\\ -354.2741 & 344.9683 & -27.0203 & -8.4885\\ 12.0172 & -27.0203 & 51.3483 & -10.2550\\ 5.9413 & -8.4885 & -10.2550 & 27.6936\end{bmatrix}\times\begin{bmatrix}1\\1\\1\\1\end{bmatrix}+\\
&+0.079\cdot\begin{bmatrix}995.4244 & -354.2741 & 12.0172 & 5.9413\\ -354.2741 & 344.9683 & -27.0203 & -8.4885\\ 12.0172 & -27.0203 & 51.3483 & -10.2550\\ 5.9413 & -8.4885 & -10.2550 & 27.6936\end{bmatrix}\times\begin{bmatrix}0.01\\0.03\\0.07\\0.12\end{bmatrix}\end{aligned}$$

Finally, $x^{*}(0.0461)=\begin{bmatrix}0.3707\\0.2873\\0.1441\\0.1972\end{bmatrix}$

- For θ=1:

The optimal mean-standard deviation combination is $(\mu,\sigma)=(0.1555,\ 0.2246)$

Now, we give the process of computation of the portfolio that yields this optimal combination between risk and return for this investor with risk aversion parameter θ=1.

The weights of this portfolio are:

$$x^{*}(0.1555)=\lambda_1\cdot\mathbf{\Sigma}^{-1}\times\mathbf{I}+\lambda_2\cdot\mathbf{\Sigma}^{-1}\times\boldsymbol{\mu}$$

We must compute the λ_1 and λ_2:

$$\lambda_1=\frac{C-B\cdot\mu_P}{D}=\frac{0.5320-8.8599\cdot0.1555}{270.1352}=-0.00313$$

$$\lambda_2=\frac{A\cdot\mu_P-B}{D}=\frac{655.2758\cdot0.1555-8.8599}{270.1352}=0.3444$$

Thus $x^{*}(0.1555)=\lambda_1\cdot\mathbf{\Sigma}^{-1}\times\mathbf{I}+\lambda_2\cdot\mathbf{\Sigma}^{-1}\times\boldsymbol{\mu}=\begin{bmatrix}-1.7599\\1.4821\\0.4946\\0.7834\end{bmatrix}$

We clearly see that the investor with risk aversion parameter θ=4, being more risk averse than the investor with risk aversion parameter θ=1, because the first one chooses a portfolio with a lower return-risk combination than the portfolio chosen by the second investor. This result is summarized in the next table:

	Risk aversion parameter θ	Combination of return-standard deviation (μ, σ)	Portfolio weights
Investor 1	4	(0.0461, 0.0640)	x1 = 37.07% x2 = 28.73% x3 = 14.41% x4 = 19.72%
Investor 2	1	(0.1555, 0.2246)	x1 = -175.99% x2 = 148.21% x3 = 49.46% x4 = 78.34%

Table 2 Portfolio weights for different values of risk aversion parameter (θ)

Now, we use the derived utility function $V(\mu,\sigma^2)=\mu-\theta\cdot\sigma^2$, and we repeat our previous calculations.

Using the following mathematical equations:

$$\sigma_*^2(\theta)=\frac{1}{A}+\frac{D}{4\cdot A\cdot\theta^2},\quad \mu_*(\theta)=\frac{B}{A}+\frac{D}{2\cdot A\cdot\theta}$$

- For θ=4:

the optimal mean-standard deviation combination is $(\mu,\sigma)=(0.0651,\ 0.0893)$

As next step, we try to compute the portfolio that yields this optimal combination for this investor.

The weights of this portfolio are:

$$x^*(0.0651) = \lambda_1 \cdot \mathbf{\Sigma}^{-1} \times \mathbf{I} + \lambda_2 \cdot \mathbf{\Sigma}^{-1} \times \boldsymbol{\mu}$$

$$\lambda_1 = \frac{C - B \cdot \mu_P}{D} = \frac{0.5320 - 8.8599 \cdot 0.0651}{270.1352} = -0.000166$$

$$\lambda_2 = \frac{A \cdot \mu_P - B}{D} = \frac{655.2758 \cdot 0.0651 - 8.8599}{270.1352} = 0.125$$

Thus $x^*(0.0651) = \lambda_1 \cdot \mathbf{\Sigma}^{-1} \times \mathbf{I} + \lambda_2 \cdot \mathbf{\Sigma}^{-1} \times \boldsymbol{\mu} = \begin{bmatrix} 0.0006 \\ 0.4945 \\ 0.2048 \\ 0.2988 \end{bmatrix}$

- For θ=1:

The optimal mean-standard deviation combination is $(\mu, \sigma) = (0.2196,\ 0.3234)$

We must compute the λ_1 and λ_2:

$$\lambda_1 = \frac{C - B \cdot \mu_P}{D} = \frac{0.5320 - 8.8599 \cdot 0.2196}{270.1352} = -0.00523$$

$$\lambda_2 = \frac{A \cdot \mu_P - B}{D} = \frac{655.2758 \cdot 0.2196 - 8.8599}{270.1352} = 0.4999$$

Thus $x^*(0.2196) = \lambda_1 \cdot \mathbf{\Sigma}^{-1} \times \mathbf{I} + \lambda_2 \cdot \mathbf{\Sigma}^{-1} \times \boldsymbol{\mu} = \begin{bmatrix} -3.0071 \\ 2.1821 \\ 0.7001 \\ 1.1269 \end{bmatrix}$

3.5 Extensions of the basic model

In spite of its theoretical interest, the basic mean–variance model is often too simplistic to represent the complexity of real-world portfolio selection problems. In order to enrich the model, we need to introduce more realistic constraints.

Consider the following portfolio selection problem

$$
\begin{array}{lll}
\min & \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}\cdot x_i \cdot x_j & \text{objective function} \\
\text{s.t.} & \sum_{i=1}^{n} R_i \cdot x_i = R_{\exp} & \text{return constraint} \\
 & \sum_{i=1}^{n} x_i = 1 & \text{budget constraint} \\
 & \underline{x}_i \le x_i \le \overline{x}_i \quad (1 \le i \le n) & \text{floor and ceiling constraints} \\
 & x_i = x_i^{(0)} \text{ or } x_i \ge (x_i^{(0)} + \underline{B}_i) & \\
 & \text{or } x_i \le (x_i^{(0)} - \underline{S}_i) \quad (1 \le i \le n) & \text{trading constraints} \\
 & |\{i \in \{1,\dots,n\} : x_i \ne 0\}| \le N & \text{maximum number of assets}
\end{array}
$$

- Return and budget constraints.
  The first constraint expresses the requirement placed on expected return. The second constraint, called budget constraint, requires that 100% of the budget be invested in the portfolio.
- Floor and ceiling constraints.
  These constraints define lower and upper limits on the proportion of each asset which can be held in the portfolio. They may model institutional restrictions on the composition of the portfolio. They may also rule out negligible holdings of asset in the portfolio, thus making its control easier.
  Notice that the floor constraints generalize the non-negativity constraints imposed in the original model.

- Trading constraints.
  Lower limits on the variations of the holdings can also be imposed in order to reflect the fact that, typically, an investor may not be able, or may not want, to modify the portfolio by buying or selling tiny quantities of assets. A first reason may be that the contracts must bear on significant volumes.
- Another reason may be the existence of relatively high fixed costs linked to the transactions. These constraints are disjunctive in nature: for each asset i either the holdings are not changed, or a minimal quantity $\underline{B}_i$ must be bought, or a minimal quantity $\underline{S}_i$ must be sold.
- Maximum number of assets.
  This constraint limits to N the number of assets included in the portfolio, e.g. in order to facilitate its management.

3.6 Solution approaches

The complexity of solving portfolio selection problems is highly related to the type of constraints that they are involved in our model.

The simplest case is obtained when the non-negativity constraints are omitted (allowed short sales) from the basic model. In this case, a closed-form solution is easily obtained by the classical Lagrange methods, for which we are giving a detailed description.

The problem becomes more complex when non-negativity constraints (not allowed short sales) are added to the formulation, as in the basic Markowitz model. The resulting is a quadratic programming problem, however, can still be solved efficiently with the use of the Solver from the Microsoft Excel program. A very crucial point is that the problem becomes increasingly hard to manage and to solve as the number of assets increases.

When the model involves constraints on minimal trading quantities or on the maximum number of assets in the portfolio, as in our model (PS), then we enter the field of mixed integer nonlinear programming and classical algorithms are typically unable to deliver the optimal solution of this problem.

In this thesis, our purpose is to investigate the solution of the complete model (PS) presented in Chapter 5 by a simulated annealing algorithm. Our goal is to develop an approach which, while giving up claims to optimality, would display some robustness with respect to various criteria, including

- quality of solutions
- speed
- ease of addition of new constraints
- ease of modification of the objective function (e.g. when incorporating higher moments than the variance)

Chapter 4

Heuristic Optimization

4.1 The problems in Optimization Problems

Optimization problems are concerned with finding the values for one or several decision variables that meet the objective(s) the best without violating the constraint(s). The identification of an efficient portfolio in the basic Markowitz model is therefore a typical optimization problem: the values for the decision variables x_i have to be found under the constraints that

(i) they must not exceed certain bounds: $0 \leq x_i \leq 1$ and: $\sum_{i=1}^{N} x_i = 1$ and

(ii) the portfolio return must have a given expected value, the objective is to find values for the assets' weights that minimize the risk which is computed in a predefined way.

The basic Markowitz model is a well-defined optimization model as the relationship between weight structure and risk and return is perfectly computable for any valid set of (exogenously determined) parameters for the assets' expected returns and (co-)variances (as well as, when applicable, the trade-off factor between portfolio risk and return).

On the other hand, if in the Portfolio Selection problem we add the constraint for only long sales ($x_i \geq 0$) there exists no general solution for this optimization problem because of the non-negativity constraint on the asset weights. Hence, there is no closed form solution whereas, in the basic unconstrained Markowitz model. Though not solvable analytically, there exist numerical procedures by which the Markowitz model can be solved for a given set of parameters values.

Depending on the objective function, optimization problems might have multiple solutions some of which might be local optima. In Figure 2.1, e.g., a function $f(x)$ is depicted, and the objective might be to find the value for x where $f(x)$ reaches its highest value, i.e. $\max_x f(x)$. As can easily be seen, all three points x_A, x_B and x_C are (local) maxima: the first order condition, $f'(x) = 0$ is satisfied (indicated by the horizontal tangency lines), and any slight increase or decrease of x would decrease the function's value: $f(x) \geq f(x \pm \varepsilon) | \varepsilon \to 0$. Nonetheless, only x_B is a *global optimum* as it yields the highest overall value for the objective function, whereas x_A and x_C are just *local optima*. Unlike for this simple example, however, it is often difficult to determine whether an identified solution is a local or the global optimum as the solution space is too complex: All of the objective functions that will be considered in the main part of this contribution have more than one decision variable, the problem space is therefore multidimensional; and the objective functions are mostly discontinuous (i.e., the first derivatives are not well behaved or do not even exist).

In portfolio management, these difficulties with the objective functions are frequently observed when market frictions have to be considered. To find solutions anyway, common ways of dealing with them would be to either eliminate these frictions (leading to models that represent the real-world in a stylized and simplified way) or to approach them with inappropriate methods (which might lead to suboptimal and misleading results without being able to recognize these errors). This contribution is mainly concerned with the effects of market frictions on financial management which are therefore explicitly taken into account. Hence, for reliable results, an alternative class of optimization techniques has to be employed that are capable of dealing with these frictions, namely *heuristic optimization* techniques.

4.2 Heuristic optimization techniques

The central common feature of all *heuristic optimization* (*HO*) methods is that they start off with a more or less arbitrary initial solution, iteratively produce new solutions by some generation rule and evaluate these new solutions, and eventually report the best solution found during the search process. The execution of the iterated search procedure is usually halted when there has been no further improvement over a given number of iterations, either when the found solution is good enough or the allowed CPU time (or other external limit) has been reached or when some internal parameter terminates the algorithm's execution. Another obvious halting condition would be exhaustion of valid candidate solutions – a case hardly ever realized in practice.

4.3 Simulated Annealing

Kirkpatrick, Gelatt, and Vecchi (1983) present one of the simplest and most general Heuristic Optimization (HO) techniques which turned out to be one of the most efficient ones, too: *Simulated Annealing* (*SA*). This algorithm mimics the crystallization process during cooling or annealing: When the material is hot, the particles have high kinetic energy and move more or less randomly regardless of their and the other particles' positions.

The cooler the material gets, however, the more the particles are "torn" towards the direction that minimizes the energy balance. The SA algorithm does the same when searching for the optimal values for the decision parameters: It repeatedly suggests random modifications to the current solution, but progressively keeps only those that improve the current situation.

SA applies a probabilistic rule to decide whether the new solution replaces the current one or not. This rule considers the change in the objective function and an equivalent to "temperature" (reflecting the progress in the iterations).

Simulated annealing is a general name for a class of optimization heuristics that perform a stochastic neighborhood search of the solution space. The major advantage of SA over classical local search methods is its ability to avoid getting trapped in local minima while searching for a global minimum. The underlying idea of the heuristic is adopted from certain thermo dynamical processes (cooling of a melted solid).

The general problem takes the form

$$\min \; F(x) \text{ s.t. } x \in X,$$

the basic principle of the SA heuristic can be described as follows. Starting from a current solution x, another solution y is generated by taking a stochastic step in some neighborhood of x. If this new proposal improves the value of the objective function, then y replaces x as the new current solution. Otherwise, the new solution y is accepted with a probability that decreases with the magnitude of the deterioration and in the course of iterations. (Notice the difference with classical descent approaches, where only improving moves are allowed and the algorithm may end up quickly in a local optimum.)

More precisely, the generic simulated annealing algorithm performs the following steps.

- Choose an initial solution $x^{(0)}$ and compute the value of the objective function $F(x^{(0)})$. Initialize the best available solution, denoted by (x^*, F^*), as: $(x^*, F^*) \leftarrow (x^{(0)}, F(x^{(0)}))$.
- Until a stopping criterion is fulfilled and for n starting from 0, do:

  1) Draw a solution x at random in the neighborhood $V(x^{(n)})$ of $x^{(n)}$

  2) If $F(x) \leq F(x^{(n)})$ then $x^{(n+1)} \leftarrow x$ and if $F(x) \leq F^*$ then $(x^*, F^*) \leftarrow (x, F(x))$.

  3) If $F(x) > F(x^{(n)})$ then draw a number p at random in $[0,1]$ and if $p \leq p(n, x, x^{(n)})$ then $x^{(n+1)} \leftarrow x$ else $x^{(n+1)} \leftarrow x^{(n)}$.

The function $p(n, x, x^{(n)})$ is often taken to be a Boltzmann function inspired from thermodynamics models:

$$p(n, x, x^{(n)}) = e^{-\frac{\Delta F_n}{T_n}}$$

where $\Delta F_n = F(x) - F(x^{(n)})$ and T_n is the temperature at step n, that is a non-increasing function of the iteration counter n. In so-called geometric cooling schedules, the temperature is kept unchanged during each successive stage, where a stage consists of a constant number L of consecutive iterations. After each stage, the temperature is multiplied by a constant factor $a \in (0,1)$.

Due to the generality of the concepts that it involves, Simulated Annealing (SA) can be applied to a wide range of optimization problems. In particular, no specific requirements need to be imposed neither on the objective function (derivability, convexity …) nor on the solution space. Moreover, it can be shown that this metaheuristic method converges asymptotically to a global minimum.

For the whole analysis, we give the pseudo code of the Simulated Annealing algorithm:

```
generate random valid solution x;
REPEAT
   generate new solution x_ by randomly modifying
      the current solution x;
   evaluate new solution x_;
IF acceptance criterion is met THEN;
     replace x with x_;
END;
  adjust acceptance criterion;
UNTIL halting criterion is met;
```

Chapter 5

Simulated Annealing for portfolio selection

5.1 Simulated Annealing for portfolio selection

5.1.1 Generalities: How to handle constraints . . .

In order to apply Simulated Annealing (SA) in the Portfolio Selection problem, we must defined two important meanings, the meaning of ***solution*** and ***neighborhood***.

We simply encode a solution of (PS) as an n-dimensional vector x, where each variable x_i represents the holdings of asset i in the portfolio. The quality of a solution is measured by the variance of the portfolio, that is $x' \cdot \Sigma \cdot x$.

After the question of how do we handle the constraints is very fundamental for our thesis and we have to answer the question of how do we make sure that the final solution produced by the SA algorithm satisfies all the constraints of Portfolio Selection problem.

The first and most obvious approach enforces feasibility throughout all iterations of the SA algorithm and forbids the consideration of any solution violating the constraints. This implies that the neighborhood of a current solution must entirely consist of feasible solutions. A second approach, by contrast, allows the consideration of infeasible solutions ***but*** adds a penalty term to the objective function for each violated constraint: the larger the violation of the constraint, the larger the increase in the value of the objective function. A portfolio which is unacceptable for the investor must be penalized enough to be rejected by the minimization process.

The ''all-feasible'' vs. ''penalty'' debate is classical in the optimization literature. Both approaches, however, are not equally convenient in all situations and making the ''right'' choice is a very difficult problem because all constraints are not the same and we ought to be very careful. Before we get to this discussion, let us first line up the respective advantages and inconvenients of each approach.

When the method of penalties is used, the magnitude of each penalty should depend on the magnitude of the violation of the corresponding constraint, but must also be scaled relatively to the variance of the portfolio. A possible expression for the penalties is

$$\alpha \times |\, violation\,|^{P},$$

where α and p are scaling factors. For example, the violation of the return constraint can be represented by the difference between the required portfolio return (R_{exp}) and the current solution return ($R' \cdot x$).The violation of the floor constraint for asset i can be expressed as the difference between the minimum admissible level $\underline{x}_i$ and the current holdings x_i, when this difference is positive.

This method, however, has two vital disadvantages. The first is that it searches a solution space whose size may be considerably larger than the size of the feasible region. This process may require many iterations and takes important computation time.

The second inconvenient begins from the scaling factors: it may be difficult to define adequate values for a and p. If these values are too small, then the penalties do not play their expected role and the final solution may be infeasible. On the other hand, if a and p are too large, then the term $x' \cdot \Sigma \cdot x$ becomes negligible with respect to the penalty, thus, small variations of x can lead to large variations of the penalty term, which overlap the effect of the variance term.

Clearly, the correct choice of a and p depends on the scale of the data. It appears very difficult to automate this choice. In our implementations, we have selected values for a and p as follows. We set $a = 10^{16}$ and $p = 2$ in our algorithm.

Let us now discuss the alternative method, the all-feasible approach, in which the neighborhood of the current solution may only contain solutions that satisfy the given subset of constraints. The idea that we implemented here (following some of the proposals made in the literature on stochastic global optimization) is to draw a direction at random and to take a small step in this direction away from the current solution. The important features of such a move are that both its direction and length are computed so as to respect the constraints. Moreover, the holdings of only a few assets are changed during the move, meaning that the feasible direction is chosen in a low-dimensional subspace. This simplifies computations and provides an immediate translation of the concept of ''neighbor''.

The main advantage of this approach is that no time is lost investigating infeasible solutions. The main disadvantage is that it is not always easy to select a neighbor in this way, so that the resulting moves may be quite contrived, their computation may be expensive and the search process may become inflexible. On the other hand, this approach seems to be the only reasonable one for certain constraints, like for example the trading constraints.

For each class of constraints, we had to point out the advantages and disadvantages of each approach. When a constraint must be strictly satisfied or when it is possible to enforce it efficiently without penalties in the objective function, then we do so. This is the case for the constraints on budget, return and maximum number of assets. A mixed approach is used for the trading, floor, ceiling constraints.

In the next sections, we successively consider each class of constraints, starting with those that are enforced without penalties.

5.1.2 Budget and Return constraints

The budget constraint must be strictly satisfied, since its unique goal is to norm the solution. Therefore, it is difficult to implement this constraint throughout the method of penalties.

The same conclusion applies to the return constraint, albeit for different reasons. Indeed, our aim is to compute the whole mean–variance frontier. To achieve this aim, we want to let the expected portfolio return vary uniformly in its feasible range and to determine the optimal risk associated with each return. In order to obtain meaningful results, the optimal portfolio computed by the procedure should have the exact required return.

As a logical flow of the above comments, we decided to restrict our algorithm to the consideration of solutions that ***strictly satisfy*** the return and the budget constraints. More precisely, given a portfolio x, the neighborhood of x contains all solutions x' with the following property: there exist three assets, labeled 1, 2 and 3 without loss of generality, such that

$$
\begin{cases}
x_1' = x_1 - step, \\
x_2' = x_2 + step \cdot \dfrac{R_1 - R_3}{R_2 - R_3}, \\
x_3' = x_3 + step \cdot \dfrac{R_2 - R_1}{R_2 - R_3}, \\
x_i' = x_i, \quad \text{for all } i > 3,
\end{cases}
$$

where step is a (small) number to be further specified below. It is straightforward to check that x' satisfies the return and budget constraints when x does so. Geometrically, all neighbors x' of the form (4) lie on a line passing through x and whose direction is defined by the intersection of the 3-dimensional subspace associated to assets 1, 2 and 3 with the two hyper planes associated to the budget constraint and the return constraint, respectively. Thus, the choice of three assets determines the direction of the move, while the value of step determines its amplitude.

Observe that, in order to start the local search procedure, it is easy to compute an initial solution which satisfies the budget and return constraints. Indeed, if x denotes an arbitrary portfolio and min (respective max) is the subscript of the asset with minimum (respective maximum) expected return, then a feasible solution is obtained upon replacing $x_{\min}$ and $x_{\max}$ by $x_{\min}'$ and $x_{\max}'$, where

$$
\left\{
\begin{array}{l}
x_{\min}' = [R_{\exp} - \dfrac{\sum\limits_{i \neq \min,\max}^{n} x_i \cdot R_i - (x_{\min} + x_{\max}) \cdot R_{\max}]}{(R_{\min} - R_{\max})}, \\
x_{\max}' = x_{\min} + x_{\max} - x_{\min}'.
\end{array}
\right\}
$$

The resulting solution may violate some of the additional constraints of the problem (trading, turnover, etc.) and penalties will need to be introduced in order to cope with this difficulty. This point will be discussed in next sections.

5.1.3 Direction of moves

Choosing a neighbor of x, as described by (4), involves choosing the direction of the move, i.e. choosing three assets whose holdings are to be modified. For starting, we simply drew the indices of these assets randomly and uniformly over $\{1,\ldots,n\}$. Many of the corresponding moves, however, were not improved our results, thus creating slowness convergence of the algorithm.

We have been able to improve this situation by guiding the choice of the three assets to be modified. Observe that the assets whose return is closest to the required portfolio return have (intuitively) a higher probability to appear in the optimal portfolio than the remaining ones. (This is most obvious for portfolios with 'extreme' returns: consider for example the case where we impose nonnegative holdings and we want to achieve the highest possible return, i.e. Rmax.). To account for this phenomenon, we initially sort all the assets by no decreasing return. For each required portfolio return Rexp, we determine the asset whose return is closest to Rexp and we store its position, say q, in the sorted list.

For each iteration of the SA algorithm, we choose the first asset to be modified by computing a random number normally distributed with mean q and with standard deviation large enough to cover the entire list: this random number points to the position of the first asset in the ordered list. The second and third assets are then chosen uniformly at random.

5.1.4 Maximum number of assets constraint

This cardinality constraint is difficult in nature. A 'natural' penalty approach based on measuring the extent of the violation:

$$violation = |\{i \in \{1,\ldots,n\} : x_i \neq 0\}| - n$$

In our thesis we choose this approach to handle this constraint indeed, all the neighbors of a solution are likely to yield the same penalty, except when an asset exceptionally appears in or disappears from the portfolio.

5.1.5 Floor, ceiling and turnover constraints

The floor, ceiling and turnover constraints are similar to each other, since each of them simply defines **a minimum or maximum bound on holdings**. Therefore, our program automatically converts all turnover purchase constraints into ceiling constraints and all turnover sales constraints into floor constraints.

Suppose now that we know which three assets (say, 1, 2 and 3) must be modified at the current move from solution x to solution x'. Then, it is easy to determine conditions on the value of step such that x' satisfies the floor and ceiling constraints. As a next step we can combine the latter constraints with Eqs. (4) and we take the following conditions:

$$\begin{cases} x_1 - \overline{x}_1 \leq step \leq x_1 - \underline{x}_1, \\ \underline{x}_2 - x_2 \leq step \cdot \dfrac{R_1 - R_3}{R_2 - R_3} \leq \overline{x}_2 - x_2, \\ \underline{x}_3 - x_3 \leq step \cdot \dfrac{R_2 - R_1}{R_2 - R_3} \leq \overline{x}_3 - x_3. \end{cases}$$

These conditions yield a feasible interval of variation for step for the move from x to x'. Let us now consider the case where the feasibility interval [lb; ub] is either empty or very narrow, meaning that x is either infeasible or close to the infeasible region. In order to handle this and other situations where infeasible solutions arise, we introduce a penalty term of the form

$$\alpha \times | \, violation \, |^{P},$$

in the objective function for each ceiling or floor constraint. Notice that the penalty approach appears to be suitable here, since limited violations of the floor, ceiling or turnover constraints can usually be tolerated in practice.

To be more specific, the penalties for floor and ceiling constraints take the form of:

Ceiling if $x_i > \overline{x}_i$, then $penalty = a \cdot (x_i - \overline{x}_i)^{P}$

Floor $\qquad$ if $x_i < \underline{x}_i$, then $penalty = a \cdot (\underline{x}_i - x_i)^P$

5.1.6 Trading constraints

The trading constraints are disjunctive: either the holdings of each asset remain at their current value $x^{(0)}$ or they are modified by a minimum admissible amount. These constraints are difficult to handle, as they disconnect the solution space into 3^n feasible subregions separated by forbidden subsets.

Observe, however, that solutions which violate the trading constraints still arise in some iterations of the algorithm. For instance, the initial solution is usually infeasible, and so are the solutions which are generated when the portfolio contains exactly N assets. The infeasible portfolios are penalized with the way that is described in the next paragraph (the parameters a and p are fixed). Observe that penalties are high at the center of the forbidden zones and decrease in the direction of admissible boundaries (associated with no trading or with minimum sales/purchases). Therefore, starting from a forbidden portfolio, the process tends to favor moves toward feasible regions.

To be more specific, the penalties for trading constraints take the form of:

Purchase

$$Q_{purchase} = x_i - x_i^{(0)};\ \text{if } Q_{purchase} \in (0, \underline{B}_i),\ \text{then}$$

$$\{\text{if } Q_{purchase} \le \frac{\underline{B}_i}{2} \text{ then } penalty = a \cdot Q_{purchase}^P$$

$$\text{else } penalty = a \cdot (\underline{B}_i - Q_{purchase})^P\}$$

Sale

$$Q_{sale} = x_i^{(0)} - x_i;\ \text{if } Q_{sale} \in (0, \underline{S}_i),\ \text{then}$$

$$\{\text{if } Q_{purchase} \le \frac{\underline{S}_i}{2} \text{ then } penalty = a \cdot Q_{sale}^P$$

$$\text{else } penalty = a \cdot (\underline{S}_i - Q_{sale})^P\}$$

5.1.7 Neighbor selection

We can summarize as follows the neighbor selection procedure.

Move direction

- If the current portfolio involves N- k assets, with $k \geq 1$, then
  - select three assets, say 1, 2 and 3, at random as explained in above section , while ensuring that at most k of them are outside the current portfolio;
  - go to Case a.
- If the current portfolio involves N assets, then
  - select three assets, say 1, 2 and 3, at random, while ensuring that at most one of them is outside the current portfolio;
  - if all three selected assets are in the current portfolio, then go to Case a; else go to Case b.

Step length

Case a

Let d be the direction of the move as defined by Eqs. (4) (with the sign of step fixed at random). Compute the feasible interval for the step of the move, with the use of the equation

step = (ub-lb) * ranmar() + lb where ub, lb is the upper and lower bound of the step and the function ranmar() gives a random number between [0,1].

- If $x + step \cdot d$ satisfies the trading constraint

  $x + step \cdot d$ is the selected neighbor; if necessary, compute penalties for the violation of the floor and ceiling constraints as we discuss in above section;

Case b

• Let assets 1 and 2 be in the current portfolio and asset 3 be outside. In Equation. (4), set the parameter step equal to x1, set $x_1' = 0$ and compute the corresponding values of x_2' and x_3'.

• If necessary, compute penalties for the violation of the floor, ceiling and trading constraints as in Tables 1 and 2.

5.2 Cooling schedule and stopping criterion

In our implementation of simulated annealing, we have adopted the geometric cooling schedule. In order to describe more completely this cooling schedule, we need to specify the value of the parameters T_0 (the initial temperature), L and a. we set the initial temperature T_0 in such a way that, during the first cooling stage (first L steps), the probability of acceptation of a move is roughly equal to a predetermined level of 0.8. In order to achieve this goal, we proceed as follows. In the first phase, the SA algorithm is run for L steps without rejecting any moves.

After L moves, the temperature is decreased according to the scheme $T_{k+1} = a \cdot T_k$. We use here the standard value $a = 0.85$. The fundamental trade-offs, involved in the determination of the stage length L are well-known, but difficult to quantify precisely. A large value of L allows exploring the solution space thoroughly, but results in long execution times. The algorithm terminates if no moves are accepted during a given number S of successive stages. In our experiments, we used S= 4.

Chapter 6

Results

6.1 The Data

We have used financial data extracted from the web database of Naftemporiki. We have retrieved the daily prices of N=307 Greek stocks covering different traditional sectors for 435 days, from 1 January 2006 to 28 September 2007, in order to estimate their mean returns and covariance matrix.

To be more specific, we choose 20 stocks from the sector of the large Capitalization and from the daily prices we compute the daily returns for the stocks, continuing we calculate the expected returns for these stocks with the use of this specific time period:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} \text{ and after with}$$

$$E(R_i) = \frac{\sum_{t=1}^{435} R_{it}}{T}, i = 1, \ldots, 20$$

After, we compute the variance of these 20 stocks

$$\sigma_i^2 = \sum_{t=1}^{T} \frac{(R_{it} - \bar{R}_i)^2}{T} \text{ and } \sigma_i = \sqrt{\sigma_i^2} \text{ for all } i = 1, \ldots, 20$$

As a last step, we find the co-variances among these 20 stocks, with the use of the relationship:

$$\sigma_{12} = \sum_{t=1}^{T} \frac{(R_{1t} - \bar{R}_1) \cdot (R_{2t} - \bar{R}_2)}{T} \text{ for the assets 1 and 2.}$$

The reader of this master thesis can find these data (expected returns, variances and co-variances) at the Appendix B in the end of thesis.

6.2 Comments on the source code

In this section, I give in a more analytical way some important points of the implementation of our program.

First of all, we declare in the beginning of each function the variables that it's used. Some important variables are:

n : number of assets in our universe (n=20)

$n\max$: maximum number of assets that we want to have our portfolio.

ns : the stage size

$mu(n)$: vector of the expected return for the twenty assets

$sigma(n,n)$: variance-covariance matrix for these 20 assets

f : value of the objective function

$muport$: required expected return

$wopt(n)$: optimal weights of the portfolio

$fopt$: optimal value of the objective function

t : the value of the parameter of initial temperature

$w(n)$: variable for the portfolio weights

$wlow(n)$: minimum value for each weight

$wup(n)$: maximum value for each weight

$w0(n)$: the initial weights for the initial portfolio

$\min tr(n)$: minimum value on the variations of the holdings for each weight

$tradec$: logical value (if tradec=1, the portfolio has trading constraint else if tradec=0, the portfolio has not trading constraint)

As a next step, we initialize the variables $mu(n)$, $sigma(n,n)$ with values that our program was read from the file data2.txt. The initializations of t, $wlow(n)$, $wup(n)$, $w0(n)$, $\min tr(n)$, $tradec$ are become in the main body of our source code.

In a typical way, we set t = 0.10, w0(n) = 0.05 , wlow(n) = -5.0 (in the case that we allowed short sales), wlow(n) = 0.0 (in the case that we not allowed short sales), mintr(n) = 0.10, tradec = 0 or 1.

The main functions of our program:

- init(n,w,mu,muport) : This function finds initial weights for the portfolio. Its main operation is to make zero the weights of assets that cannot contribute in the portfolio expected return. Here we set the restriction to invest our whole budget.

- fcn(n,sigma,w,f,wup,wlow,nmax,w0,mintr,tradec) : Define the function to be optimized (we minimize the objective function $\min \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}\cdot x_i \cdot x_j$)

- pick(n,d,d1,d2,d3) : This function chooses which 3 weights to change at each repeating of the algorithm of Simulated Annealing).

- sa (n, mu, sigma, w, f, wopt, fopt, wlow, wup, nmax, w0, mintr, tradec, t, ns) This routine implements the continuous simulated annealing global optimization algorithm. SA tries to find the global optimum of an N dimensional function. It moves both up and downhill and as the optimization process proceeds, it focuses on the most promising area. To start, it randomly chooses a trial point within the step. The function is evaluated at this trial point and its value is compared to its value at the initial point. In a maximization problem, all uphill moves are accepted and the algorithm continues from that trial point. Downhill moves may be accepted, the decision is made by the Metropolis criteria. It uses the parameter temperature (t) and the size of the downhill move in a probabilistic manner. The smaller temperature and the size of the downhill move are, the more likely that move will be accepted. If the trial is accepted, the algorithm moves on from that point. If it is rejected, another point is chosen instead for the trial evaluation.

A fall in temperature (t) is imposed upon the system with the equation of $t(i+1) = 0.85 \cdot t(i)$ where i is the i-th iteration. Thus, as t declines, downhill moves are less likely to be accepted and the percentage of rejections rises. The Simulated Annealing algorithm focuses upon the most promising area for optimization.

- exprep(rdum) : this function replaces exp to avoid under- and overflows. Note that the maximum and minimum values of exprep are such that they has no effect on the algorithm.

- subroutine rmarin(ij,kl) and function ranmar() : this is the initialization routine for the random number generator ranmar(). Note: the seed variables can have values between: $0 \le ij \le 31328$ and $0 \le kl \le 30081$.

The parameter temperature (t) is crucial in using Simulated Annealing successfully. It influences the step length over which the algorithm searches for optima. For a small initial t, the step length may be to small, thus not enough of the function might be evaluated to find the global optima.

6.3 Computational experiments

The algorithms described above have been implemented in FORTRAN and run on a PC Pentium 3.2 GHz under Windows XP. A graphical interface was developed with Excel and MATLAB. All computation times mentioned in coming sections are approximate real times, not CPU times. Unless otherwise stated, the parameter settings for the basic SA algorithm are defined as follows:

- Stage size: ns=70000
- Stopping criterion: terminate when no moves are accepted for S= 4 consecutive stages

For the purpose of constructing realistic problem instances, we have used financial data extracted from the web database of Naftemporiki. We have retrieved the daily prices of n=307 Greek stocks covering different traditional sectors for 484 days, from 1 January 2006 to 28 September 2007, in order to estimate their mean returns and covariance matrix. But for the needs of my thesis I used the 20 mostly major stocks with large capitalization. (Note that our goal was not to draw any conclusions regarding the firms, or the stock market, or even the composition of optimal portfolios, but only to test the computational performance of our algorithm.)

These data have been used to generate several instances of model (PS) involving different subsets of constraints.

For each instance, we have approximately computed the mean-variance frontier by letting the expected portfolio return (Rexp) vary from -8% to 28% by steps of 1%.

Now, we must give the input parameters to our model:

Input Parameters		
Parameters	Units	Inputs
Return constraint	Fraction	Depended
Budget constraint	Integer	1
Floor constraint	Fraction	Depended
Ceiling constraint	Fraction	Depended
Trading constraint	Fraction	Depended
Number of assets in universe	Integer	20
Cardinality constraint	Integer	Depended

Table 3 Input parameters for our model

6.3.1 The unconstraint case

This is the case in which, we have used the simulated annealing (SA) algorithm to solve instances of the Markowitz mean–variance model without any constraint on the weights of the portfolio. Since these instances can easily be solved to optimality by the Lagrange techniques, we are able to check the quality of the solutions obtained by the SA algorithm. Our algorithm finds the exact optimal risk for all values of the expected return.

In this case the table for the input parameters takes the form:

Input Parameters		
Parameters	Units	Inputs
Return constraint	Fraction	-10%-27%
Budget constraint	Integer	100%
Floor constraint	Fraction	No
Ceiling constraint	Fraction	No
Trading constraint	Fraction	No
Number of assets in universe	Integer	20
Cardinality constraint	Integer	No

Table 4 Input parameters for the unconstraint case

The mean–risk frontier for the case of no restrictions is plotted in Fig. 8.

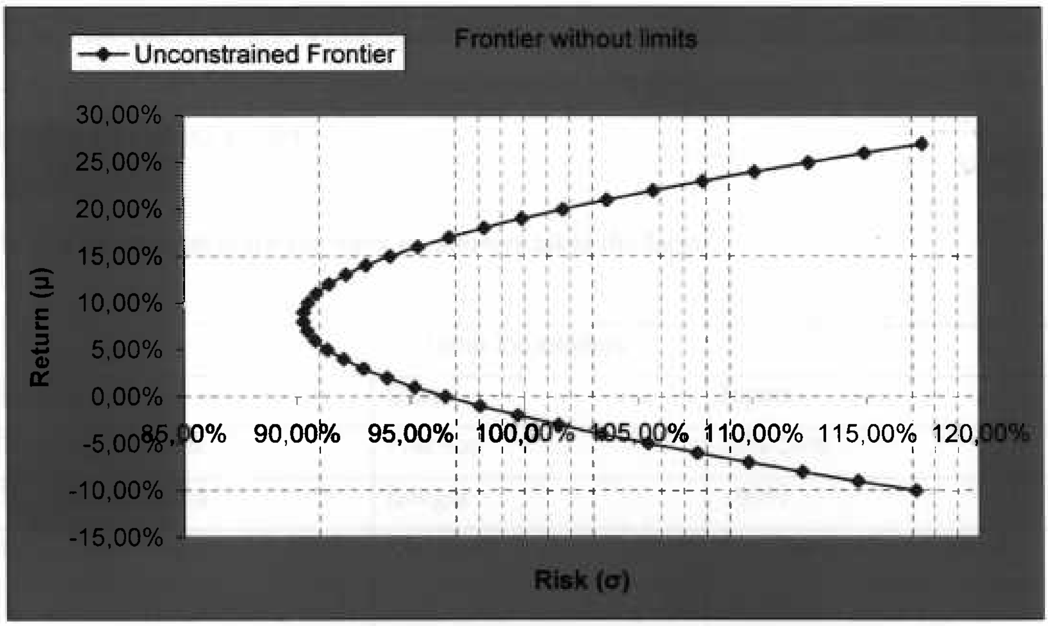


Figure 8 Efficient Frontier without limits

6.3.2 The case of only long sales

This is the case in which, we have used the simulated annealing (SA) algorithm to solve instances of the Markowitz mean–variance model with the no-negativity constraint in the portfolio weights ($x_i \geq 0$).

In this case the table for the input parameters takes the form:

Input Parameters		
Parameters	Units	Inputs
Return constraint	Fraction	-5%-26%
Budget constraint	Integer	100%
Floor constraint	Fraction	Yes (the lower bound of x_i is 0)
Ceiling constraint	Fraction	No
Trading constraint	Fraction	No
Number of assets in universe	Integer	20
Cardinality constraint	Integer	No

Table 5 Input parameters for the only long sales case

The mean–risk frontier for the case of only long sales is plotted in Fig. 9.

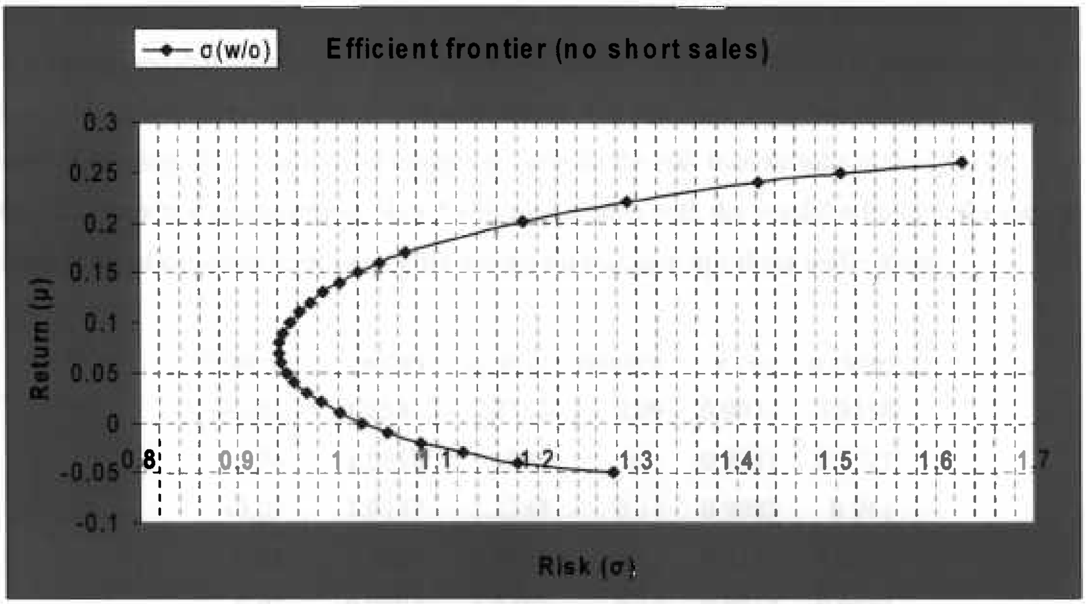


Figure 9 Efficient Frontier with short sales restriction

At this point, it is very important to compare the efficient frontier of the unconstraint case with the only long sales frontier.

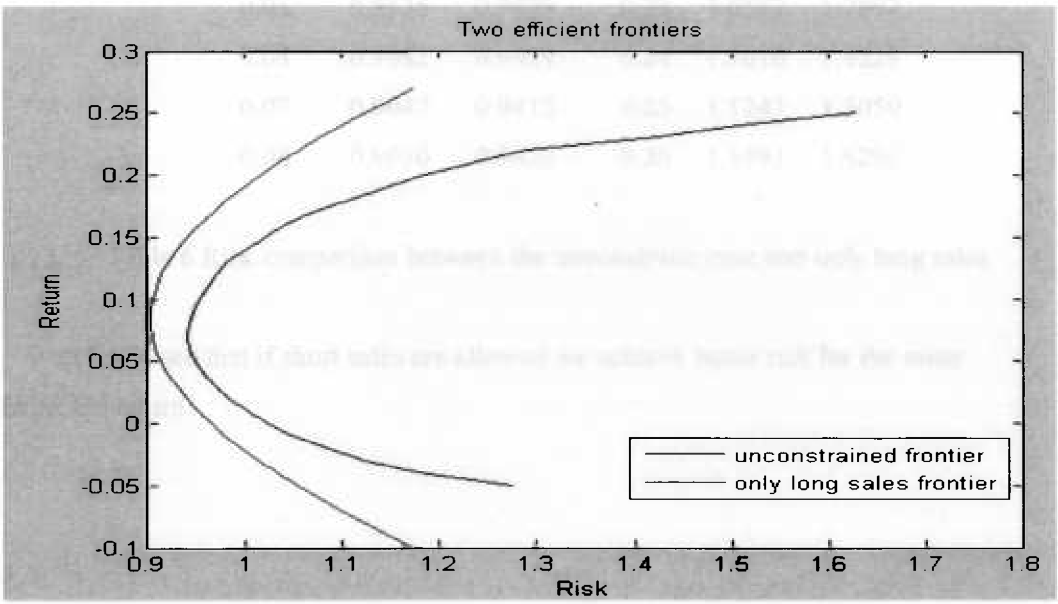


Figure 10 Plot together the efficient frontiers

We clearly see that the effect of short sales restriction is that the Efficient Frontier moves inside the feasible set and it loses extreme values. For the same expected return in the case of no short sales, we have much more risk than the risk that we take in the case of the unconstrained optimization. Now I will give a table with the portfolio holding for the same level of expected return and after a diagram that sum ups these differences:

Return	σ (w)	σ (w/o)	Return	σ (w)	σ (w/o)
-0.05	1.0541	1.2752	0.09	0.9031	0.9458
-0.04	1.0338	1.1798	0.1	0.9051	0.9521
-0.03	1.0147	1.1243	0.11	0.9088	0.961
-0.02	0.9969	1.0829	0.12	0.9143	0.9723
-0.01	0.9804	1.0497	0.13	0.9215	0.9861
0	0.9653	1.0244	0.14	0.9305	1.0022
0.01	0.9518	1.0022	0.15	0.9411	1.0205
0.02	0.9397	0.9842	0.16	0.9534	1.0421
0.03	0.9293	0.9687	0.17	0.9671	1.0688
0.04	0.9206	0.9564	0.2	1.0170	1.1853
0.05	0.9135	0.9479	0.22	1.0567	1.2897
0.06	0.9082	0.9429	0.24	1.1010	1.4229
0.07	0.9047	0.9412	0.25	1.1247	1.5059
0.08	0.9030	0.9422	0.26	1.1493	1.6292

Table 6 Risk comparison between the unconstraint case and only long sales

We clearly see that if short sales are allowed we achieve better risk for the same expected return.

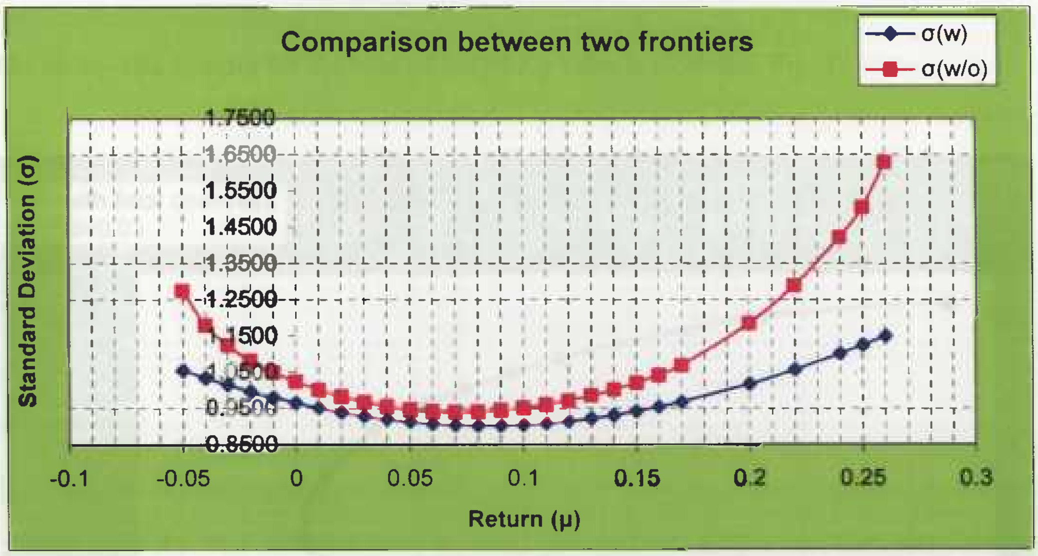


Figure 11 Comparison between two frontiers

6.3.3 The case of floor constraint

This is the case in which, we have used the simulated annealing (SA) algorithm to solve instances of the Markowitz mean–variance model with the floor constraint in the portfolio weights ($\underline{x}_i \geq 0.01$).

In this case the table for the input parameters takes the form:

Input Parameters		
Parameters	Units	Inputs
Return constraint	Fraction	-5%-26%
Budget constraint	Integer	100%
Floor constraint	Fraction	Yes $\underline{x}_i \geq 0.01$
Ceiling constraint	Fraction	No
Trading constraint	Fraction	No
Number of assets in universe	Integer	20
Cardinality constraint	Integer	No

Table 7 Input parameters for the case of floor constraint

The mean–risk frontier for the case of only long sales is plotted in Fig. 12.

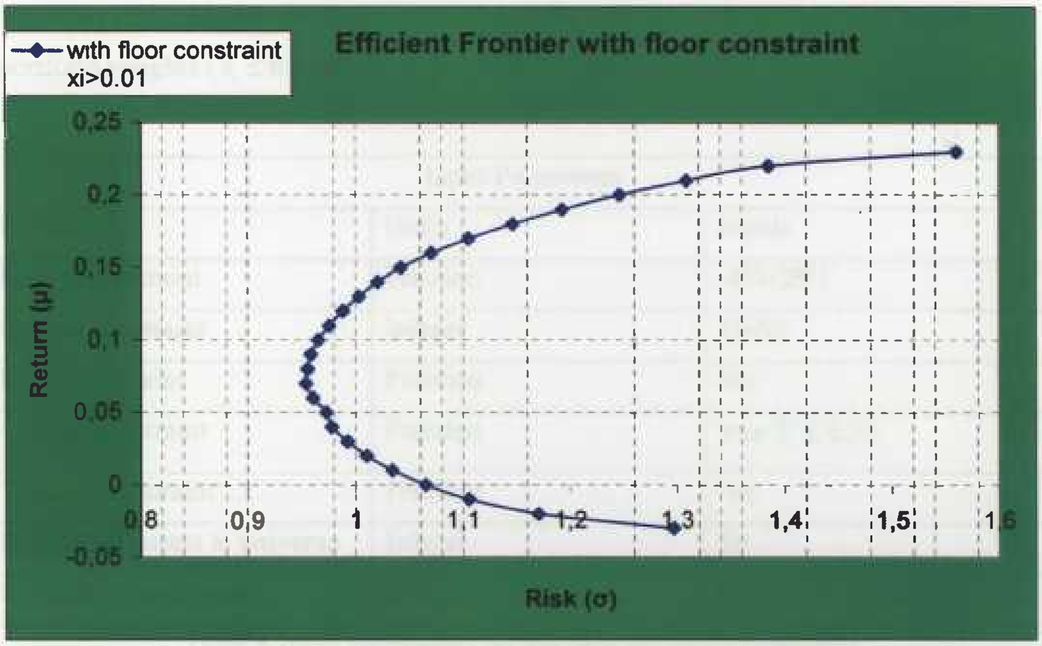


Figure 12 Efficient Frontier with floor constraint

This constraint defines lower limits on the proportion of each asset which can be held in the portfolio. They may model institutional restrictions on the composition of the portfolio. They may also rule out negligible holdings of asset in the portfolio, thus making its control easier.

6.3.4 The case of ceiling constraint

This is the case in which, we have used the simulated annealing (SA) algorithm to solve instances of the Markowitz mean–variance model with the ceiling constraint in the portfolio weights ($\overline{x}_i \leq 0.25$).

Input Parameters		
Parameters	Units	Inputs
Return constraint	Fraction	-8%-28%
Budget constraint	Integer	100%
Floor constraint	Fraction	No
Ceiling constraint	Fraction	Yes $\overline{x}_i \leq 0.25$
Trading constraint	Fraction	No
Number of assets in universe	Integer	20
Cardinality constraint	Integer	No

Table 8 Input parameters for the case of ceiling constraint

The mean–risk frontier for the case of is plotted in Fig. 13.

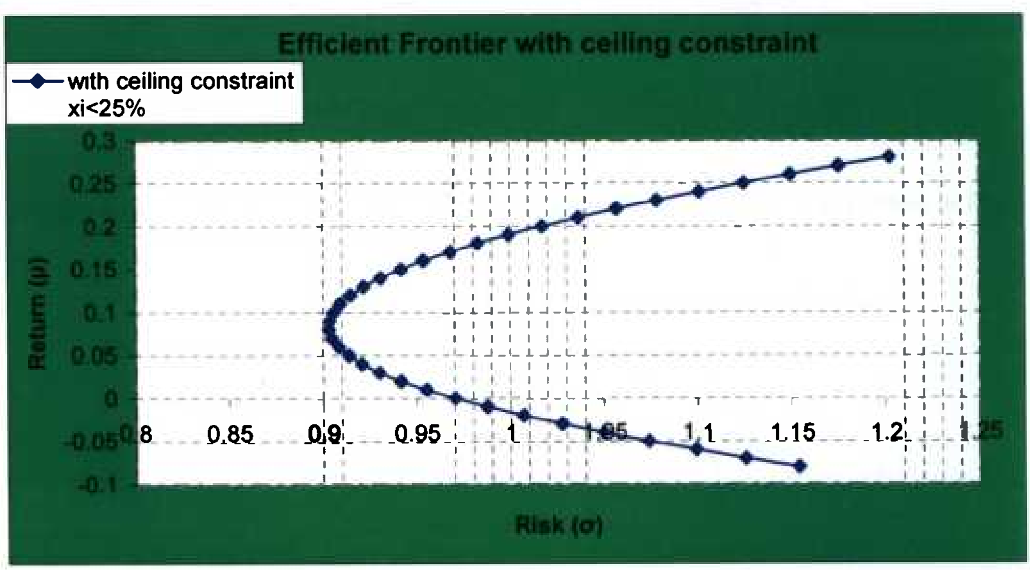


Figure 13 Efficient Frontier with ceiling constraint

6.3.5 The case of cardinality constraint (Maximum number of assets included in our portfolio)

This is the case in which, we have used the simulated annealing (SA) algorithm to solve instances of the Markowitz mean–variance model with the maximum number of assets constraint.

In this case the table for the input parameters takes the form:

Input Parameters		
Parameters	Units	Inputs
Return constraint	Fraction	-6%-26%
Budget constraint	Integer	100%
Floor constraint	Fraction	No
Ceiling constraint	Fraction	No
Trading constraint	Fraction	No
Number of assets in universe	Integer	20
Cardinality constraint	Integer	Yes (Maximum number of assets=5)

Table 9 Input parameters for the case of cardinality constraint

The mean–risk frontier for the case of cardinality constraint $N_{\max} = 5$ is plotted in Fig. 14.

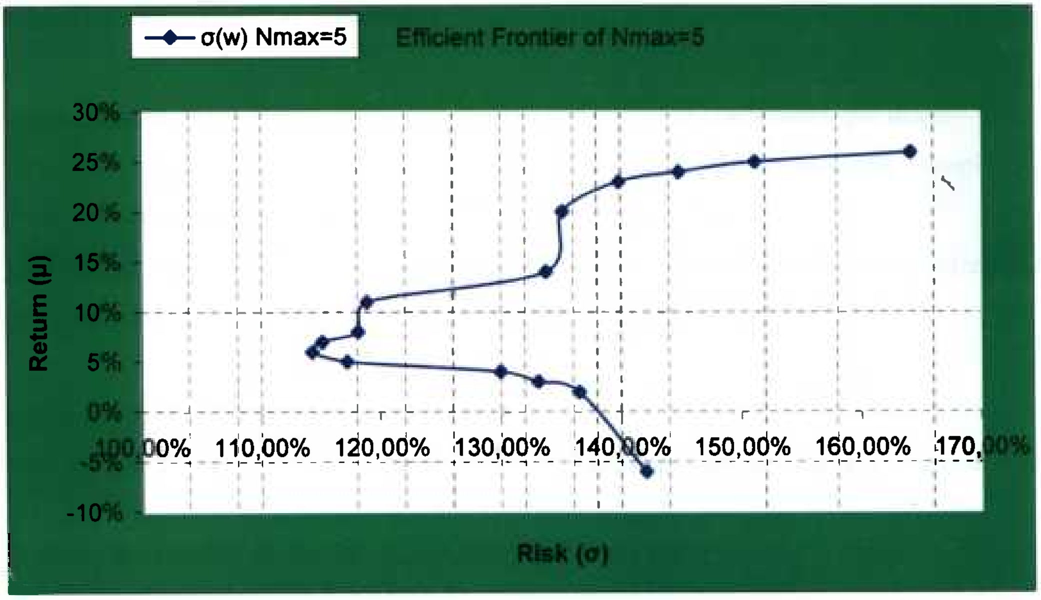


Figure 14 Efficient Frontier with cardinality constraint

This constraint limits to five the number of assets included in the portfolio, e.g. in order to facilitate its management.

6.3.6 The case of trading constraint

When the model only involves floor, ceiling and turnover constraints, the mean–variance frontiers are smooth curves. When we introduce trading constraints, however, sharp discontinuities may arise.

When the number of securities increases, the optimization problem becomes extremely difficult to solve. The trading constraints defined as follows:

- $\underline{B}_i = \underline{S}_i = 0.10 \ (i = 1,\ldots,n)$
- the initial portfolio $x^{(0)}$ is the best portfolio of 20 stocks with an expected return of 9.26%

In this case the table for the input parameters takes the form:

Input Parameters		
Parameters	Units	Inputs
Return constraint	Fraction	-4%-24%
Budget constraint	Integer	100%
Floor constraint	Fraction	No
Ceiling constraint	Fraction	No
Trading constraint	Fraction	Yes($\underline{B}_i = \underline{S}_i = 0.10 \ (i = 1,\ldots,n)$
Number of assets in universe	Integer	20
Cardinality constraint	Integer	No

Table 10 Input parameters for the case of trading constraint

The mean–risk frontier for the case of trading constraint is plotted in Fig. 15.

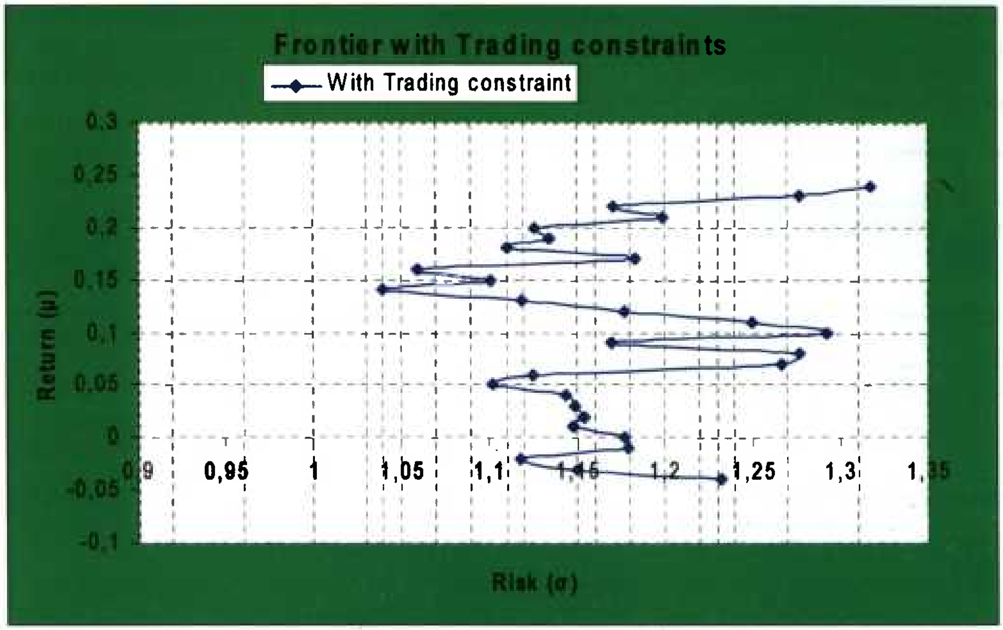


Figure 15 Efficient Frontier with trading constraint

However, as expected, the frontier is not as smooth as in the simpler cases. The question is to know whether we succeeded in computing the actual frontier or whether the SA algorithm erred in this complex case. The simplex method cannot be used anymore to compute the optimal solutions, because of the mixed integer constraints.

6.3.7 The complete case

Investigating each class of constraint separately was important in order to understand the behavior of the algorithm, but our final aim was to develop an approach that could handle more realistic situations where all the constraints are simultaneously imposed.

In this case the table for the input parameters takes the form:

Input Parameters		
Parameters	Units	Inputs
Return constraint	Fraction	-5%-26%
Budget constraint	Integer	100%
Floor constraint	Fraction	Yes (the lower bound of x_i is 0)
Ceiling constraint	Fraction	No
Trading constraint	Fraction	Yes($\underline{B}_i = \underline{S}_i = 0.10$ $(i = 1,\ldots,n)$
Number of assets in universe	Integer	20
Cardinality constraint	Integer	Yes (Maximum number of assets=7)

Table 11 Input parameters for the case of all constraints

The mean–risk frontier for the complete case of constraints is plotted in Fig. 16.

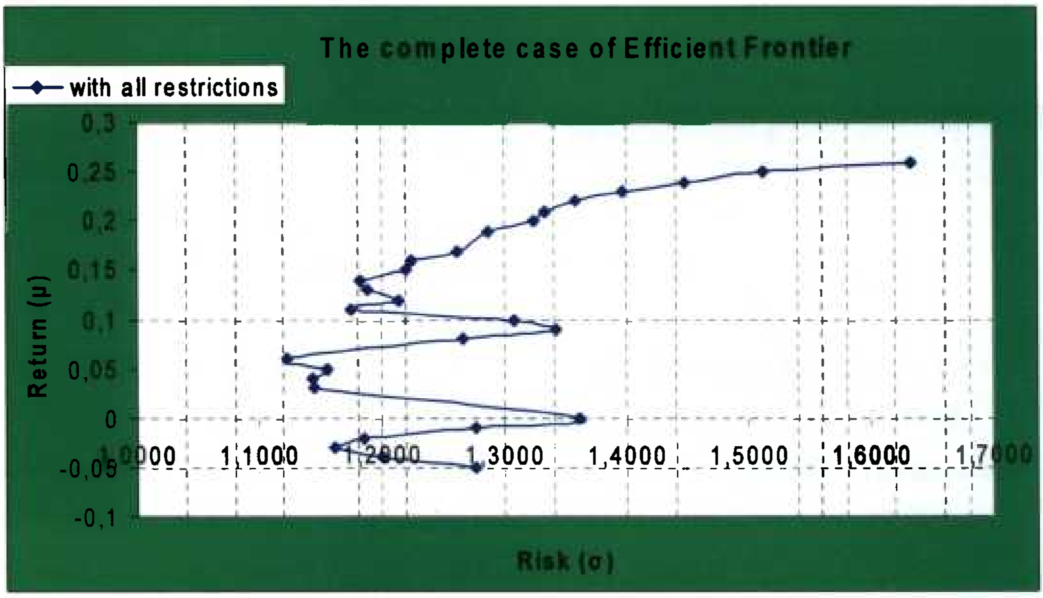


Figure 16 Efficient Frontier with all constraints

Figure 17 sums up all the previous results. It illustrates the effect of each class of constraints on the problem and allows some comparison of the mean–variance frontiers computed in each case.

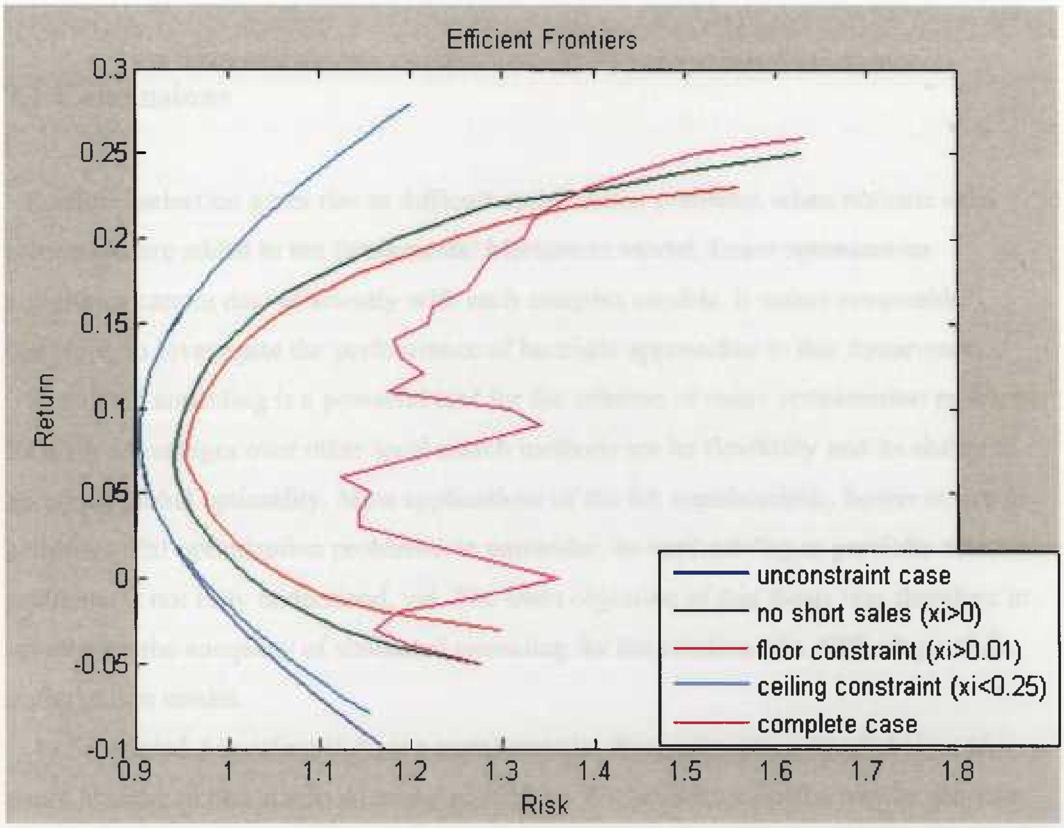


Figure 17 The diagram for all cases

The efficient frontier without any restriction includes the biggest feasible set of portfolios. The impact of short sales restriction and of the floor restriction is that the Efficient Frontier goes inner and we take a smaller space of feasible sets. The impact of ceiling constraint is to lose some extreme combinations in the lower part of Efficient Frontier and for this reason this constraint is not very important. In the complete case, the trading constraint creates sharp discontinuities.

Chapter 7

CONCLUSION AND FUTURE DEVELOPMENTS

7.1 Conclusions

Portfolio selection gives rise to difficult optimization problems when realistic side-constraints are added to the fundamental Markowitz model. Exact optimization algorithms cannot deal efficiently with such complex models. It seems reasonable, therefore, to investigate the performance of heuristic approaches in this framework.

Simulated annealing is a powerful tool for the solution of many optimization problems. Its main advantages over other local search methods are its flexibility and its ability to approach global optimality. Most applications of the SA metaheuristic, however, are to combinatorial optimization problems. In particular, its applicability to portfolio selection problems is not fully understood, yet. The main objective of this thesis was therefore to investigate the adequacy of simulated annealing for the solution of a difficult portfolio optimization model.

As Simulated Annealing (SA) is a metaheuristic, there are quite a lot of choices to make in order to turn it into an actual algorithm. We have developed a way to generate neighbors of a current solution. We have also proposed specific approaches to deal with each specific class of constraint, either by explicitly restricting the portfolios to remain in the feasible region or by penalizing infeasible portfolios.

Let us now try to draw some conclusions from this thesis. On the positive side, we can say that the research was successful, in the sense that the resulting algorithm allowed us to approximate the mean–variance frontier for medium-size problems within acceptable computing times. The algorithm is able to handle more classes of constraints than most other approaches found in the literature. Although there is a clear trade-off between the quality of the solutions and the time required to compute them, the algorithm can be said to be quite versatile since it does not rely on any restrictive properties of the model. For instance, the algorithm does not assume any underlying factor model for the generation of the covariance matrix. Also, the objective function could conceivably be replaced by any other measure of risk (semi-variance or functions of higher moments) without requiring any modification of the algorithm.

On the negative side, it must be noticed that the tailoring work required to account for different classes of constraints and to fine-tune the parameters of the algorithm was rather delicate. The trading constraints, in particular, are especially difficult to handle because of the discontinuities they introduce in the space of feasible portfolios. Introducing additional classes of constraints or new features in the model (e.g. transaction costs) would certainly prove quite difficult again.

Some of the insights gained from the research were:

- Both floor and ceiling constraints have a substantial negative impact on portfolio performance and should be examined critically relative to their associated administration
- The optimal portfolio with cardinality constraints often contains a large number of stocks with very low weightings.
- The number of assets in a portfolio increases the time of computation and the complexity of the optimization problem as a result of the complex constraints
- The implementation of cardinality constraints is essential for finding the best performing
  portfolio. The ability of the heuristic method to deal with cardinality constraints is one of its most powerful features.

7.2 Future work

Further work is suggested in the following areas:

Style, class or sector constraints can be added to the model. These constraints limit the proportion of the portfolio that can be invested in shares which fall into a style definition (e.g. value/growth, cyclical/defensive, small cap, liquid, rand-hedge etc.) or a market sector.

Different cardinality-constrained efficient frontiers will be generated for different values of *K*. Clearly, as *K* decreases (relative to the total number of stocks in the universe, *N*) the portfolio's potential performance (albeit with higher risk) increases and the frontier will move further away (upwards) from the cardinality-unconstrained efficient frontier. The magnitude of the sensitivity of this movement to different values of the ratio (*K/N*) is worth investigating.

The input forecasts for return and risk are point forecasts, making the model deterministic. A stochastic approach could be taken by attaching distributions to the input forecasts, resulting in an objective function which is also a distribution. While it is usually the mean which will be optimized, its variance can also be monitored.

"Before you begin a thing, remind yourself that difficulties and delays quite impossible to foresee are ahead. If you could see them clearly, naturally you could do a great deal to get rid of them but you can't. You can only see one thing clearly and that is your goal. Form a mental vision of that and cling to it through thick and thin."

Kathleen Norris

BIBLIOGRAPHY

[1] Bernd S., Martin D., Introduction to Modern Portfolio Optimization, USA 2005

[2] Busetti F., Metaheuristic Approaches to Realistic Portfolio Optimization, Master Thesis University of South Africa, 2000

[3] Crama, M. Schyns, European Journal of Operational Research 150 (2003) 546-571

[4] Farrell J., Portfolio Management, Theory and Applications 1997

[5] Kordas G., Lectures for the master lesson Portfolio Management 2006-2007

[6] Maringer D., Portfolio Management with Heuristic Optimization Netherlands, 2005

Appendix A

The source code

```
program portfolio

      parameter (n = 20, nmax = 20, ns=70000)
double precision mu(n), sigma(n,n), f, muport, wopt(n), fopt, t, w(n), wlow(n),
wup(n), w0(n), mintr(n), sumw
      logical tradec

      tradec = .false.

      t = 0.10 // Initial temperature parameter

      muport = 0.10 // Definition of required return

      OPEN (8, FILE = 'data2.txt')  // Open the source file for reading the
values of
                                                                expected returns
and the values of variance-
                                                                covariance
matrix

      read (8,*) mu(1), mu(2), mu(3), mu(4), mu(5)
      read (8,*) mu(6), mu(7), mu(8), mu(9), mu(10)
      read (8,*) mu(11), mu(12), mu(13), mu(14), mu(15)
      read (8,*) mu(16), mu(17), mu(18), mu(19), mu(20)
      read (8,*) sigma(1,1), sigma(1,2), sigma(1,3), sigma(1,4)
      read (8,*) sigma(1,5), sigma(1,6), sigma(1,7), sigma(1,8)
      read (8,*) sigma(1,9), sigma(1,10), sigma(1,11), sigma(1,12)
      read (8,*) sigma(1,13), sigma(1,14), sigma(1,15), sigma(1,16)
      read (8,*) sigma(1,17), sigma(1,18), sigma(1,19), sigma(1,20)
      read (8,*) sigma(2,1), sigma(2,2), sigma(2,3), sigma(2,4)
      read (8,*) sigma(2,5), sigma(2,6), sigma(2,7), sigma(2,8)
      read (8,*) sigma(2,9), sigma(2,10), sigma(2,11), sigma(2,12)
      read (8,*) sigma(2,13), sigma(2,14), sigma(2,15), sigma(2,16)
      read (8,*) sigma(2,17), sigma(2,18), sigma(2,19), sigma(2,20)
      read (8,*) sigma(3,1), sigma(3,2), sigma(3,3), sigma(3,4)
      read (8,*) sigma(3,5), sigma(3,6), sigma(3,7), sigma(3,8)
      read (8,*) sigma(3,9), sigma(3,10), sigma(3,11), sigma(3,12)
      read (8,*) sigma(3,13), sigma(3,14), sigma(3,15), sigma(3,16)
      read (8,*) sigma(3,17), sigma(3,18), sigma(3,19), sigma(3,20)
      read (8,*) sigma(4,1), sigma(4,2), sigma(4,3), sigma(4,4)
      read (8,*) sigma(4,5), sigma(4,6), sigma(4,7), sigma(4,8)
      read (8,*) sigma(4,9), sigma(4,10), sigma(4,11), sigma(4,12)
      read (8,*) sigma(4,13), sigma(4,14), sigma(4,15), sigma(4,16)
      read (8,*) sigma(4,17), sigma(4,18), sigma(4,19), sigma(4,20)
      read (8,*) sigma(5,1), sigma(5,2), sigma(5,3), sigma(5,4)
      read (8,*) sigma(5,5), sigma(5,6), sigma(5,7), sigma(5,8)
      read (8,*) sigma(5,9), sigma(5,10), sigma(5,11), sigma(5,12)
      read (8,*) sigma(5,13), sigma(5,14), sigma(5,15), sigma(5,16)
      read (8,*) sigma(5,17), sigma(5,18), sigma(5,19), sigma(5,20)
      read (8,*) sigma(6,1), sigma(6,2), sigma(6,3), sigma(6,4)
      read (8,*) sigma(6,5), sigma(6,6), sigma(6,7), sigma(6,8)
      read (8,*) sigma(6,9), sigma(6,10), sigma(6,11), sigma(6,12)
      read (8,*) sigma(6,13), sigma(6,14), sigma(6,15), sigma(6,16)
      read (8,*) sigma(6,17), sigma(6,18), sigma(6,19), sigma(6,20)
      read (8,*) sigma(7,1), sigma(7,2), sigma(7,3), sigma(7,4)
      read (8,*) sigma(7,5), sigma(7,6), sigma(7,7), sigma(7,8)
      read (8,*) sigma(7,9), sigma(7,10), sigma(7,11), sigma(7,12)
      read (8,*) sigma(7,13), sigma(7,14), sigma(7,15), sigma(7,16)
```

```
read (8,*) sigma(7,17), sigma(7,18), sigma(7,19), sigma(7,20)
read (8,*) sigma(8,1), sigma(8,2), sigma(8,3), sigma(8,4)
read (8,*) sigma(8,5), sigma(8,6), sigma(8,7), sigma(8,8)
read (8,*) sigma(8,9), sigma(8,10), sigma(8,11), sigma(8,12)
read (8,*) sigma(8,13), sigma(8,14), sigma(8,15), sigma(8,16)
read (8,*) sigma(8,17), sigma(8,18), sigma(8,19), sigma(8,20)
read (8,*) sigma(9,1), sigma(9,2), sigma(9,3), sigma(9,4)
read (8,*) sigma(9,5), sigma(9,6), sigma(9,7), sigma(9,8)
read (8,*) sigma(9,9), sigma(9,10), sigma(9,11), sigma(9,12)
read (8,*) sigma(9,13), sigma(9,14), sigma(9,15), sigma(9,16)
read (8,*) sigma(9,17), sigma(9,18), sigma(9,19), sigma(9,20)
read (8,*) sigma(10,1), sigma(10,2), sigma(10,3), sigma(10,4)
read (8,*) sigma(10,5), sigma(10,6), sigma(10,7), sigma(10,8)
read (8,*) sigma(10,9), sigma(10,10), sigma(10,11), sigma(10,12)
read (8,*) sigma(10,13), sigma(10,14), sigma(10,15), sigma(10,16)
read (8,*) sigma(10,17), sigma(10,18), sigma(10,19), sigma(10,20)
read (8,*) sigma(11,1), sigma(11,2), sigma(11,3), sigma(11,4)
read (8,*) sigma(11,5), sigma(11,6), sigma(11,7), sigma(11,8)
read (8,*) sigma(11,9), sigma(11,10), sigma(11,11), sigma(11,12)
read (8,*) sigma(11,13), sigma(11,14), sigma(11,15), sigma(11,16)
read (8,*) sigma(11,17), sigma(11,18), sigma(11,19), sigma(11,20)
read (8,*) sigma(12,1), sigma(12,2), sigma(12,3), sigma(12,4)
read (8,*) sigma(12,5), sigma(12,6), sigma(12,7), sigma(12,8)
read (8,*) sigma(12,9), sigma(12,10), sigma(12,11), sigma(12,12)
read (8,*) sigma(12,13), sigma(12,14), sigma(12,15), sigma(12,16)
read (8,*) sigma(12,17), sigma(12,18), sigma(12,19), sigma(12,20)
read (8,*) sigma(13,1), sigma(13,2), sigma(13,3), sigma(13,4)
read (8,*) sigma(13,5), sigma(13,6), sigma(13,7), sigma(13,8)
read (8,*) sigma(13,9), sigma(13,10), sigma(13,11), sigma(13,12)
read (8,*) sigma(13,13), sigma(13,14), sigma(13,15), sigma(13,16)
read (8,*) sigma(13,17), sigma(13,18), sigma(13,19), sigma(13,20)
read (8,*) sigma(14,1), sigma(14,2), sigma(14,3), sigma(14,4)
read (8,*) sigma(14,5), sigma(14,6), sigma(14,7), sigma(14,8)
read (8,*) sigma(14,9), sigma(14,10), sigma(14,11), sigma(14,12)
read (8,*) sigma(14,13), sigma(14,14), sigma(14,15), sigma(14,16)
read (8,*) sigma(14,17), sigma(14,18), sigma(14,19), sigma(14,20)
read (8,*) sigma(15,1), sigma(15,2), sigma(15,3), sigma(15,4)
read (8,*) sigma(15,5), sigma(15,6), sigma(15,7), sigma(15,8)
read (8,*) sigma(15,9), sigma(15,10), sigma(15,11), sigma(15,12)
read (8,*) sigma(15,13), sigma(15,14), sigma(15,15), sigma(15,16)
read (8,*) sigma(15,17), sigma(15,18), sigma(15,19), sigma(15,20)
read (8,*) sigma(16,1), sigma(16,2), sigma(16,3), sigma(16,4)
read (8,*) sigma(16,5), sigma(16,6), sigma(16,7), sigma(16,8)
read (8,*) sigma(16,9), sigma(16,10), sigma(16,11), sigma(16,12)
read (8,*) sigma(16,13), sigma(16,14), sigma(16,15), sigma(16,16)
read (8,*) sigma(16,17), sigma(16,18), sigma(16,19), sigma(16,20)
read (8,*) sigma(17,1), sigma(17,2), sigma(17,3), sigma(17,4)
read (8,*) sigma(17,5), sigma(17,6), sigma(17,7), sigma(17,8)
read (8,*) sigma(17,9), sigma(17,10), sigma(17,11), sigma(17,12)
read (8,*) sigma(17,13), sigma(17,14), sigma(17,15), sigma(17,16)
read (8,*) sigma(17,17), sigma(17,18), sigma(17,19), sigma(17,20)
read (8,*) sigma(18,1), sigma(18,2), sigma(18,3), sigma(18,4)
read (8,*) sigma(18,5), sigma(18,6), sigma(18,7), sigma(18,8)
read (8,*) sigma(18,9), sigma(18,10), sigma(18,11), sigma(18,12)
read (8,*) sigma(18,13), sigma(18,14), sigma(18,15), sigma(18,16)
read (8,*) sigma(18,17), sigma(18,18), sigma(18,19), sigma(18,20)
read (8,*) sigma(19,1), sigma(19,2), sigma(19,3), sigma(19,4)
read (8,*) sigma(19,5), sigma(19,6), sigma(19,7), sigma(19,8)
read (8,*) sigma(19,9), sigma(19,10), sigma(19,11), sigma(19,12)
read (8,*) sigma(19,13), sigma(19,14), sigma(19,15), sigma(19,16)
read (8,*) sigma(19,17), sigma(19,18), sigma(19,19), sigma(19,20)
read (8,*) sigma(20,1), sigma(20,2), sigma(20,3), sigma(20,4)
read (8,*) sigma(20,5), sigma(20,6), sigma(20,7), sigma(20,8)
```

```
      read (8,*) sigma(20,9), sigma(20,10), sigma(20,11), sigma(20,12)
      read (8,*) sigma(20,13), sigma(20,14), sigma(20,15), sigma(20,16)
      read (8,*) sigma(20,17), sigma(20,18), sigma(20,19), sigma(20,20)


      open (2, file=' output.txt')     // Open the destination file for writing
the values of
                                                              the portfolio
weights and the standard deviation of
                                                              the portfolio.


      call init(n,w,mu,muport)
      // The weights of the initial portfolio

      w0(1) = 0.05,       w0(2) = 0.05, w0(3) = 0.05, w0(4) = 0.05, w0(5) = 0.05
      w0(6) = 0.05, w0(7) = 0.05,      w0(8) = 0.05, w0(9) = 0.05, w0(10) = 0.05
      w0(11) = 0.05, w0(12) = 0.05, w0(13) = 0.05, w0(14) = 0.05, w0(15) = 0.05
      w0(16) = 0.05, w0(17) = 0.05, w0(18) = 0.05, w0(19) = 0.05,w0(20) = 0.05


      mintr(1) = 0.05, mintr(2) = 0.05, mintr(3) = 0.05, mintr(4) = 0.05
      mintr(5) = 0.05, mintr(6) = 0.05, mintr(7) = 0.05, mintr(8) = 0.05
      mintr(9) = 0.05, mintr(10) = 0.05, mintr(11) = 0.05, mintr(12) = 0.05
      mintr(13) = 0.05, mintr(14) = 0.05, mintr(15) = 0.05, mintr(16) = 0.05
      mintr(17) = 0.05, mintr(18) = 0.05, mintr(19) = 0.05, mintr(20) = 0.05



              // The lower value for one stock weight

      wlow(1) = -5.0, wlow(2) = -5.0, wlow(3) = -5.0, wlow(4) = -5.0
      wlow(5) = -5.0, wlow(6) = -5.0, wlow(7) = -5.0, wlow(8) = -5.0
      wlow(9) = -5.0, wlow(10) = -5.0, wlow(11) = -5.0, wlow(12) = -5.0
      wlow(13) = -5.0, wlow(14) = -5.0, wlow(15) = -5.0, wlow(16) = -5.0
      wlow(17) = -5.0, wlow(18) = -5.0, wlow(19) = -5.0, wlow(20) = -5.0


            // The upper value for one stock weight

      wup(1) = 0.25, wup(2) = 0.25, wup(3) = 0.25, wup(4) = 0.25, wup(5) = 0.25
      wup(6) = 0.25, wup(7) = 0.25, wup(8) = 0.25, wup(9) = 0.25, wup(10) =
0.25
      wup(11) = 0.25, wup(12) = 0.25, wup(13) = 0.25, wup(14) = 0.25,
           wup(15) = 0.25, wup(16) = 0.25, wup(17) = 0.25, wup(18) = 0.25
      wup(19) = 0.25, wup(20) = 0.25

      f = 0.0  // Initialization of the variable f

      call sa(n,mu,sigma,w,f,wopt,fopt,wlow,wup,nmax,w0,mintr,tradec,t,
      1     ns)

      muport = 0.0  // Initialization of the variables muport and sumw
      sumw = 0.0
      do i=1, n
            muport = muport + wopt(i) * mu(i)
         sumw = sumw + wopt(i)
      enddo
```

```
        // Write at the destination file and printing to the screen
      write(*,*) , 'Portfolio Weights'
      do i = 1, n
      write(*,900), wopt(i)
      write(2,900), wopt(i)
      enddo

      write(*,*) , 'Objective Function'
      write(*, *) fopt
      write(*,*) , 'Portfolio Return'
      write(*, 900) muport
      write(2,*) , 'Objective Function'
      write(2, *) fopt
      write(2,*) , 'Portfolio Return'
      write(2, 900) muport
900   format (F10.6)

      stop
      end

      subroutine init(n,w,mu,muport) // this function find an initial weight.

      integer n, k
      double precision mu(n), w(n), muport

      k = 0
      do i = 1, n
            if(mu(i) .lt. muport) k = k + 1
            w(i) = 0.0
      enddo
            if(k .gt. n-1) k = n - 2
      if(k .eq. 0)    k = 1

      w(k) = (muport-mu(k+1))/(mu(k)-mu(k+1))
      w(k+1) = 1.0 - w(k)

      return
      end

subroutine fcn(n,sigma,w,f,wup,wlow,nmax,w0,mintr,tradec)  // Define the
function to be

optimized.

integer n, nmax, count
double precision sigma(n,n), w(n), f, wup(n), wlow(n), pen, ceps, w0(n),
mintr(n), trade(n)
logical tradec

      f = 0.D0
     do 100, i = 1, n
            do j = 1, n
                  f = f + (w(i) * w(j) * sigma(i,j))
            enddo
100   continue

C    Set Penalties
      pen = 0.0

C    Floor and Ceilling Penalties
      do i = 1, n
          if(w(i) .lt. wlow(i)) pen = pen + 1.0D+16 * (w(i)-wlow(i))**2
```

```
          if(w(i) .gt. wup(i))  pen = pen + 1.0D+16 * (w(i)-wup(i))**2
      enddo

C     Number of Assets Penalty
      ceps = 1.0D-5
      count = 0
      do i = 1, n
          if(abs(w(i)) .gt. ceps) count = count + 1
      enddo
      if(count .gt. nmax) pen = pen + 1.0D+16 * (nmax-count)**2

C  Trading Penalties. (Allow very small deviations due to lack of numerical
accuracy)
      if(tradec) then
          do i = 1, n
              trade(i) = abs(w(i) - w0(i))
              if((trade(i) .le. mintr(i)) .and. (trade(i) .ge. 1.0D-4))
then
                  if(trade(i) .le. (mintr(i)/2)) then
                      pen = pen + 1.0D+16 * trade(i)**2
                  else
                      pen = pen + 1.0D+16 * (mintr(i)-trade(i))**2
                  end if
              end if
          enddo
      end if

      f = sqrt(f+pen)
     return
     end

C    Pick which 3 weights to change.
      subroutine pick(n,d,d1,d2,d3)

      integer n, d, d1, d2, d3

      if(d .le. (n-2)) then
          d1 = d
          d2 = d + 1
          d3 = d + 2
      else
          if(d .eq. (n-1)) then
              d1 = d
              d2 = d + 1
              d3 = 1
          else
              d1 = d
              d2 = 1
              d3 = 2
              d = 0
          end if
      end if
          return
      end


C    Simmulated Annealing
     subroutine sa(n,mu,sigma,w,f,wopt,fopt,wlow,wup,nmax,w0,mintr,
      1    tradec,t,ns)

      integer n
      integer d, d1, d2, d3, nacc, nnew, nrej, ns
      double precision mu(n), sigma(n,n), f, w(n), wp(n), wlow(n),
```

```
     1   wup(n), step, fp, lb1, lb2, lb3, ub1, ub2, ub3, lb, ub,
    2   w0(n), mintr(n), wopt(n), fopt, t, p, pp
     double precision exprep,num2,num3, fstar(4)
     logical tradec, quit
     real ranmar

C     Initialize the random number generator.
      call rmarin(21421,21480)

C     Set initial values
      nacc = 0
      do i = 1, 4
            fstar(i) = -1.0D+20
      enddo

C     Evaluate the objective at the initial point.
      call fcn(n,sigma,w,f,wup,wlow,nmax,w0,mintr,tradec)
      fopt = f

C     Begin iterations.
100   nnew = 0, nrej = 0, d = 0

      do 200, iter = 1, ns

            d = d + 1
            do i = 1, n
                  wp(i) = w(i)
            enddo

C           Pick 3 weights to change.
            call pick(n,d,d1,d2,d3)

C           Compute lower and upper bounds for the step.
            num2 = (mu(d1)-mu(d3))/(mu(d2)-mu(d3))
            num3 = (mu(d2)-mu(d1))/(mu(d2)-mu(d3))

            lb1 = w(d1) - wup(d1)
            ub1 = w(d1) - wlow(d1)
            if(num2 .gt. 0) then
                  lb2 = (wlow(d2)-w(d2))/num2
                  ub2 = (wup(d2)-w(d2))/num2
            else
                  lb2 = (wup(d2)-w(d2))/num2
                  ub2 = (wlow(d2)-w(d2))/num2
            endif
            if(num3 .gt. 0) then
                  lb3 = (wlow(d3)-w(d3))/num3
                  ub3 = (wup(d3)-w(d3))/num3
            else
                  lb3 = (wup(d3) - w(d3))/num3
                  ub3 = (wlow(d3)-w(d3))/num3
            endif

            lb = max(lb1,lb2,lb3)
            ub = min(ub1,ub2,ub3)

C              Take a step in these 3 weights.
               step = (ub-lb) * ranmar() + lb
            wp(d1) = w(d1) - step
            wp(d2) = w(d2) + step * num2
            wp(d3) = w(d3) + step * num3
```

```
C         Evaluate the obective at the new point.
             call fcn(n,sigma,wp,fp,wup,wlow,nmax,w0,mintr,tradec)

C         Keep the new point if it decreases the objective.
             if(fp .lt. f) then
                   do i = 1, n
                         w(i) = wp(i)
                   enddo
                   f = fp
                   nacc = nacc + 1
C          If smaller than any other point record a new minimum.
                   if(fp .lt. fopt) then
                         do i = 1, n
                               wopt(i) = wp(i)
                         enddo
                         fopt = fp
                         nnew = nnew + 1
                   endif
C If the point does not improve the objective use the Metropolis algorithm to
accept or reject it.
             else
                   p = exprep(-(fp-f)/t)
                   pp = ranmar()
                   if(pp .lt. p) then
                         do i=1, n
                               w(i) = wp(i)
                         enddo
                         f = fp
                         nacc = nacc + 1
                   else
                         nrej = nrej + 1
                   endif
             endif
200    continue

C      Check termination criteria.
       fstar(1) = f
       if((fstar(1) - fopt) .le. 0.000001) quit = .true.
       do i = 1, 4
             if(abs(f-fstar(i)) .gt. 0.000001) quit = .false.
       enddo
C      Terminate SA if appropriate
       if(quit) then
       do i = 1, n
             w(i) = wopt(i)
       enddo
       f = fopt
       return
       endif

C      Lower the temperature and prepare for another loop.
       t= 0.85 * t
       do i = 4, 2, -1
             fstar(i) = fstar(i-1)
       enddo
       do i = 1, n
             w(i) = wopt(i)
       enddo
       f = fopt
       goto 100
       end

       function  exprep(rdum)
```

```
      !DEC$ ATTRIBUTES DLLEXPORT :: exprep
c  this function replaces exp to avoid under- and overflows and is
c  designed for ibm 370 type machines. it may be necessary to modify
c  it for other machines. note that the maximum and minimum values of
c  exprep are such that they has no effect on the algorithm.

      double precision  rdum, exprep

      if (rdum .gt. 174.) then
         exprep = 3.69d+75
      else if (rdum .lt. -180.) then
         exprep = 0.0
      else
         exprep = exp(rdum)
      end if
      return
      end

      subroutine rmarin(ij,kl)
      !DEC$ ATTRIBUTES DLLEXPORT :: rmarin
c  this subroutine and the next function generate random numbers. see
c  the comments for sa for more information. the only changes from the
c  orginal code is that (1) the test to make sure that rmarin runs first
c  was taken out since sa assures that this is done (this test didn't
c  compile under ibm's vs fortran) and (2) typing ivec as integer was
c  taken out since ivec isn't used. with these exceptions, all following
c  lines are original.

c this is the initialization routine for the random number generator ranmar()
c note: the seed variables can have values between:    0 <= ij <= 31328
c                                                       0 <= kl <= 30081
      real u(97), c, cd, cm
      integer i97, j97
      common /raset1/ u, c, cd, cm, i97, j97
      i = mod(ij/177, 177) + 2
      j = mod(ij    , 177) + 2
      k = mod(kl/169, 178) + 1
      l = mod(kl,     169)
      do 2 ii = 1, 97
         s = 0.0
         t = 0.5
         do 3 jj = 1, 24
            m = mod(mod(i*j, 179)*k, 179)
            i = j
            j = k
            k = m
            l = mod(53*l+1, 169)
            if (mod(l*m, 64) .ge. 32) then
               s = s + t
            endif
            t = 0.5 * t
3        continue
         u(ii) = s
2     continue
      c = 362436.0 / 16777216.0
      cd = 7654321.0 / 16777216.0
      cm = 16777213.0 /16777216.0
      i97 = 97
      j97 = 33
      return
      end

      function ranmar()
```

```
      real u(97), c, cd, cm
      integer i97, j97
      common /raset1/ u, c, cd, cm, i97, j97
         uni = u(i97) - u(j97)
         if( uni .lt. 0.0 ) uni = uni + 1.0
         u(i97) = uni
         i97 = i97 - 1
         if(i97 .eq. 0) i97 = 97
         j97 = j97 - 1
         if(j97 .eq. 0) j97 = 97
         c = c - cd
         if( c .lt. 0.0 ) c = c + cm
         uni = uni - c
         if( uni .lt. 0.0 ) uni = uni + 1.0
         ranmar = uni
      return
      end
```

Appendix B

I will give for the 20 stocks that I used in my master thesis the expected returns, the standard deviations, the variance-covariance matrix and the inverse of the variance-covariance matrix for these twenty assets.

Furthermore, I will find the analytical solution for the unconstraint case of portfolio optimization for the twenty assets. We suppose that these twenty assets follows a multivariate Normal Distribution.

$$\begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ . \\ . \\ . \\ R_{20} \end{bmatrix} \sim N_{20}\left(\begin{bmatrix} -0.0632 \\ -0.0336 \\ -0.0154 \\ . \\ . \\ . \\ 0.2677 \end{bmatrix}, \Sigma_{20x20} \right).$$

In the next table, we find the expected return and standard deviation of these twenty assets:

Asset Data	Expected Return	Standard Deviation
EMPORIKI	-6.316%	149.34%
AGROTIKI	-3.362%	183.05%
INTRACOM	-1.545%	260.20%
OPAP	-0.217%	162.39%
MOTOROIL	-0.018%	173.24%
ASPIS	0.517%	216.97%
COSMOTE	7.048%	162.11%
FOLLI	7.192%	185.58%
ETHNIKI	8.411%	191.82%
EUROBANK	8.423%	165.29%
ALPHA	8.950%	173.15%
EGNATIA	10.736%	257.60%
DEI	11.000%	181.51%
COCACOLA	12.819%	183.62%
MINOAN	14.746%	229.20%
PIREOS	15.213%	174.08%
INTRALOT	18.005%	227.24%
FORTHNET	21.712%	264.40%
KIPROU	25.179%	221.19%
VIVARTIA	26.774%	208.94%

As a first step, we must calculate the inverse matrix of Σ (20x20), for this calculation we use the MATLAB program for taking this matrix (see table in Appendix B):

After we compute the qualities:

$A = \mathbf{1}' \times \mathbf{\Sigma}^{-1} \times \mathbf{1} = 1.2268$

$B = \mathbf{1}' \times \mathbf{\Sigma}^{-1} \times \boldsymbol{\mu} = 0.1035$

$C = \boldsymbol{\mu}' \times \mathbf{\Sigma}^{-1} \times \boldsymbol{\mu} = 0.0697$

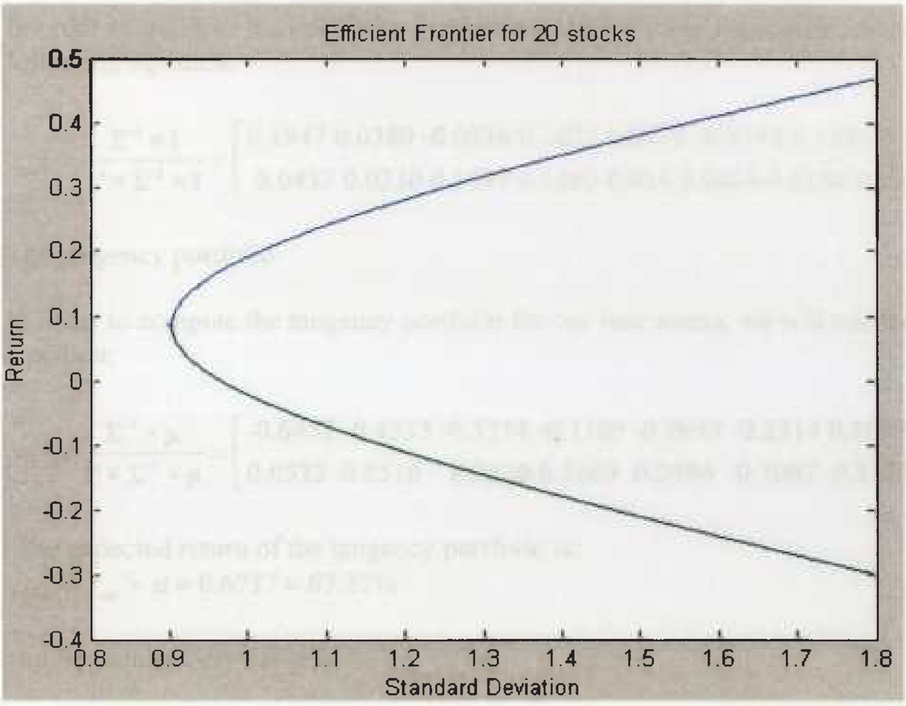
$D = A \cdot C - B^2 = 0.0748$

Therefore the equation of the Efficient Frontier in Mean-Standard Deviation space is

$$\mu_P = \frac{B}{A} \pm \frac{1}{A}\sqrt{D(A \cdot \sigma_P{}^2 - 1)}$$

$$= \frac{0.1035}{1.2268} \pm \frac{1}{1.2268} \cdot \sqrt{0.0748 \cdot (1.2268 \cdot \sigma_P{}^2 - 1)}$$

$$= 0.0844 \pm \frac{1}{1.2268} \cdot \sqrt{0.0918 \cdot \sigma_P{}^2 - 0.0748}$$



Efficient Frontier for 20 risky assets

The minimum standard-deviation portfolio is attained at

$$\mu_P = \frac{B}{A} = \frac{0.1035}{1.2268} = 0.0844 = 8.44\%$$

which yields a minimum standard deviation of

$$\sigma_P = \sqrt{\frac{C - \frac{B^2}{A}}{D}} = \sqrt{\frac{0.0697 - \frac{0.1035^2}{1.2268}}{0.0748}} = 0.90283$$

Finally, the asymptotes of the hyperbola are

$$\mu_P = \frac{B}{A} \pm \sigma_P \cdot \sqrt{\frac{D}{A}} = \frac{0.1035}{1.2268} \pm \sigma_P \cdot \sqrt{\frac{0.0748}{1.2268}} = 0.0844 \pm \sigma_P \cdot 0.2469$$

At this point, we will calculate the above two interesting portfolios, *the minimum-variance portfolio* and the *tangency portfolio*.

The minimum-variance portfolio

In order to compute the minimum-variance portfolio for our four assets, we will use the following equation:

$$x_{\min} = \frac{\mathbf{\Sigma}^{-1} \times \mathbf{1}}{\mathbf{1}' \times \mathbf{\Sigma}^{-1} \times \mathbf{1}} = \begin{bmatrix} 0.1947 & 0.0380 & -0.0936 & 0.1622 & 0.0475 & -0.0342 & 0.1285 & 0.1021 & -0.1406 & 0.0582 \\ 0.0437 & 0.0230 & 0.1097 & 0.1102 & 0.034 & 0.0464 & 0.0194 & 0.0139 & 0.0096 & 0.1267 \end{bmatrix}^T$$

The tangency portfolio

In order to compute the tangency portfolio for our four assets, we will use the following equation:

$$x_{\tan} = \frac{\mathbf{\Sigma}^{-1} \times \boldsymbol{\mu}}{\mathbf{1}' \times \mathbf{\Sigma}^{-1} \times \boldsymbol{\mu}} = \begin{bmatrix} -0.6452 & -0.4333 & -0.3214 & -0.1189 & -0.4957 & -0.2514 & 0.1009 & 0.1786 & -0.3482 & -0.0892 \\ 0.0532 & 0.0516 & 0.3520 & 0.3669 & 0.2496 & 0.7007 & 0.3357 & 0.2474 & 0.5007 & 0.5660 \end{bmatrix}^T$$

The expected return of the tangency portfolio is:
$\mu_{\tan} = x_{\tan}' \times \mu = 0.6737 = 67.37\%$

and its standard deviation is:

$$\sigma_{\tan} = \sqrt{x_{\tan}' \times \Sigma \times x_{\tan}} = 2.5513$$

Now, we use the derived utility function $V(\mu,\sigma^2)=\mu-\theta\cdot\sigma^2$, and we make calculations in order to find the optimal combinations for different values of risk aversion parameter.

We use the following mathematical equations:

$$\sigma_*^2(\theta)=\frac{1}{A}+\frac{D}{4\cdot A\cdot\theta^2},\quad \mu_*(\theta)=\frac{B}{A}+\frac{D}{2\cdot A\cdot\theta}$$

For θ =1:

$$\sigma_*^2(1)=\frac{1}{A}+\frac{D}{4\cdot A\cdot\theta^2}=\frac{1}{1.2268}+\frac{0.0748}{4\cdot 1.2268\cdot 1^2}=0.815+0.015=0.83$$

$$\mu_*(1)=\frac{B}{A}+\frac{D}{2\cdot A\cdot\theta}=\frac{0.1035}{1.2268}+\frac{0.0748}{2\cdot 1.2268\cdot 1}=0.0844+0.03=0.1144$$

The optimal mean-standard deviation combination is $(\mu,\sigma)=(0.1144,\ 0.911)$

As next step, we try to compute the portfolio that yields this optimal combination for this investor.

The weights of this portfolio are:

$$x^*(0.1144)=\lambda_1\cdot\mathbf{\Sigma}^{-1}\times\mathbf{I}+\lambda_2\cdot\mathbf{\Sigma}^{-1}\times\boldsymbol{\mu}$$

$$\lambda_1=\frac{C-B\cdot\mu_P}{D}=\frac{0.0697-0.1035\cdot 0.1144}{0.0748}=0.7735$$

$$\lambda_2=\frac{A\cdot\mu_P-B}{D}=\frac{1.2268\cdot 0.1144-0.1035}{0.0748}=0.4926$$

Thus

$$x^*(0.1144)=\begin{bmatrix}0.1518\ 0.0140\ \text{-}0.1052\ 0.1479\ 0.0198\ \text{-}0.0452\ 0.1271\ 0.1060\ \text{-}0.1512\ 0.0507\\ 0.0442\ 0.0244\ 0.1221\ 0.1233\ 0.0454\ 0.0797\ 0.0356\ 0.0258\ 0.0347\ 0.1491\end{bmatrix}^T$$

For θ =2

$$\sigma_*^2(2)=\frac{1}{A}+\frac{D}{4\cdot A\cdot\theta^2}=\frac{1}{1.2268}+\frac{0.0748}{4\cdot 1.2268\cdot 2^2}=0.815+0.0038=0.8188$$

$$\mu_*(2)=\frac{B}{A}+\frac{D}{2\cdot A\cdot\theta}=\frac{0.1035}{1.2268}+\frac{0.0748}{2\cdot 1.2268\cdot 2}=0.0844+0.015=0.0994$$

The optimal mean-standard deviation combination is $(\mu,\sigma)=(0.0994,\ 0.905)$

As next step, we try to compute the portfolio that yields this optimal combination for this investor.

The weights of this portfolio are:

$$x^*(0.0994) = \lambda_1 \cdot \mathbf{\Sigma}^{-1} \times \mathbf{I} + \lambda_2 \cdot \mathbf{\Sigma}^{-1} \times \boldsymbol{\mu}$$

$$\lambda_1 = \frac{C - B \cdot \mu_P}{D} = \frac{0.0697 - 0.1035 \cdot 0.0994}{0.0748} = 0.7943$$

$$\lambda_2 = \frac{A \cdot \mu_P - B}{D} = \frac{1.2268 \cdot 0.0994 - 0.1035}{0.0748} = 0.2466$$

$$x^*(0.0994) = \begin{bmatrix} 0.1732\ 0.0260\ \text{-}0.0994\ 0.1551\ 0.0336\ \text{-}0.0397\ 0.1278\ 0.1041\ \text{-}0.1459\ 0.0545 \\ 0.0440\ 0.0237\ \ 0.1159\ 0.1168\ 0.0399\ \ 0.0631\ 0.0275\ \ 0.0198\ 0.0222\ \ 0.1379 \end{bmatrix}^T$$

For θ =4

$$\sigma_*^2(4) = \frac{1}{A} + \frac{D}{4 \cdot A \cdot \theta^2} = \frac{1}{1.2268} + \frac{0.0748}{4 \cdot 1.2268 \cdot 4^2} = 0.815 + 0.00095 = 0.81595$$

$$\mu_*(4) = \frac{B}{A} + \frac{D}{2 \cdot A \cdot \theta} = \frac{0.1035}{1.2268} + \frac{0.0748}{2 \cdot 1.2268 \cdot 4} = 0.0844 + 0.0076 = 0.092$$

The optimal mean-standard deviation combination is $(\mu, \sigma) = (0.092,\ 0.9033)$

As next step, we try to compute the portfolio that yields this optimal combination for this investor.

The weights of this portfolio are:

$$x^*(0.092) = \lambda_1 \cdot \mathbf{\Sigma}^{-1} \times \mathbf{I} + \lambda_2 \cdot \mathbf{\Sigma}^{-1} \times \boldsymbol{\mu}$$

$$\lambda_1 = \frac{C - B \cdot \mu_P}{D} = \frac{0.0697 - 0.1035 \cdot 0.092}{0.0748} = 0.8045$$

$$\lambda_2 = \frac{A \cdot \mu_P - B}{D} = \frac{1.2268 \cdot 0.092 - 0.1035}{0.0748} = 0.1252$$

$$x^*(0.092) = \begin{bmatrix} 0.1838\ 0.0319\ \text{-}0.0965\ 0.1586\ 0.0404\ \text{-}0.0370\ 0.1282\ 0.1031\ \text{-}0.1433\ 0.0563 \\ 0.0439\ 0.0233\ \ 0.1129\ \ 0.1136\ 0.0372\ 0.0549\ 0.0235\ 0.0169\ \ 0.0160\ \ 0.1323 \end{bmatrix}^T$$

The Variance-Covariance matrix:

ETHNIKI	FOLLI	COSMOTE	ASPIS	MOTOROIL	OPAP	INTRACOM	AGROTIKI	EMPORIKI	VCV Matrix
1.13141	0.598354	0.490228	1.06131	0.784329	0.50892	1.25726	1.0643	2.24962	EMPORIKI
1.69618	0.717928	0.604965	1.76652	1.21724	1.01983	2.3623	3.32763	1.0643	AGROTIKI
1.88273	1.33779	1.01563	2.55642	1.9691	1.39226	6.87804	2.3623	1.25726	INTRACOM
1.3935	0.648085	0.805106	1.3731	0.854813	2.6729	1.39226	1.01983	0.50892	OPAP
1.3099	0.985929	0.730244	1.7842	2.97506	0.854813	1.9691	1.21724	0.784329	MOTOROIL
1.91526	1.25951	0.773767	4.59089	1.7842	1.3731	2.55642	1.76652	1.06131	ASPIS
1.05586	0.610411	2.62227	0.773767	0.730244	0.805106	1.01563	0.604965	0.490228	COSMOTE
0.40513	3.40992	0.610411	1.25951	0.985929	0.648085	1.33779	0.717928	0.598354	FOLLI
3.69631	0.40513	1.05586	1.91526	1.3099	1.3935	1.88273	1.69618	1.13141	ETHNIKI
1.75354	0.696134	0.72963	1.36977	0.950604	0.930204	1.5982	1.1891	0.859663	EUROBANK
1.89004	0.821541	0.786498	1.57234	1.18484	1.03703	2.01679	1.21461	0.840777	ALPHA
1.08899	0.793726	0.457257	1.67007	0.893263	0.608922	1.40883	1.05595	0.92772	EGNATIA
1.12184	0.304513	0.721882	0.943951	0.930581	0.671902	1.57639	0.984222	0.69657	DEI
1.20455	0.397557	0.975037	0.641352	0.845396	0.51063	0.763949	0.847077	0.497487	COCACOLA
1.46336	0.54923	0.761089	1.84122	1.50042	0.579636	2.69568	1.36717	0.938013	MINOAN
2.01854	0.757284	0.608568	1.83045	1.36143	0.957101	1.75235	1.44906	1.12822	PIREOS
1.74434	0.852342	0.79351	1.94583	1.52304	1.00349	2.54398	1.51865	0.939663	INTRALOT
1.67804	0.79764	0.806259	1.59812	1.55038	0.734183	2.18029	1.33228	0.85078	FORTHNET
2.09007	1.0068	0.717333	2.2441	1.40466	1.09672	2.38056	1.62734	0.948791	KIPROU
1.10108	0.430986	0.315637	0.633038	0.542421	0.461209	1.12441	0.798689	0.336405	VIVARTIA

VIVARTIA	KIPROU	FORTHNET	INTRALOT	PIREOS	MINOAN	COCACOLA	DEI	EGNATIA	ALPHA	EUROBANK
0.336405	0.948791	0.85078	0.939663	1.12822	0.938013	0.497487	0.69657	0.92772	0.840777	0.859663
0.798689	1.62734	1.33228	1.51865	1.44906	1.36717	0.847077	0.984222	1.05595	1.21461	1.1891
1.12441	2.38056	2.18029	2.54398	1.75235	2.69568	0.763949	1.57639	1.40883	2.01679	1.5982
0.461209	1.09672	0.734183	1.00349	0.957101	0.579636	0.51063	0.671902	0.608922	1.03703	0.930204
0.542421	1.40466	1.55038	1.52304	1.36143	1.50042	0.845396	0.930581	0.893263	1.18484	0.950604
0.633038	2.2441	1.59812	1.94583	1.83045	1.84122	0.641352	0.943951	1.67007	1.57234	1.36977
0.315637	0.717333	0.806259	0.79351	0.608568	0.761089	0.975037	0.721882	0.457257	0.786498	0.72963
0.430986	1.0068	0.79764	0.852342	0.757284	0.54923	0.397557	0.304513	0.793726	0.821541	0.696134
1.10108	2.09007	1.67804	1.74434	2.01854	1.46336	1.20455	1.12184	1.08899	1.89004	1.75354
0.656685	1.52943	1.3436	1.14576	1.50081	1.41876	0.718706	0.921955	1.03818	1.46176	2.73395
0.660651	1.48146	1.47436	1.42578	1.54394	1.40444	0.768996	1.04657	0.915576	3.02291	1.46176
1.00721	1.34504	1.32232	1.09591	1.01825	1.11523	0.624079	0.597125	6.73692	0.915576	1.03818
0.358187	0.918738	1.08293	0.962833	0.900893	0.828883	0.595984	3.33018	0.597125	1.04657	0.921955
0.317102	0.693435	0.809498	0.685051	0.592202	0.900439	3.40083	0.595984	0.624079	0.768996	0.718706
0.900896	1.68261	1.76224	1.51189	1.2254	5.2769	0.900439	0.828883	1.11523	1.40444	1.41876
0.67913	2.03465	1.23026	1.35982	3.05209	1.2254	0.592202	0.900893	1.01825	1.54394	1.50081
0.819177	1.824414	1.95252	5.21012	1.35982	1.51189	0.685051	0.962833	1.09591	1.42578	1.14576
0.514302	1.70805	7.00959	1.95252	1.23026	1.76224	0.809498	1.08293	1.32232	1.47436	1.3436
0.801752	4.90889	1.70805	1.824414	2.03465	1.68261	0.693435	0.918738	1.34504	1.48146	1.52943
4.40533	0.801752	0.514302	0.819177	0.67913	0.900896	0.317102	0.358187	1.00721	0.660651	0.656685

The inverse Variance-Covariance Matrix:

Inverse VCV matrix	EMPORIKI	AGROTIKI	INTRACOM	OPAP	MOTOROIL	ASPIS	COSMOTE	FOLLI	ETHNIKI
EMPORIKI	0.6203	-0.0751	-0.0129	0.0266	-0.0020	-0.0036	-0.0163	-0.0435	-0.0592
AGROTIKI	-0.0751	0.5131	-0.0812	-0.0379	-0.0114	-0.0460	0.0299	-0.0025	-0.0642
INTRACOM	-0.0129	-0.0812	0.2665	-0.0280	-0.0313	-0.0237	-0.0101	-0.0222	0.0361
OPAP	0.0266	-0.0379	-0.0280	0.5336	-0.0214	-0.0658	-0.0750	-0.0276	-0.1061
MOTOROIL	-0.0020	-0.0114	-0.0313	-0.0214	0.5668	-0.0784	-0.0168	-0.0604	0.0094
ASPIS	-0.0036	-0.0460	-0.0237	-0.0658	-0.0784	0.4076	0.0082	-0.0509	-0.0440
COSMOTE	-0.0163	0.0299	-0.0101	-0.0750	-0.0168	0.0082	0.4957	-0.0480	-0.0630
FOLLI	-0.0435	-0.0025	-0.0222	-0.0276	-0.0604	-0.0509	-0.0480	0.3768	0.1155
ETHNIKI	-0.0592	-0.0642	0.0361	-0.1061	0.0094	-0.0440	-0.0630	0.1155	0.6877
EUROBANK	-0.0239	-0.0168	0.0013	-0.0399	0.0366	0.0112	-0.0190	-0.0379	-0.1127
ALPHA	0.0072	0.0212	-0.0518	-0.0346	-0.0122	-0.0129	-0.0074	-0.0482	-0.1408
EGNATIA	-0.0348	-0.0042	0.0015	0.0036	0.0036	-0.0384	0.0039	-0.0082	0.0118
DEI	-0.0347	-0.0246	-0.0385	-0.0085	-0.0411	0.0074	-0.0434	0.0314	-0.0087
COCACOLA	-0.0031	-0.0439	0.0206	0.0123	-0.0550	0.0270	-0.0898	-0.0126	-0.0869
MINOAN	-0.0252	-0.0032	-0.0597	0.0451	-0.0506	-0.0387	-0.0147	0.0336	0.0100
PIREOS	-0.1109	-0.0451	0.0104	0.0135	-0.1014	-0.0523	0.0277	-0.0111	-0.1594
INTRALOT	-0.0154	-0.0179	-0.0395	-0.0048	-0.0402	-0.0285	-0.0096	-0.0040	-0.0375
FORTHNET	0.0009	-0.0062	-0.0097	0.0162	-0.0379	0.0035	-0.0062	-0.0059	-0.0259
KIPROU	0.0221	-0.0173	-0.0273	-0.0036	0.0031	-0.0477	0.0066	-0.0282	-0.0659
VIVARTIA	0.0222	-0.0202	-0.0153	0.0013	-0.0011	0.0241	0.0052	-0.0207	-0.0688

VIVARTIA	KIPROU	FORTHNET	INTRALOT	PIREOS	MINOAN	COCACOLA	DEI	EGNATIA	ALPHA	EUROBANK
0.0222	0.0221	0.0009	-0.0154	-0.1109	-0.0252	-0.0031	-0.0347	-0.0348	0.0072	-0.0239
-0.0202	-0.0173	-0.0062	-0.0179	-0.0451	-0.0032	-0.0439	-0.0246	-0.0042	0.0212	-0.0168
-0.0153	-0.0273	-0.0097	-0.0395	0.0104	-0.0597	0.0206	-0.0385	0.0015	-0.0518	0.0013
0.0013	-0.0036	0.0162	-0.0048	0.0135	0.0451	0.0123	-0.0085	0.0036	-0.0346	-0.0399
-0.0011	0.0031	-0.0379	-0.0402	-0.1014	-0.0506	-0.0550	-0.0411	0.0036	-0.0122	0.0366
0.0241	-0.0477	0.0035	-0.0285	-0.0523	-0.0387	0.0270	0.0074	-0.0384	-0.0129	0.0112
0.0052	0.0066	-0.0062	-0.0096	0.0277	-0.0147	-0.0898	-0.0434	0.0039	-0.0074	-0.0190
-0.0207	-0.0282	-0.0059	-0.0040	-0.0111	0.0336	-0.0126	0.0314	-0.0082	-0.0482	-0.0379
-0.0688	-0.0659	-0.0259	-0.0375	-0.1594	0.0100	-0.0869	-0.0087	0.0118	-0.1408	-0.1127
-0.0027	-0.0265	-0.0218	0.0086	-0.1104	-0.0624	-0.0086	-0.0377	-0.0221	-0.0963	0.6526
0.0043	0.0169	-0.0196	-0.0167	-0.0848	-0.0262	-0.0104	-0.0432	-0.0001	0.6091	-0.0963
-0.0267	-0.0113	-0.0124	-0.0015	0.0025	-0.0011	-0.0121	0.0000	0.1744	-0.0001	-0.0221
0.0046	0.0029	-0.0124	-0.0020	-0.0033	0.0148	-0.0080	0.3796	0.0000	-0.0432	-0.0377
0.0105	0.0054	-0.0001	0.0052	0.0385	-0.0235	0.3698	-0.0080	-0.0121	-0.0104	-0.0086
-0.0210	-0.0192	-0.0167	-0.0005	0.0211	0.2803	-0.0235	0.0148	-0.0011	-0.0262	-0.0624
0.0008	-0.1049	0.0174	0.0096	0.6989	0.0211	0.0385	-0.0033	0.0025	-0.0848	-0.1104
-0.0099	-0.0214	-0.0268	0.2766	0.0096	-0.0005	0.0052	-0.0020	-0.0015	-0.0167	0.0086
0.0102	-0.0133	0.1839	-0.0268	0.0174	-0.0167	-0.0001	-0.0124	-0.0124	-0.0196	-0.0218
0.0011	0.3405	-0.0133	-0.0214	-0.1049	-0.0192	0.0054	0.0029	-0.0113	0.0169	-0.0265
0.2576	0.0011	0.0102	-0.0099	0.0008	-0.0210	0.0105	0.0046	-0.0267	0.0043	-0.0027

I want to say some words in order to give a complete reference for portfolio optimization and the skewness and kurtosis are two important factors for our problem.

Consider a series $\{X_t\}_{t=1}^{T}$ with mean μ and standard deviation σ. Let $\mu_r = E[(x-\mu)^r]$ be the r-th central moment of X_t with $\mu_2 = \sigma^2$. The coefficient of skewness and kurtosis are defined as:

$$skewness \quad \tau = \frac{\mu_3}{\sigma^3} = \frac{E[(x-\mu)^3]}{E[(x-\mu)^2]^{3/2}}$$

$$kurtosis \quad k = \frac{\mu_4}{\sigma^4} = \frac{E[(x-\mu)^4]}{E[(x-\mu)^2]^2}$$

If X_t is symmetrically distributed, $\mu_3 = 0$ and thus τ will be zero. The Gaussian distribution has $\tau = 0$ and $k = 3$. When k>3, the distribution of X_t is said to have fat tails. Normality is often a maintained assumption in estimation and finite sample inference. A joint test of $\tau = 0$ and $k - 3 = 0$ is often used as a test of normality.

In the next table, I will give the values of skewness and kurtosis for our twenty assets.

Name of asset	skewness	kurtosis
EMPORIKI	0.348611	3.704856
AGROTIKI	-0.26157	3.046742
INTRACOM	0.03599	4.218056
OPAP	0.104495	2.292909
MOTOROIL	-0.1725	1.372067
ASPIS	0.296248	2.441136
COSMOTE	0.432716	0.663797
FOLLI	0.462191	1.869819
ETHNIKI	0.075708	3.1727
EUROBANK	-0.09349	1.79495
ALPHA	0.171901	1.840069
EGNATIA	1.547973	8.289168
DEI	0.497616	2.373892
COCACOLA	0.081976	0.380145
MINOAN	0.173097	1.995385
PIREOS	0.710706	5.638078
INTRALOT	-0.3915	5.20948
FORTHNET	1.678378	11.97687
KIPROU	-0.22867	2.607486
VIVARTIA	0.497616	2.373892

Δωρεά