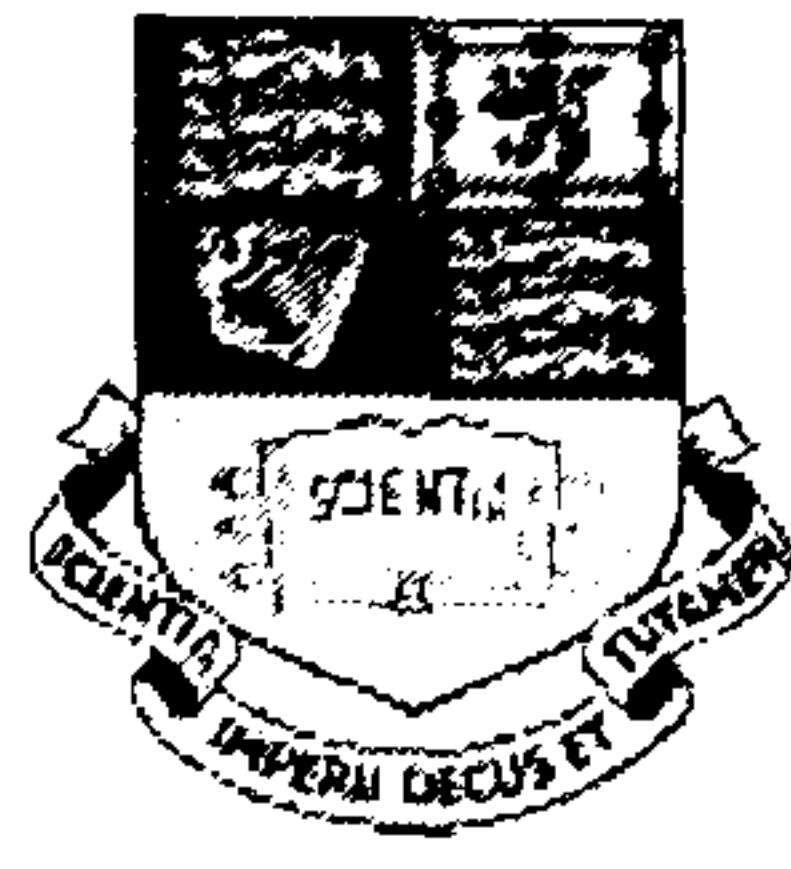




Centre for Quantitative Finance
Imperial College of Science, Technology and Medicine
University of London

Asset Allocation under Liquidity Constraints

Agostino Russo

A thesis submitted for the degree of Doctor of Philosophy of the
University of London

September, 2003

To my daughter

ABSTRACT

Liquidity has been thoroughly studied in market microstructure research and its effects are widely recognised by market practitioners, but little research has been devoted to liquidity in the area of asset allocation, where often only simplistic models are used. The purpose of the thesis is to address the impact of liquidity costs and to suggest an algorithm to perform portfolio optimisation when more realistic liquidity structures are used. We illustrate the implications of the uncertainty of future liquidity premia in a general equilibrium framework, obtaining a modified Capital Asset Pricing Model (CAPM). The equilibrium obtained addresses some of the inconsistencies revealed by empirical research regarding the classical Sharpe-Lintner model. We then introduce a more complex liquidity structure and solve the resulting asset allocation problem within a mean-variance framework. When fixed costs are included a complex combinatorial problem is obtained. We suggest a heuristic algorithm to find approximate solutions. In the case of variable unit liquidity costs the model involves a non-convex objective function subject to non-linear constraints. Two different approaches are proposed to solve the problem. In the first case, a sequence of convex subproblems is obtained using either a generalisation of the Sequential Convex Programming (SCP) algorithm or a modified version of the Space Mapping algorithm (SM). The advantage of SCP is that the subproblems obtained are convex and separable. This feature allows us to represent the subproblems as generalised network flows so that very efficient algorithms can be used. SM produces a sequence of quadratic subproblems that can be easily solved using available quadratic optimisers. In the second case, we analyse the performance of three stochastic search heuristics, namely Genetic Algorithms, Simulated Annealing, and Tree Annealing. Since the problem domain is represented by a multidimensional tetrahedron, we apply a new method to perform a uniform mapping from a hypercube to a tetrahedron. This allows us to express candidate solutions as points of a hypercube, avoiding the bias implied by other techniques. The routines presented are not dependent on the particular functional form of the objective function and more complex risk measures can also be used.

Table of Contents

List of Figures	6
List of Tables	9
1. Introduction.....	10
1.1. Definition of Liquidity	13
1.2. Market Microstructure Underpinnings	15
1.3. The Reaction Function.....	18
1.4. Review of Financial Literature	20
2. Liquidity and the CAPM.....	23
2.1. Utility Maximisation with Liquidity Constraints.....	23
2.2. Market Equilibrium.....	26
2.3. The Modified CAPM	29
2.4. Including Liquidity Constraints into the Black and Litterman Model.	33
3. Liquidity and Asset Allocation	38
3.1. Mean-Variance Model	38
3.2. Liquidity Cost Function	40
3.3. Numerical Examples	46
3.4. Fixed Transaction Costs.....	54
3.5. Portfolio Rebalancing Problem.....	57
4. Network Flows and Sequential Convex Programming.....	59
4.1. Sequential Convex Programming	59
4.2. Piecewise-Linear Approximation of the Subproblem.....	66
4.3. Network Flow Formulation of the Subproblem.....	70
4.4. Generalised Network Simplex Algorithm	77
4.5. Parallel Arcs.....	84
4.6. Computational Results.....	88
5. Space Mapping.....	92
5.1. Space Mapping Algorithm.....	92
5.2. Space Mapping Algorithm and Asset Allocation	97

5.3. Computational Results	100
6. Stochastic Search Heuristics for Portfolio Optimisation	103
6.1. Genetic Algorithms	104
6.2. Invisible Sections of the Efficient Frontier	110
6.3. Low Discrepancy Sequences and Marginal Probability Distributions 112	
6.4. Simulated Annealing.....	118
6.5. Tree Annealing.....	123
6.6. Genetic Algorithms and Boltzmann Distribution	125
6.7. Computational Results	126
7. Conclusions.....	130
7.1. Future Research	132
Bibliography	133

List of Figures

Figure 2.1: Relation between expected return and systematic risk with and without the liquidity effect.....	31
Figure 2.2: Relation between the expected gross return and the expected future relative spread.	33
Figure 2.3: Implied market equilibrium excess return.....	36
Figure 3.1: Liquidity cost function, representing the actual price paid as a function of the transaction size.	41
Figure 3.2: Spread as a function of the transaction size.	41
Figure 3.3: Transaction costs as a function of the transaction size.....	42
Figure 3.4: Two-asset portfolio weights.	49
Figure 3.5: Variance.....	49
Figure 3.6: Expected rate of return.	49
Figure 3.7: Feasible combinations of two assets (gross).	49
Figure 3.8: Feasible combinations of two assets (net).....	49
Figure 3.9: Variance.....	50
Figure 3.10: Feasible combinations of two assets (net).....	50
Figure 3.11: Variance (gross).	51
Figure 3.12: Variance (net).....	51
Figure 3.13: Expected rate of return (gross).	51
Figure 3.14: Expected rate of return (net).....	51
Figure 3.15: Feasible combinations of three assets (gross).	51
Figure 3.16: Feasible combinations of three assets (net).....	51
Figure 3.17: Isovariance and isoreturn lines for three-asset portfolios (gross).	52
Figure 3.18: Isovariance and isoreturn lines for three-asset portfolios (net). ..	52
Figure 3.19: Feasible combinations of four assets (gross).....	53

Figure 3.20: Feasible combinations of four assets with liquidity coefficients of 3% and 5%.	53
Figure 3.21: Feasible combinations of four assets with liquidity coefficients of 10% and 25%.	53
Figure 3.22: Approximation of the fixed cost function.	56
Figure 4.1: Method of the moving asymptotes.	62
Figure 4.2: Globally Convergent Method of Moving Asymptotes.....	64
Figure 4.3: Moving asymptotes updating.	65
Figure 4.4: Convex approximation evolution with parameter δ	66
Figure 4.5: Network representation of the generalised minimum cost flow problem when the non-linear functions are approximated by PL functions.....	72
Figure 4.6: Network representation of the budget constraint.	73
Figure 4.7: Network representation of the objective function.	73
Figure 4.8: Network representation of the constraint on the expected portfolio return.	74
Figure 4.9: Subgraph representing portion of the portfolio rebalancing problem.	76
Figure 4.10: Network representation of the portfolio rebalancing problem. ...	77
Figure 4.11: Non degenerate augmented forest.	86
Figure 4.12: Pivot operation.	87
Figure 4.13: 10 Assets (LC 0.1 and 0.25).	90
Figure 4.14: 30 Assets (LC 0.1 and 0.25).	91
Figure 5.1: Space Mapping Algorithm.	95
Figure 5.2: Modified Space Mapping Algorithm.	99
Figure 5.3: 10 Assets (LC 0.1 and 0.25).	101
Figure 5.4: 30 Assets (LC 0.1 and 0.25).	102
Figure 6.1: Invisible section of the efficient frontier.	111
Figure 6.2: Uniform deviates.	113
Figure 6.3: Sobol sequence.	113

Figure 6.4: Mapping coordinates on the tetrahedron. 115

Figure 6.5: Non-invertible mapping. 117

Figure 6.6: Uniform mapping. 117

Figure 6.7: Objective function and associated density function $p(w,T)$ for
different values of T 120

Figure 6.8: Set of undominated improved solutions..... 129

Figure 6.9: Genetic Algorithm..... 129

Figure 6.10: Simulated Annealing. 129

Figure 6.11: Tree Annealing. 129

Figure 6.12: GA with Boltzmann tournament selection. 129

List of Tables

Table 3.1: Four-asset example. 47

Table 3.2: Two-asset portfolios. 48

Table 4.1: Computational results for MMA and MMA PLA. 89

Table 5.1: Computational results for SM.....101

Table 6.1: Computational results for search heuristics.....127

1. Introduction

The importance of liquidity in asset allocation decisions is widely recognised by market practitioners. There is also a growing interest in the topic in market microstructure research where many papers have been published. In finance, many authors have provided empirical evidence for the relevance of the liquidity premium. Among others, Amihud and Mendelson (1989) found a concave relationship between excess returns and the relative spread. Brennan and Subrahmanyam (1996) and Brennan et al. (1998) found a significant relationship between returns and trading volumes for both the NYSE and the NASDAQ. Hamon and Jaquillat (1997) presented evidence of a negative link between risk-adjusted expected returns and free float on the one hand and of a positive link between risk-adjusted expected returns and relative spread on the other. Despite the empirical evidence and the general interest in the topic, there has been a lack of financial research in the area of asset allocation under liquidity constraints. Most of the papers that have been published, regarding either static or dynamic portfolio management, include at most linear transaction costs and ignore the impact of a more complex liquidity structure and of the uncertainty deriving from the stochastic nature of liquidity.

The principal aim of this thesis is to investigate the impact of liquidity on asset allocation and to present a portfolio management model that is able to accommodate different types of liquidity structures, and, in particular, the case in which the bid-ask spread depends on the transaction size.

There are two major difficulties that can explain the lack of research activity in the topic. First, liquidity is difficult to define, multifaceted, and hard to estimate. Most simply it is the ability to perform a transaction without cost. In practice, liquidity is usually defined according to different dimensions: the

divergence between buying and selling prices, often referred to as the bid-ask spread, is a measure of the *tightness* of the market for low volume transactions. Larger orders almost always face somewhat worse execution. This additional premium is a function of the *depth* of the market, defined as the volume able to be traded at a particular price level. In the model proposed, the liquidity cost is represented by the relative bid-ask spread (following Amihud and Mendelson (1986), and Jacoby et al. (2001)). However, the way liquidity is introduced into the optimisation framework is general enough to allow for different representations of the liquidity cost.

A second shortcoming is that the introduction of liquidity into an optimisation model might imply non-linearities in both the objective function and the constraints, making the problem much harder to handle in practice. We suggest two different approaches to solve the problem. In the first case, the objective function is approximated by a sequence of convex functions, in the second case, stochastic search heuristics are used.

We first analyse the impact of a random liquidity spread (represented by the bid-ask spread) within a general equilibrium framework. A modified Capital Asset Pricing Model (CAPM)¹ is obtained where the systematic risk depends on the after spread returns. The model implies a non-linear relation when gross returns are used and it can therefore explain some of the empirical results that cast doubt on the validity of the Sharpe-Lintner model. The new CAPM can easily be integrated within models that are based on the security market line, like the Black and Litterman model.

We then consider a more complex liquidity structure, incorporating the effect of the market depth, so that the spread depends on the transaction size. The

problem is analysed within a mean-variance framework, obtaining a modification of the Markowitz model. Whilst solving the problem, the stability and interpretability of the solution is also considered. Once liquidity depends on the transaction size it is reasonable to expect that extreme allocations will be avoided by the optimisers, whenever this would involve a large trade. In this interpretation, the introduction of liquidity might be thought of as a generalisation of the additional ad-hoc constraints used by market participants when solving a mean-variance model.

The thesis is divided as follows: In Chapter 2, a simple general equilibrium model is presented that considers the future spread as a random variable within a utility maximisation framework. A modified CAPM is obtained. In particular, it turns out that once liquidity is included into the CAPM, the modified beta is non-linear in the traditional beta and, furthermore, it reflects the uncertainties deriving from both the asset returns and the future liquidity premium. The results obtained from the modified CAPM are used to extend the Black and Litterman model so that equilibrium expected excess returns, net of the liquidity costs, can be estimated directly.

In Chapter 3, some of the insights from the utility maximisation approach are used to formulate an equivalent mean-variance model that can accommodate more complex liquidity structures. A heuristic algorithm is proposed to solve the problem when fixed costs are included. When the bid-ask spread is dependent on the transaction size the model implies the minimisation of a non-convex objective function subject to non-linear constraints.

In Chapter 4, an algorithm to solve the problem above is presented. A sequence of convex separable subproblems is first obtained using the Globally

¹ The CAPM was introduced by Sharpe (1964), Lintner (1965), and Mossin (1996). It is an equilibrium model where

Convergent Method of Moving Asymptotes. For each subproblem the objective function (which is convex and separable) and the constraints are transformed into piecewise-linear functions. The problem is then formulated as a minimum cost generalised network flow problem with parallel arcs that can be solved using an algorithm based on the primal simplex method, which exploits the particular structure of the network formulation.

In Chapter 5, an alternative approach is proposed that uses an adaptation of the Space Mapping (SM) algorithm developed by Bandler (1998) in order to generate a sequence of quadratic problems. The algorithm uses a quasi-Newtonian iterative approach to force convergence between the approximate model and the non-convex model.

In Chapter 6, we outline three routines based on stochastic search heuristics, namely Genetic Algorithms, Simulated Annealing, and Tree Annealing. A variant of a Genetic Algorithm with parent selection based on the Boltzmann distribution is also analysed. In the models above each solution is represented by a point in a hypercube and a new technique is used to perform a mapping into a tetrahedron while maintaining a uniform coverage. Differently from the previous methods, the routines proposed in this chapter are global optimisers. The concluding remarks are in Chapter 7.

1.1. Definition of Liquidity

Market liquidity is an elusive concept, reflecting its multi-faceted nature. A definition that is generally agreed upon is the following: if a stock is liquid a high volume of transactions not motivated by information will not affect the

asset returns are a linear function of the measure of systematic risk.

stock price. According to Keynes liquidity is "the ease with which a large trade can be executed without affecting prices significantly."

There are generally four dimensions associated with liquidity: *Tightness* is how far transaction prices diverge from mid-market prices, and can be generally measured by the bid-ask spread. *Depth* denotes either the volume of trades possible without affecting prevailing market prices, or the amount of orders on the order-books of market-makers at a given time. *Resiliency* refers to the speed with which price fluctuations resulting from trades are dissipated. *Thickness* is defined by either the number of shares or the number of transactions per unit time.

In the model proposed we assume that investor transactions will not change the market prices (the investor is a price taker) but the liquidity cost, designated as the relative spread, is a function of the transaction size. The model therefore captures both the tightness and the depth of the market. The choice of such a function is motivated by the fact that the greater the volume, the further the transaction price must deviate from the quotes in order to reach enough waiting orders. A steeper reaction function, meaning a sparser backlog of limit orders, entails a greater price impact for a fixed size transaction.

The impact of liquidity costs on a portfolio might be quite large. According to Barra (1998), for U.S. institutional investors, the average spread for an S&P 500 stock is approximately 7¢ per share, plus the processing costs of about 5¢ per share. Considering a typical institutional transaction of a stock in the S&P 500, the median quote depth is typically 3,500 shares, and the median share price is \$40. For a typical institutional order of 10,000 shares, Barra estimates an additional market impact to execute the trade of 13¢, since it exceeds the typical quote depth of 3,500. The average total transaction cost for such a trade is therefore 25¢ per share. Assuming a portfolio turnover of 100% per

year, each stock will be bought and sold each year. The total yearly transaction costs for each stock will therefore be 50¢. If the total rate of return is 12% and the price of the average stock is \$40, the yearly transaction cost of 50¢ per share represents roughly 10% of the total stock rate of return.

Moreover it must be added that the uncertainty of future liquidity conditions also impacts the risk profile of the portfolio, particularly considering that liquidity shocks are usually highly correlated among different asset classes. Disregarding the impact of liquidity on a portfolio might therefore imply an asset allocation that is sub-optimal. In a paper from Bangia et al. (1998), the authors show that ignoring the liquidity effect can produce underestimates of market risk in emerging markets by as much as 25-30%.

As Nicholas Dunbar (Risk, October 1998) observed: "Portfolios are usually marked to market at the middle of the bid-ask spread and many hedge funds used models that incorporate this assumption. In late August, there was only one realistic value for the portfolio: the bid price. Amid such massive sell-offs, only the first seller obtains a reasonable price for its security; the rest lose a fortune by having to pay a liquidity premium if they want a sale. ...Models should be revised to include the bid-ask behaviour".

1.2. Market Microstructure Underpinnings

Microstructure theory proposes two explanations for illiquidity in the markets: inventory control and asymmetric information. These represent the sources of risk facing liquidity suppliers.

Let us consider a generic equity market to which investors send orders for execution. For executions to occur, offsetting buy and sell orders must be matched together. Some matching occurs automatically from buy and sell

orders arriving in the market simultaneously. This natural match rate is surprisingly low, however. Barra estimates, from a sample of orders made available by the New York Stock Exchange, that incoming market orders cross with one another about 20% of the time. Given the low natural match rate, some market participants must modify their behaviour so as to accommodate mismatched order flow. An investor who carries on with his trading plans independent of the presence or absence of matching order flow is a liquidity demander. An example of such an investor would be one who submits market orders. By contrast, we term an investor who adjusts his trading to accommodate mismatched order flow a liquidity provider. Examples of liquidity providers include specialists and limit order submitters.

Several authors explained the motivation behind the different transaction needs of the market participants. The trader in a hurry is an uninformed agent who must realise a transaction before a given deadline, as shown by Grossman and Miller (1988), and Harris (1998). If the deadline is sufficiently far off and the spread is wide he will have a tendency to place a limit order. A hurried trader doesn't necessarily behave irrationally: his behaviour can be considered as the result of costly access to information. The degree of aversion towards illiquidity is also linked to investment horizon and to management style. For Amihud and Mendelson (1986) the differences in horizon are at the root of the phenomenon of liquidity clienteles with the investor with the shorter horizon overweighting his portfolio with liquid stocks and the investor with the longer horizon overweighting his portfolio with less liquid stocks. In equilibrium a non-linear concave relation is then predicted between stock return and liquidity. Handa and Schwartz (1996) make the distinction between patient investors, who are providers of liquidity and mostly place limit orders, and hurried investors, who are asking for liquidity and will place their orders at any price. The latter are the root of excessive short-term volatility (they give priority to the time at which a trade is carried out) and the strategies of the

former are akin to volatility capturing strategies (they give priority to the price).

The compensation the liquidity providers receive for rendering their service determines the extent of illiquidity in the market. When we examine the provision of liquidity, we see that there is significant competition among liquidity providers to render this service. However, even though barriers to entry are fairly low, there is some opportunity for segmentation. One form of segmentation is for a liquidity provider to concentrate on a few stocks. In fact, this is the route taken by the specialist. Another form of segmentation is to concentrate on meeting specific types of liquidity demand. This is the route taken by principal bid desks. This suggests that one should measure the quantity of liquidity provided in terms of the amount of risk assumed by the liquidity provider.

When the liquidity provider takes a position into inventory, he faces two basic uncertainties. First, he does not know the price at which he will be able to remove the position from his inventory. Second, he does not know how long the position will remain in his inventory. We refer to the length of time it takes the liquidity provider to get out of the position as the time-to-clear. A long time-to-clear is undesirable since it implies a lower rate of return, a greater exposure to the risk of adverse price movement, and a lower rate of capital utilisation.

As noted by several authors (e.g. Campbell, Lo and MacKinlay, 1997) an additional risk exists due to the possibility of taking the other side of a potentially information-based trade. Since informed traders agents tend to trade more heavily in order to profit from their fleeting information advantage, larger trades might imply a greater probability of incurring adverse selection costs. If informed traders do tend to transact greater quantities in order to

maximise profits from their information, then the specialist will widen quotes when larger volume orders enter the market.

The size of the premium charged by immediacy providers to cover these expected costs determines the liquidity cost. Moreover the risk increases with the trade size, because of a longer time-to-clear and higher probability of informed agents being present in the market. Easley and O'Hara (1987) and Hasbrouck (1988) find a positive correlation between trade size and price impact. McNish and Wood (1992) reveal an association between trade size and the bid-ask spread.

The variability in the extent of informational asymmetry and any implications of such on time-varying liquidity is less thoroughly documented. If informed and liquidity traders have different trading tendencies, then the distribution of market information may be somewhat signalled by the nature of trading activity at any given moment. And because liquidity is supplied according to the presumed presence of insiders in the market, we can expect to find the depth of the market to be time varying in a manner related to the market conditions. The thickness of trading, may also be a function of asymmetry of information.

1.3. The Reaction Function

The reaction function describes the relationship between trade sizes and transaction costs, or, differently interpreted, the price impact caused by large trades. The estimation of the reaction function is beyond the scope of this thesis, and we will generally assume that the functional form of the reaction function is known. For completeness some estimation techniques will be briefly illustrated.

A first method might categorise the trades by asset traded and trade size. Then the mean of past costs at a given trade size would provide a forecast of future costs. However, there are three major econometric difficulties:

Market impact is not directly observable; rather it must be estimated. To reduce estimation error, large statistical samples are required, which implies gathering data over extended time periods. Thus, only time-averaged behaviour can be measured with any precision. But the underlying phenomenon is a product of the highly dynamic market environment. Thus, long-term averages have little forecasting ability.

Trade data suffers from a censoring problem. Since investors avoid costly trades, we never see the most costly potential trades in the record. Unless we somehow compensate for this, estimates derived from trade data will systematically underestimate market impact costs.

There is an uneven level of information available for assets. We know more about heavily traded assets and less about thinly traded assets. Yet it is the thinly traded assets that tend to have the higher market impact costs, and it is the more costly-to-trade assets that are most important for the portfolio construction problem.

Elkins/McSherry calculates the market impact as the percentage difference between the execution price and a benchmark price, where the benchmark is the mean of the day's open, close, high and low prices.

As an alternative, multifactor models, based on microstructure theory and calibrated using a cross-section of the available market data, are also used. Barra, for instance, developed a model in which a forecast of the transaction costs depends on the traded volume, the elasticity (the response of order flow

to price signals), the volatility of the asset, the trading intensity, and the distribution of different transaction sizes. These models try to mimic the behaviour of liquidity suppliers.

Models also have been proposed that analyse the cross-sectional variation of the measured price impact, explaining this variation as a function of predetermined firm characteristics. For instance, Hasbrouck (1991) investigates the relation between firm size and a measure of price impact. Glosten and Harris (1988) investigate the cross-sectional relation using proxies for adverse-selection.

A different approach, suggested by Engle and Lange, (1997) is to try to estimate the market depth directly. It uses "price duration", the time it takes for a price to move a fixed prespecified amount. The price duration often corresponds to a price event. The number of events changes day by day, and, by selecting a threshold the number of events can be adjusted to the resolution desired. Price duration measures how fast excess demand must be absorbed by the market, and the expected value of a price duration is shown by Engle and Russell (1998) to be inversely proportional to expected volatility. The net volume divided by price duration is therefore a measure of market depth.

1.4. Review of Financial Literature

Broadly speaking, our approach falls within the Markowitz framework, where a trade-off between mean return and variance is present. The origin of the field has been attributed to Markowitz (1952) and Roy (1952). Implications for the valuation of assets arose with the capital asset pricing model (CAPM) of Sharpe (1964), Lintner (1965), and Mossin (1966).

The first treatments for problems with linear costs were modifications of the simplex method for quadratic programming, which can handle a quadratic objective but not quadratic constraints. The variance is included in the program objective, weighted by a parameter λ , and the solutions on the efficient frontier are found by varying the parameter. See, e.g., Perold (1984), where a method for efficiently ranging over such a parameterisation of the efficient frontier is proposed. These models consider a fixed spread so that the total transaction costs are linear in the transaction size, focusing on the reallocation effects of transaction costs and implying that the magnitude of portfolio reallocation decreases as transaction costs increase. A different approach was proposed by Konno and Yamazaki (1991). In their model the quadratic term of the objective function is replaced by a risk measure expressed as an absolute deviation, so that linear programming algorithms can be applied. Adcock and Mead (1994) combined a modulus function for transaction costs with the Mean Absolute Deviation (MAD)² model.

More recently, advances in non-linear convex optimisation, particularly in the area of efficient interior-point methods, have enabled the development of more complex models that can handle large number of variables and constraints. Provided that the constraints can be cast as second order cone constraints, efficient algorithms can be used. A convex cost function, where the spread is itself an increasing function of the transaction size, can therefore be solved by these models. See, e.g., Vanderbei (1994), Jones and Mitra (1997).

In order to capture the realism of portfolio planning a number of discrete restrictions have been considered by several researches. Buy-in thresholds (minimum levels below which assets are not purchased), cardinality

² In this formulation the variance of the portfolio return is replaced by the mean absolute deviation. Konno and Yamazaki (1991) showed that the MAD problem is equivalent to the Mean-Variance problem when asset returns are normally distributed.

constraints (number of assets in the portfolio), and roundlots (discrete numbers of assets taken as the basic unit of investment) are some examples of complication of the basic model that imply a quadratic mixed integer problem. The same models are often used to solve for piecewise linear transaction costs that include a fixed component (so that part of the cost function is concave). Kellerer, Masini and Speranza (2000) use a linear programming based heuristic to solve problems incorporating both roundlots and fixed costs. Chang et al. (2000) use heuristic algorithms to solve cardinality constrained problems. Most of these models use a linear measure of the risk as proposed by Konno and Yamazaki (1991).

Regarding the network formulation of the asset allocation problem, a proposal was first made by Crum, Klingman, and Tavis (1979). Successively, Mulvey and Vladimirou (1988), Dembo, Mulvey and Zenios (1989), presented more detailed network models using specialised algorithms. Mulvey and Vladimirou (1992), and Mulvey and Ziemba (1995) also expressed the problem in terms of a stochastic network flow representation for the multiple-scenario, multi-period portfolio optimisation.

Dynamic models investigating the optimal execution strategy for liquidating portfolios include Jarrow and Subramanian (1997), Bertsimas and Lo (1998), Almgren and Chriss (1999). Longstaff (1999) solves the intertemporal portfolio choice problem when the investor is limited to trading strategies of bounded variation, approximating the thin-trading illiquidity in actual markets where investors can not trade unlimited amounts of securities instantaneously. The resulting optimal trading strategy endogenously imposes borrowing and short-selling constraints on the investor. The implied discounts for illiquidity can be substantial.

2. Liquidity and the CAPM

In the model proposed in this chapter we consider the unit liquidity cost as a scalar. Differently from existing literature, we will distinguish between the liquidity costs at the beginning of the period, when the assets are purchased and the costs at the end of the period, when the assets are liquidated. At the beginning of the period the transaction cost is assumed to be deterministic. At the end of the period, the spread is unknown and it is modelled as a random variable. In section 2.1, the single investor utility maximisation problem is first solved. In section 2.2, the conditions for a market equilibrium are illustrated and a modified CAPM is obtained. In section 2.3, we analyse the effects of liquidity on the market equilibrium comparing the results to the classical Sharpe-Lintner model. In section 2.4, we show how the modified CAPM can be used as an estimator of the net expected excess return vector by extending the methodology originally presented by Black and Litterman (1992).

2.1. Utility Maximisation with Liquidity Constraints

In this section we will introduce the formulation of the problem in terms of utility maximisation. The first order conditions are equivalent to the ones obtained from a mean-variance optimisation when the net asset returns are joint-normally distributed or when the utility functions are quadratic.

The model considers a one period world where the liquidity cost is expressed by the relative spread, and where the future bid-ask spread, which does not depend on the asset weights, is uncertain. It is assumed that investors are risk-averse, utility maximisers of the end-period wealth and price-takers. It is also

assumed that assets are infinitely divisible. After-spread asset returns are assumed to follow a joint-normal distribution or, alternatively, utility functions are quadratic. There are no liquidity costs for the risk-free asset, and it is possible to borrow and lend unlimited amounts at the risk-free rate.

Each investor solves the following problem:

$$\underset{b, w_i}{\text{Max}} E[U_j(W_{1,j})] \quad [2.1]$$

s.t.

$$W_{0,j} = b + \sum_i w_i (1 + s_{0,i}) = 1 \quad [2.2]$$

$$W_{1,j} = b(1 + r_f) + \sum_i w_i (1 + r_i)(1 - s_{1,i}) = bR_f + \sum_i w_i R_i (1 - s_{1,i}) \quad [2.3]$$

$$s_i = \frac{\text{Ask}_i - \text{Mid}_i}{\text{Mid}_i} = \frac{\text{Mid}_i - \text{Bid}_i}{\text{Mid}_i} \quad [2.4]$$

Where:

U_j is the utility function of investor j , $U' > 0$, $U'' < 0$.

$W_{0,j}$ is the initial relative wealth of investor j .

$W_{1,j}$ is the terminal relative wealth of investor j , after all positions are liquidated.

b is the amount invested in the risk-free asset.

w_i is the amount invested in the risky asset i (net of transaction costs).

$s_{0,i}$ is the deterministic relative spread of asset i at the beginning of the period.

$s_{1,i}$ is the random relative spread of asset i at the end of the period.

r_f is the rate of return on the risk-free asset.

$R_f = 1 + r_f$ is the return on the risk free asset.

r_i is the the gross rate of return of asset i .

$R_i = 1 + r_i$ is the gross return of asset i .

The relative liquidity spread is expressed by the variable s . Note that when s is 1, the ask price is twice the mid price and the bid price is 0. The liquidity cost in this case is infinite and we should expect an infinite gross rate of return. We will show in the following sections that this is indeed the case. It is therefore natural to assume that s assumes values in the interval $[0,1)$. Negative asset weights can also be considered provided that the sign of s is reversed. The budget constraint in [2.2] implies that when the assets are first purchased a deterministic spread is paid. Further transaction costs are incurred when the positions are closed as shown in [2.3]. At the end period, indicated by the subscript 1, the transaction costs are unknown and s is a random variable. By including the constraints into the objective function the problem can be reformulated as follows.

$$\underset{w_i}{Max} E[U_j (R_f (1 - \sum_i w_i (1 + s_{0,i})) + \sum_i w_i R_i (1 - s_{1,i}))] \quad [2.5]$$

The first order condition with respect to w_i implies:

$$\frac{\partial E[U_j]}{\partial w_i} = E[U'_j (W_{1,j}) (R_i (1 - s_{1,i}) - R_f (1 + s_{0,i}))] = 0 \quad [2.6]$$

$$\begin{aligned} \frac{\partial E[U_j]}{\partial w_i} &= E[U'_j] E[R_i (1 - s_{1,i}) - R_f (1 + s_{0,i})] + \\ &+ COV(U'_j, R_i (1 - s_{1,i})) = 0 \end{aligned} \quad [2.7]$$

Given our assumptions for the net returns and/or utility function, [2.7] can also be expressed as¹:

¹ The proof is straightforward when the utility function is quadratic, since all the terms of 3rd order or greater in a Taylor expansion around the mean are 0. When the net returns follow a Gaussian distribution see Hanoch and Levy (1969).

$$E[U'_j]E[R_i(1-s_{1,i})-R_f(1+s_{0,i})] + E[U''_j]COV(W_{1,j}, R_i(1-s_{1,i})) = 0 \quad [2.8]$$

Or alternatively as:

$$E[R_i(1-s_{1,i})] - R_f(1+s_{0,i}) = \lambda_j COV\left(\sum_k w_k R_k(1-s_{1,k}), R_i(1-s_{1,i})\right) \quad [2.9]$$

where

$$\lambda_j = -\frac{E[U''_j]}{E[U'_j]} \quad [2.10]$$

The parameter λ can be interpreted as the Arrow-Pratt absolute risk aversion index. Also the first order condition [2.9] is equivalent to the one that would be obtained from a mean-variance optimisation under similar assumptions, provided that the after-spread returns are used. It is easy to prove that the right hand side of the equation is indeed equivalent to the first order derivative of the variance of the after-spread portfolio return. Similarly, the terms on the left hand side of the equation are equivalent to the first order conditions derived from the constraint on the net return and the budget constraint respectively. The corresponding mean-variance problem is illustrated in more detail in the next chapter.

2.2. Market Equilibrium

In order to better understand the impact of liquidity on the market it is useful to obtain the equilibrium conditions implied by the optimisation illustrated

above. Following Jacoby et al. (2000), we aggregate the first order conditions across investors and across assets in order to find an equation for the general equilibrium. We assume that investors are homogeneous.

Aggregating across investors, the first order condition for each asset i becomes:

$$E[R_i(1 - s_{1,i})] - R_f(1 + s_{0,i}) = \lambda COV(W_1, R_i(1 - s_{1,i})) \quad [2.11]$$

Where:

$$\lambda = \sum_j \lambda_j$$

$$W_1 = \sum_j W_{1,j}$$

In equilibrium, all the investors hold the market portfolio, whose weights are denoted by ω_i . The market weights can also be interpreted as the relative capitalisation of each asset. Aggregating across assets using the market weights, we obtain the equilibrium condition for the market portfolio.

$$\begin{aligned} E\left[\sum_k (R_k(1 - s_{1,k})\omega_k)\right] - R_f \sum_k ((1 + s_{0,k})\omega_k) = \\ = \lambda COV(W_1, \sum_k (R_k(1 - s_{1,k})\omega_k)) \end{aligned} \quad [2.12]$$

Dividing [2.11] by [2.12] we obtain:

$$\begin{aligned} \frac{E[R_i(1 - s_{1,i})] - R_f(1 + s_{0,i})}{E\left[\sum_k (R_k(1 - s_{1,k})\omega_k)\right] - R_f \sum_k ((1 + s_{0,k})\omega_k)} = \\ = \frac{COV(W_1, R_i(1 - s_{1,i}))}{COV(W_1, \sum_k (R_k(1 - s_{1,k})\omega_k))} \end{aligned} \quad [2.13]$$

Rearranging, we get the equilibrium condition for the expected net return of asset i :

$$E\left[R_i \frac{(1-s_{1,i})}{(1+s_{0,i})}\right] = R_f + \frac{COV(W_1, R_i(1-s_{1,i}))}{COV(W_1, \sum_k (R_k(1-s_{1,k})\omega_k))} * \\ * \frac{\sum_k ((1+s_{0,k})\omega_k)}{(1+s_{0,i})} \left(E\left[\frac{\sum_k (R_k(1-s_{1,k})\omega_k)}{\sum_i ((1+s_{0,k})\omega_k)}\right] - R_f \right) \quad [2.14]$$

Considering that:

$$\frac{COV(W_1, R_i(1-s_{1,i}))}{COV(W_1, \sum_k (R_k(1-s_{1,k})\omega_k))} \frac{\sum_k ((1+s_{0,k})\omega_k)}{(1+s_{0,i})} = \\ = \frac{COV\left(\frac{\sum_k (R_k(1-s_{1,k})\omega_k)}{\sum_k ((1+s_{0,k})\omega_k)}, \frac{R_i(1-s_{1,i})}{(1+s_{0,i})}\right)}{VAR\left(\frac{\sum_k (R_k(1-s_{1,k})\omega_k)}{\sum_k ((1+s_{0,k})\omega_k)}\right)} \quad [2.15]$$

We can rewrite [2.14] as:

$$E[R_{iN}] = R_f + \beta_{iN} (E[R_{MN}] - R_f) \quad [2.16]$$

Where:

$R_{iN} = (1+r_i) \frac{(1-s_{1,i})}{(1+s_{0,i})}$ is the net return of asset i . Note that the second term

includes the liquidity cost paid when the asset is first purchased at the

beginning of the period and the one paid at the end of the period when the position is closed.

$$R_{MN} = \frac{\sum_k ((1 + r_k)(1 - s_{1,k})\omega_k)}{\sum_k ((1 + s_{0,k})\omega_k)} \text{ is the net return of the market portfolio.}$$

$\beta_{iN} = \frac{COV(R_{MN}, R_{iN})}{VAR(R_{MN})}$ is the measure of systematic risk of asset i adjusted for the liquidity spread.

2.3. The Modified CAPM

As we have seen in the previous section, when liquidity costs are considered the modified CAPM is represented by equation [2.16]. We will now examine the implications of the model outlined above. In order to distinguish between the parameters of a traditional CAPM with no transaction costs and the parameters of the modified CAPM in [2.16], the subscript "N" will be used for the latter (to denote that the parameters refer to net returns). Given that

$$\beta_{iN} = \frac{\sum_k ((1 + s_{0,k})\omega_k)}{(1 + s_{0,i})} \frac{COV\left(\sum_k (R_k(1 - s_{1,k})\omega_k), R_i(1 - s_{1,i})\right)}{VAR\left(\sum_k (R_k(1 - s_{1,k})\omega_k)\right)} \quad [2.17]$$

While

$$\beta_i = \frac{COV\left(\sum_k (R_k\omega_k), R_i\right)}{VAR\left(\sum_k (R_k\omega_k)\right)} \quad [2.18]$$

There is a non-linear relation between the traditional β and the modified β_N . The measure of systematic risk, β_N , includes the uncertainty regarding the future spread, which is an integral part of the systematic risk itself. If in the market the equilibrium were dictated by a modified CAPM, the relationship between gross expected returns and β would not be linear since gross returns incorporate also a liquidity premium. This is clearly illustrated by rewriting the modified CAPM, and isolating the expected gross return². Using [2.14] and decomposing the expression for the expected net return we get:

$$E[R_i] = R_f \frac{1 + s_{0,i}}{1 - E[s_{1,i}]} + \frac{COV(R_i, s_{1,i})}{1 - E[s_{1,i}]} + \beta_{iN} \frac{(1 + s_{0,i})}{1 - E[s_{1,i}]} (E[R_{MN}] - R_f) \quad [2.19]$$

Moreover, since

$$E[R_{MN}] = E\left[\frac{\sum_k (R_k (1 - s_{1,k}) \omega_k)}{\sum_k ((1 + s_{0,k}) \omega_k)}\right] \leq E[R_M] = E\left[\sum_k (R_k \omega_k)\right] \quad [2.20]$$

The risk premium per unit of systematic risk is lower than suggested by the traditional CAPM. A comparison between the standard CAPM and the modified version, which includes a random spread, is illustrated in Figure 2.1.

² Since the uncertainty regarding the future spread is embedded within the measure of systematic risk, it is not possible to explicitly express the expected gross return as a function of the traditional β . Given that the relation between β and β_N is nonlinear, the relation between the expected gross return and β is also nonlinear.

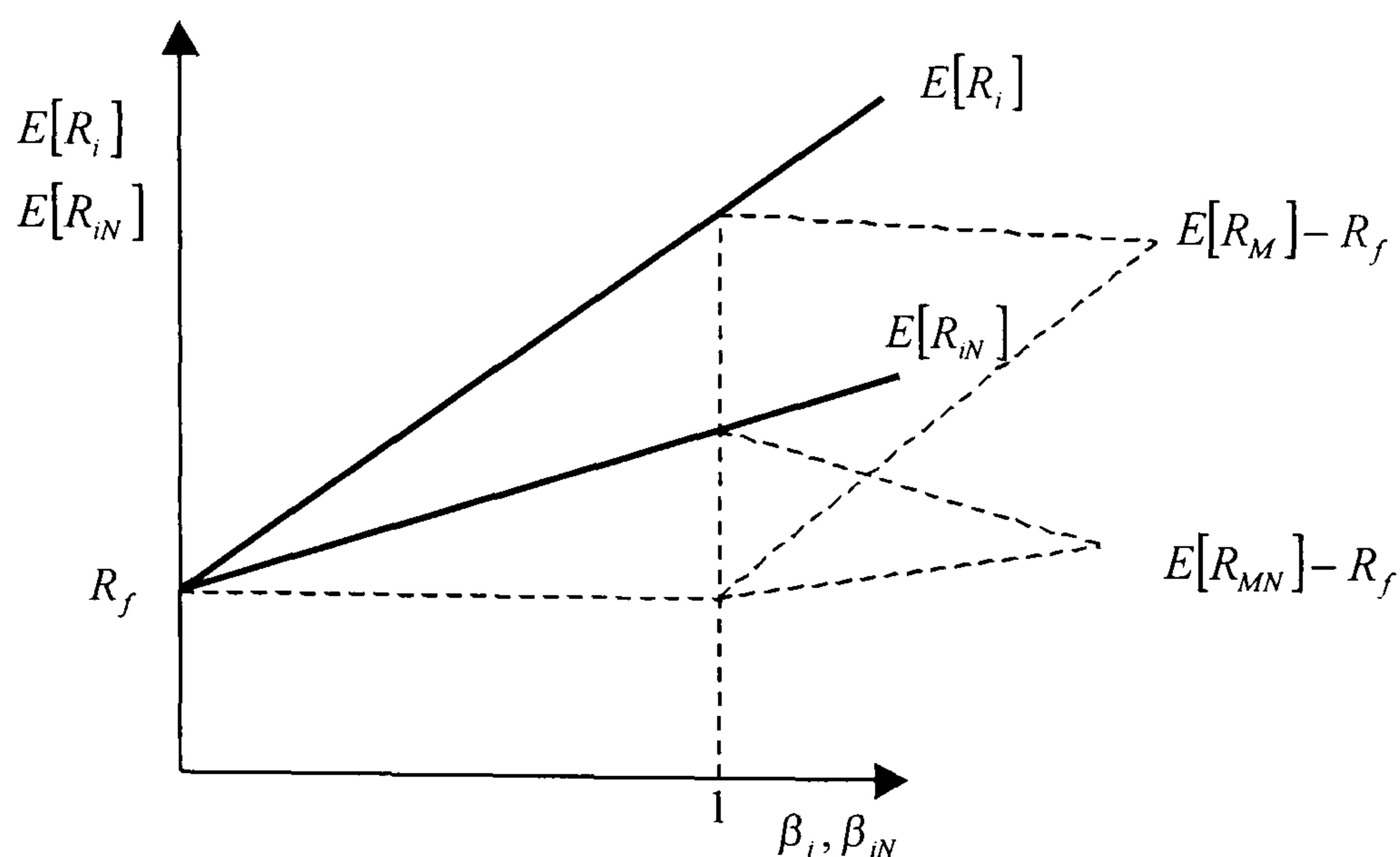


Figure 2.1: Relation between expected return and systematic risk with and without the liquidity effect³.

Equation [2.19] seems to explain empirical results better than the traditional CAPM. While the CAPM received early empirical support, it was subsequently challenged on the basis of incompleteness (Litzenberger and Ramaswamy 1980, Banz 1981). More recently, Fama and French (1992), suggested that the measure of systematic risk was not statistically significant in explaining the cross-sectional variation of expected returns. As an alternative they proposed a model where the firm size and the book-to-market ratio are also included in the regression. This model proved to be a better estimator of the expected gross return. Merton (1987) has also drawn similar conclusions.

It must be noted though that the size effect can be interpreted as a proxy of liquidity. As shown by Amihud and Mendelson (1989), once the liquidity effect is accounted for (the authors use the bid-ask spread as a proxy), the significance of other factors is extremely low. Also Datar et al. (1998), and

³ Both curves appear to be linear since each one is plotted against a corresponding set of parameters. The mapping from one parameter space to the other is nonlinear and therefore if both curves were plotted using the same coordinate system only one of them would be linear. The difference in slope is due to the fact that the risk premium for the modified CAPM is smaller.

Brennan et al. (1998) show that trading volume explains returns better than the size of the firm.

The reason why early tests (Black et al 1972, Fama and MacBeth 1973) confirmed the validity of the CAPM might be explained by the fact that those tests were based on highly liquid stocks (NYSE), while later tests also included securities from the AMEX and the NASDAQ. Also, when longer time periods are used β is statistically significant (Kothari et al. 1995). This result is expected, since, for longer holding periods, liquidity cost has a reduced impact. Finally, it must be noted that when there are transaction costs, not all security price anomalies will be adjusted, due to the lack of excess return. As a result the security market line will no longer be a line, but a range on either side of the SML.

It is possible to analyse the relationship between the gross expected return and the expected future bid-ask spread using [2.19]. If we assume that the sum of spreads is a martingale, the expected end of period spread cost for the market portfolio is insensitive to changes in the expected end of period spread ratio for a single security. In this case, it turns out that the first and second derivatives of the gross return with respect to the expected future spread are both positive. Furthermore, as expected, there is a vertical asymptote when the expected spread approaches one (this would imply that the ask price is twice the mid price and the bid price is zero). The relationship between the future random spread and the gross expected return is illustrated in Figure 2.2.

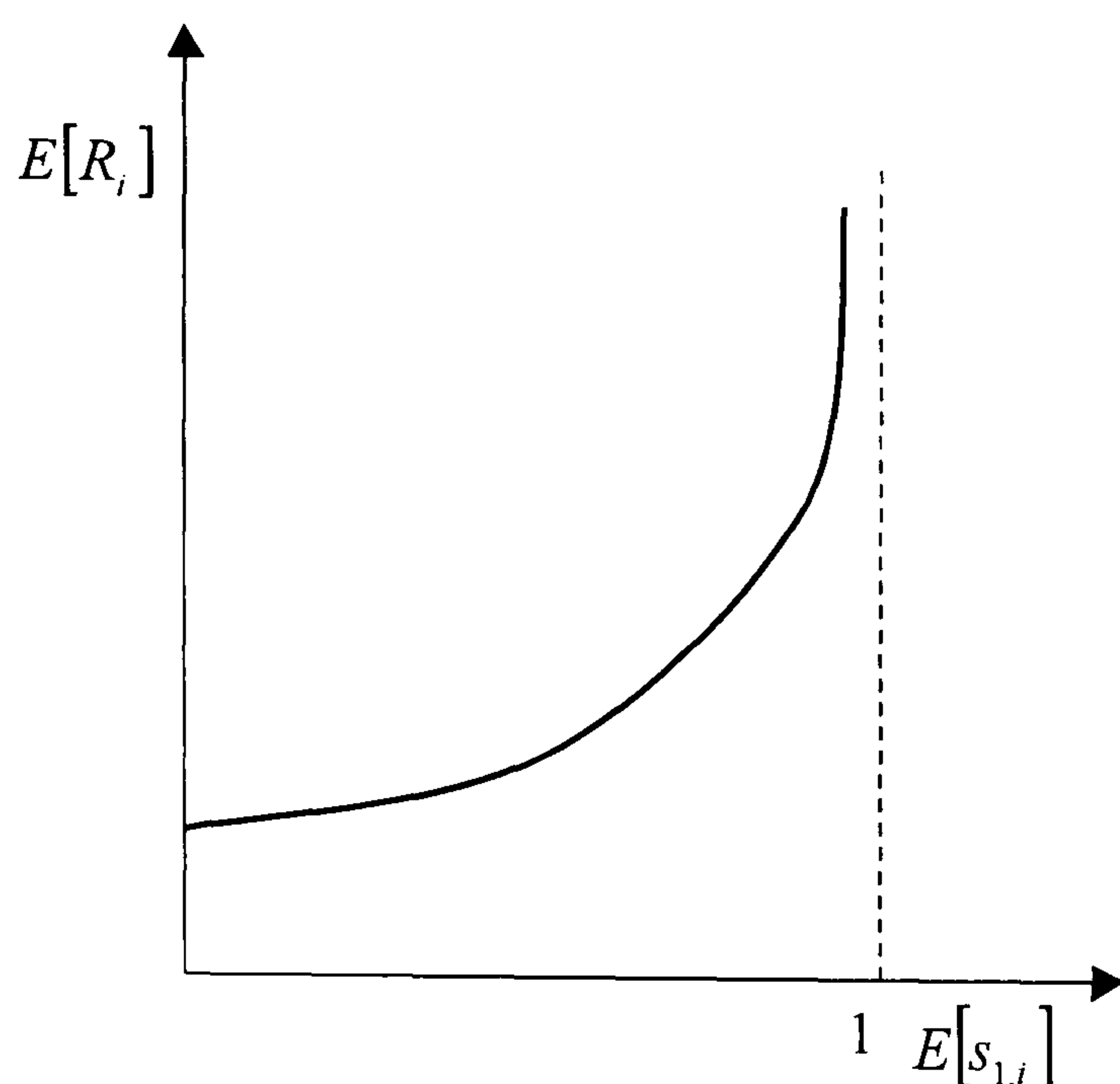


Figure 2.2: Relation between the expected gross return and the expected future relative spread.

In the paper of Amihud and Mendelson (1986) a concave relation is obtained. This is because the authors assume that the more liquid assets are traded more often. Therefore investors demand a higher liquidity premium for a given change in the relative spread for the more active assets. A concave relation might be realistic only for relatively low spreads. For less liquid assets, as the spread approaches 100%, the liquidity cost tends to infinity and therefore beyond a given threshold the model will be convex.

2.4. Including Liquidity Constraints into the Black and Litterman Model

Considering the results obtained in the previous section, it is possible to extend the Black and Litterman (1992) model so as to cope with net (of the liquidity cost) returns.

Black and Litterman suggest that the expected return vector should be estimated using the CAPM model. Their argument is that the optimal portfolio

weights can be extremely sensitive to small changes in expected returns. The underlying cause is the high correlation between many asset classes. To an optimiser they appear as reasonable substitutes. In other words, the extreme weights resulting from the traditional MV optimisation are due to a particular structure of the covariance matrix that is not offset by an appropriate expected return vector. Given the difficulty in estimating the expected return vector, often the results of a MV optimisation are unstable and difficult to interpret.

Black and Litterman describe a Bayesian approach to portfolio construction. Instead of using raw expected returns they back out the consensus expected returns, defined as equilibrium expected returns and calculated according to each asset beta, and then move from these consensus forecasts toward the raw expected forecast in proportion to the information content of the model. When equilibrium expected excess returns are estimated according to the CAPM, higher betas are compensated by proportionally higher expected returns. By using Bayesian statistics, investor prior information is then incorporated into the expected returns by tilting the equilibrium point in a direction consistent with the investor prior information and their confidence in their forecasts.

The authors assume that the market return, R_M , and the vector of market weights ω are known. Usually the vector of market weights used is the relative market capitalisation of each asset. Keeping ω constant, the traditional MV problem is then solved for the vector of equilibrium expected excess returns R_E . Using matrix notation the standard MV problem is:

$$\underset{\omega}{Min} \omega^T \Sigma \omega \quad [2.21]$$

s.t.

$$\omega^T R_E - R_M = 0 \quad [2.22]$$

$$\omega^T e = 1 \quad [2.23]$$

Where

ω is the vector of market weights.

Σ is the covariance matrix of the gross returns.

R_M is the gross excess return of the market portfolio.

R_E is the vector of gross expected excess returns.

e is a vector of ones.

Considering the variables ω as constants and solving for R_E , the first order conditions can be reduced to

$$R_E = R_M \frac{\Sigma \omega}{\sigma_M^2} = R_M \beta \quad [2.24]$$

This is the CAPM when excess returns over the risk-free rate are considered. Indeed, $\Sigma \omega$ is the vector of covariances between each asset and the market portfolio.

By construction, feeding the vector of equilibrium expected excess returns into a standard mean-variance optimiser and solving for the optimal weights w^* , the solution that maximises the Sharpe ratio would imply $w^* = \omega$. The advantage of using the CAPM as an estimator of the vector of expected excess returns is that when such values are used and the investor's views are incorporated, the vector of optimal weights obtained from an optimisation is more stable.

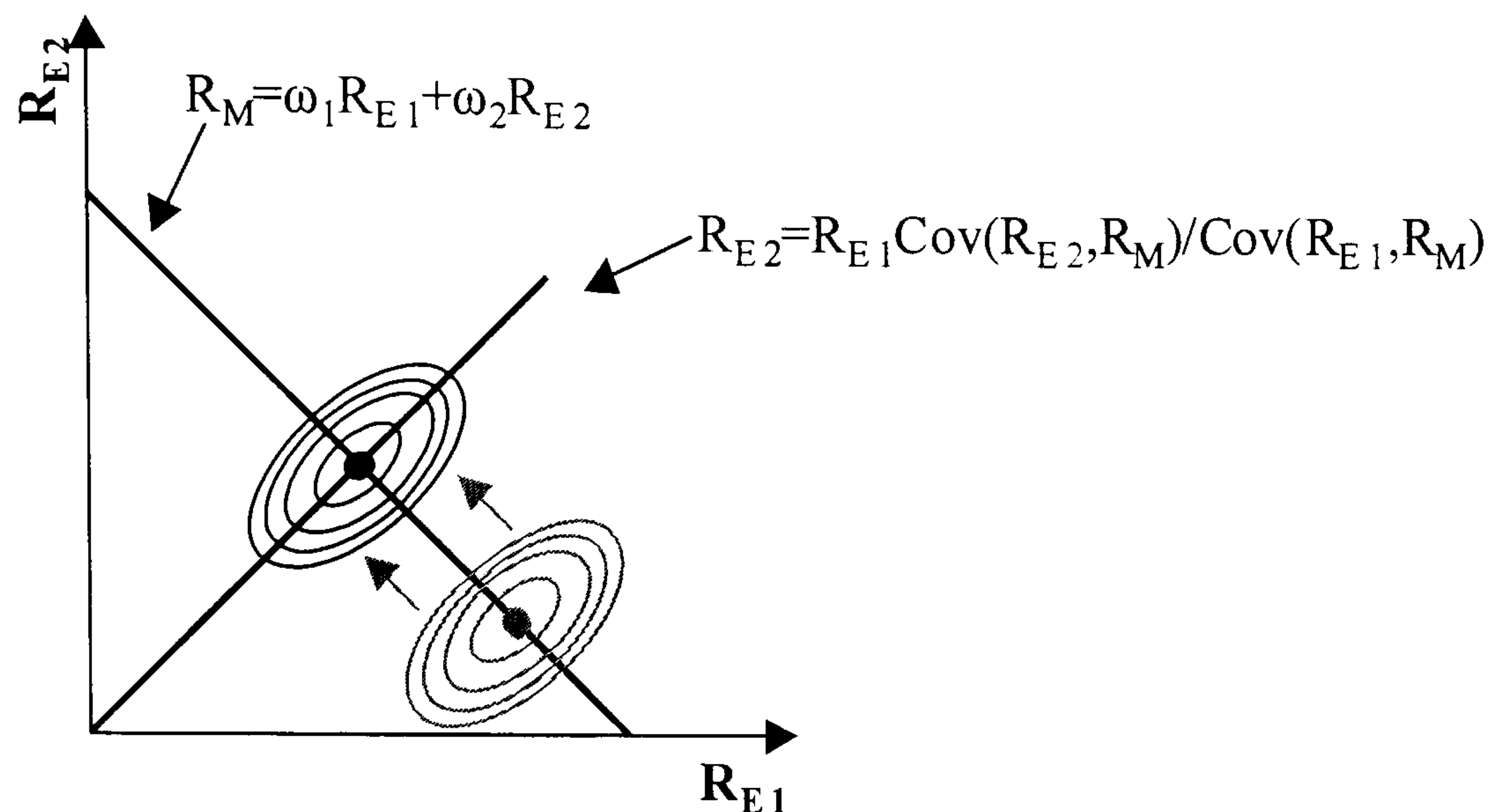


Figure 2.3: Implied market equilibrium excess return.⁴

The model implies that each expected return is proportional to the systematic risk of the corresponding asset, and, therefore, the vector of expected returns is consistent with the structure of the covariance matrix. The process is illustrated in Figure 2.3. Furthermore there is a section of the efficient frontier (around the market portfolio) in which all the assets have positive weights. These two properties guarantee that the assets are not under/overweighted in the final allocation.

Investor views are expressed as a probability distribution of a linear combination of the assets' expected excess returns:

$$\Psi E[R] = \varphi + \varepsilon \quad [2.25]$$

Where ε follows a normal distribution, $N(0, \Omega)$, and Ω is a diagonal matrix with diagonal elements representing the uncertainty of the investor's views,

⁴ In the Black and Litterman model R_M , ω , and the covariance matrix, Σ , are known. The figure shows the effect of the equilibrium conditions for a 2-asset portfolio. The returns are distributed among assets according to their systematic risk.

and φ is a vector whose elements represent the misalignment between the equilibrium excess returns and the investor's view.

For instance, if an investor had the view that asset 1 will outperform asset 2 by 2%, we have to set $\Psi(1,1)=1$, $\Psi(1,2)=-1$, $\varphi(1)=2\%$. The investor confidence in such a view will be expressed by an appropriate number, $\Omega(1,1)$. The vector of expected returns to be used in the MV optimisation is therefore given by the expected excess returns conditional on the equilibrium risk premia, R_E , the covariance matrix of asset returns Σ , Ψ , φ , and Ω .

$$F(E[R]|R_E, \Sigma, \Psi, \varphi, \Omega) \rightarrow N(A, B) \quad [2.26]$$

Where

$$B = [\Sigma^{-1} + \Psi^T \Omega^{-1} \Psi]^{-1} \quad [2.27]$$

and

$$A = B[\Sigma^{-1} R_E + \Psi^T \Omega^{-1} \varphi] \quad [2.28]$$

When liquidity costs are considered, by using the modified CAPM obtained in [2.16], the Black and Litterman model could be extended in order to directly estimate the net equilibrium expected returns, provided that the covariance matrix of net returns is also used. In this case we obtain a vector of net returns and therefore a standard mean-variance optimisation can be used since the transaction costs are embedded within the expected return estimates.

3. Liquidity and Asset Allocation

In this chapter we present a one period asset allocation model which is a modification of the traditional mean-variance problem. The model is capable of coping with relatively complex liquidity cost structures. Under some conditions the model is shown to be equivalent to the utility maximisation problem. Levy and Markowitz (1984) showed empirically that the ordering of portfolios by the mean-variance rule is almost identical to the order obtained by using expected utility for various utility functions and historical distributions of returns. Also a study comparing alternative utility functions by Kallberg and Ziemba (1983) showed that similar absolute risk-aversion indices give similar optimal compositions, regardless of the functional form and parameters of the utility function. Hence, mean-variance analysis is justified for any general concave utility function of the Von Neumann-Morgenstern type. The simplicity of the mean-variance paradigm allows us to concentrate our attention on the effects of liquidity. The model is first presented in section 3.1. In section 3.2, the liquidity cost function is discussed in further detail. In section 3.3, we illustrate the impact of transaction costs using numerical examples. In section 3.4, a heuristic solution is proposed when fixed costs are included. In section 3.5, the portfolio rebalancing problem is analysed.

3.1. Mean-Variance Model

As we have seen in the previous chapter, in order to be consistent with the utility maximisation approach, net returns should be used. Most of the existing literature incorporates transaction costs only within the budget constraint, disregarding their effect on the expected return and on the risk measure of the

portfolio returns. In a one-period world this might lead to a misestimation of the risk involved. In our model the full impact of liquidity costs is considered. We use the same set of assumptions as in chapter 2.1. For simplicity the risk-free asset is not included.

The investor solves the following problem.

$$\underset{w}{Min} VAR[W_1] \quad [3.1]$$

s.t.

$$W_0 = \sum_i w_i (1 + s_{0,i}) = 1 \quad [3.2]$$

$$W_1 = \sum_i w_i (1 + r_i)(1 - s_{1,i}) = \sum_i w_i R_i (1 - s_{1,i}) \quad [3.3]$$

$$E[W_1] = \sum_i w_i + \sum_i w_i E[r_i - s_{1,i} - r_i s_{1,i}] = 1 + r_{PN} = R_{PN} \quad [3.4]$$

$$s_i = \frac{Ask_i - Mid_i}{Mid_i} = \frac{Mid_i - Bid_i}{Mid_i} \quad [3.5]$$

Where R_{PN} is a constant expressing the required expected net return on the portfolio¹. Note that the covariance matrix used in the objective function represents the covariances between net returns, since they include the spread paid when the position is closed. In particular by expressing the objective function by matrix notation we obtain:

$$\begin{aligned} VAR[W_1] &= E[(W_1 - E[W_1])^2] = \\ &= w^T E[(r - s - sr - E[r - s - sr])^2] w = w^T \Sigma_N w \end{aligned} \quad [3.6]$$

Where:

¹ Note that for a nonzero initial spread, $\sum w_i < 1$ since w_i represents the weights net of initial transaction costs and therefore $r_{PN} \neq \sum w_i E[r_i - s_{1,i} - r_i s_{1,i}]$. The net portfolio rate of return is calculated using the initial and final wealth according to the formula $r_{PN} = E[W_1]/W_0 - 1 = E[W_1] - 1 = R_{PN} - 1$.

w is the vector of weights.

r is the vector of gross rates of return $\{r_i\}$.

s is the vector of future spreads $\{s_{1,i}\}$.

sr is the vector representing the liquidity cost paid on capital gains $\{r_i s_{1,i}\}$.

Σ_N is the covariance matrix of net returns.

Whenever the net returns are joint-normally distributed or the utility function is quadratic, it is straightforward to prove equivalence to the utility maximisation problem analysed in the previous chapter.

The assumption of multivariate normality has been severely criticised. In practice, the distribution of individual assets returns tends to exhibit a higher probability of extreme values than is consistent with normality (leptokurtosis). This departure from normality means that distribution moments higher than the first two moments need to be considered to fully describe portfolio behaviour. The resulting optimisation problem often becomes quite complex as it exhibits multiple local extrema. Variable transaction costs imply further irregularities in the objective function, which becomes strongly nonlinear. The analysis of different risk measures is beyond the scope of this thesis but the algorithms proposed are independent of the specific form of the objective function and can be used when the variance is replaced by more complex measures.

3.2. Liquidity Cost Function

In accordance with market microstructure theory we assume that the spread is an increasing function of the transaction size.

As we have seen in the previous chapter the relative spread can not be greater than 1 in absolute terms², therefore the liquidity cost function should be concave (convex) as the transaction size tends to plus (minus) infinity. Using the data from Barra³ we can show the relation between prices and transaction size.

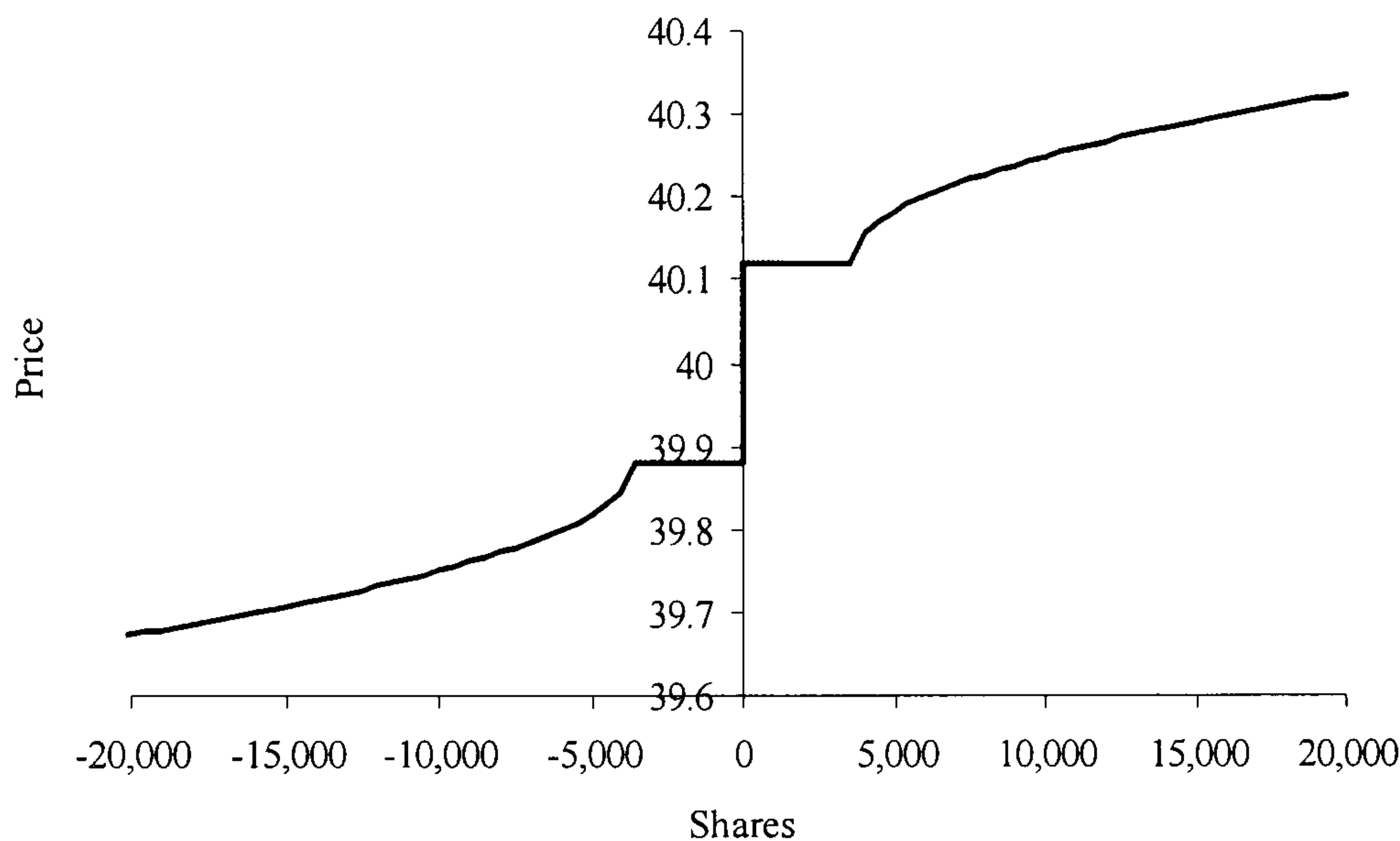


Figure 3.1: Liquidity cost function, representing the actual price paid as a function of the transaction size.

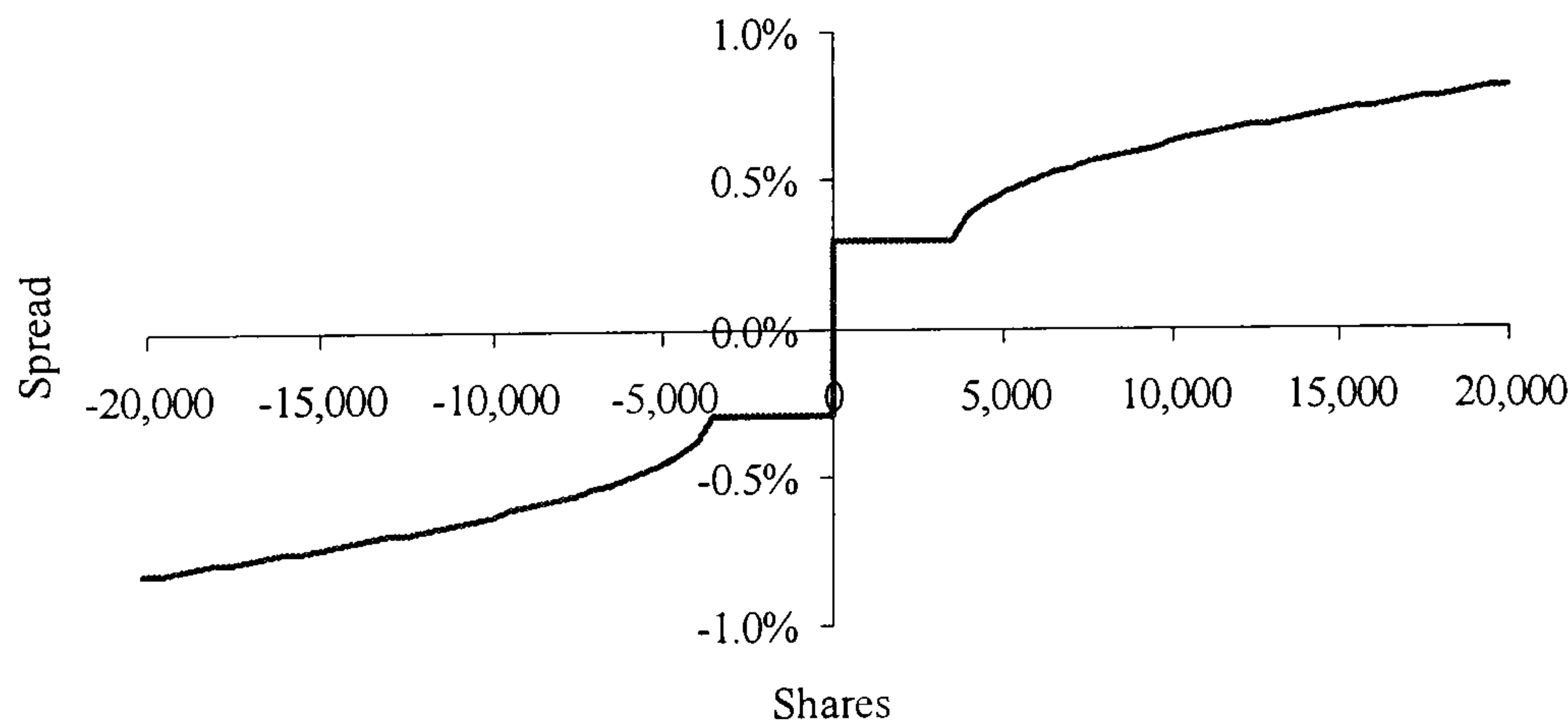


Figure 3.2: Spread as a function of the transaction size.

² In what follows a negative transaction size denotes that the shares are being sold.

³ As mentioned in the introduction, Barra estimates that for a typical transaction of a share worth \$40, the spread is 12¢ for orders within the quoted depth (typically 3,500 shares). For larger orders the spread increases. Barra estimates that for a typical order of 10,000 shares, an additional 13¢ would be paid.

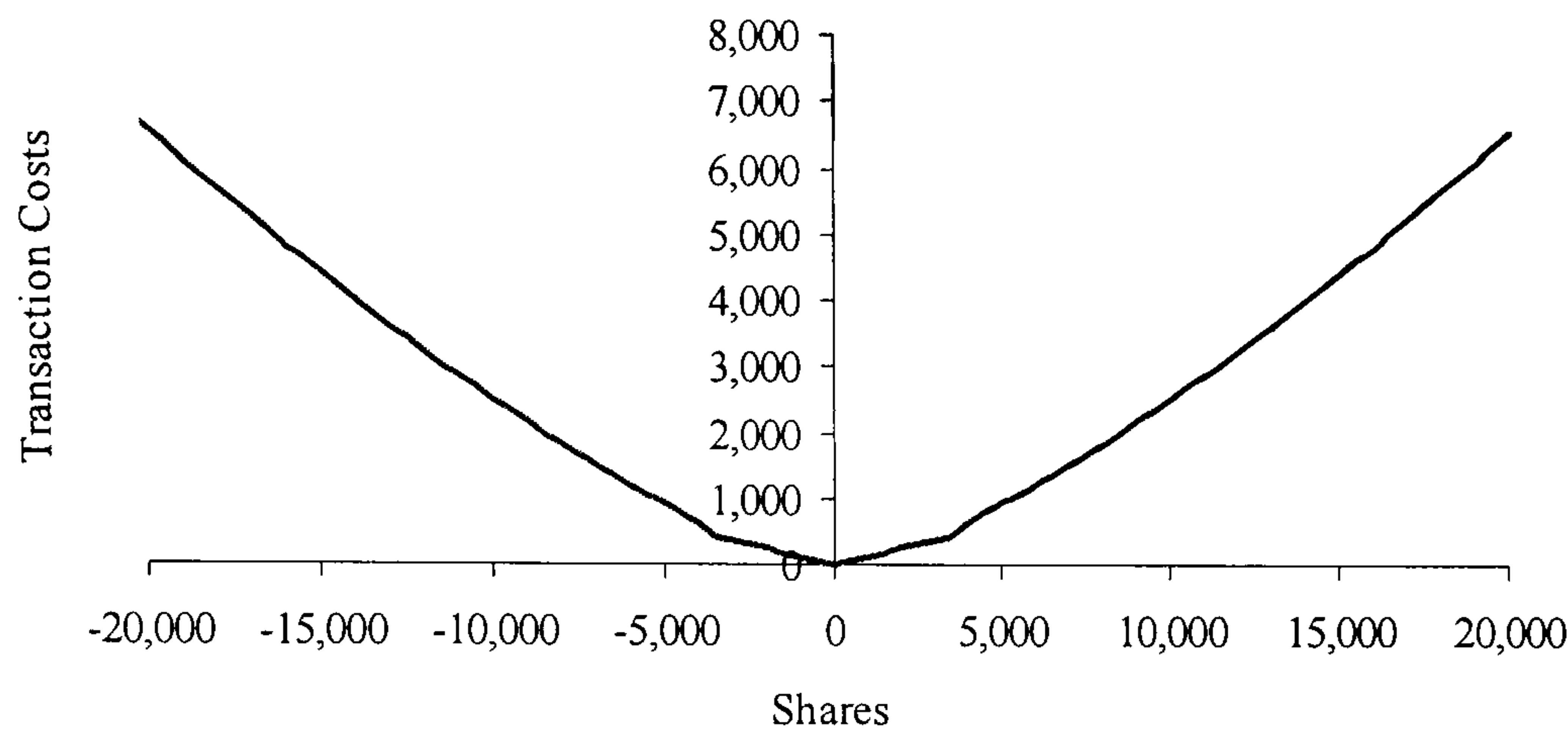


Figure 3.3: Transaction costs as a function of the transaction size.

In order to accommodate these assumptions we need to modify the model presented in the previous section by considering the spread (s_0 and s_1) as a function of the weights⁴. We need to distinguish the gross weights, denoted by g , from the weights net of the initial transaction costs, denoted by w . Given that the spread is an increasing function of the transaction size the total transaction costs will be a pseudo-convex function of the amount traded. The spread function can be characterised as follows:

$$s_{0,i}(g_i) = \begin{cases} s_{0,i}^+(g_i), & g_i \geq 0 \\ s_{0,i}^-(g_i), & g_i < 0 \end{cases} \quad [3.7]$$

$$s_{1,i}(w_i) = \begin{cases} s_{1,i}^+(w_i), & w_i \geq 0 \\ s_{1,i}^-(w_i), & w_i < 0 \end{cases} \quad [3.8]$$

$$w_i = \frac{g_i}{(1 + s_{0,i}(g_i))} \quad [3.9]$$

$$\frac{\partial s_{0,i}^+(g_i)}{\partial g_i} > 0, \quad \frac{\partial s_{0,i}^-(g_i)}{\partial g_i} > 0 \quad [3.10]$$

⁴ In practice, the liquidity cost is a function of the absolute transaction size. Given that in our analysis relative weights are used, the liquidity cost function needs to be modified accordingly.

$$\frac{\partial s_{1,i}^+(w_i)}{\partial w_i} > 0, \quad \frac{\partial s_{1,i}^-(w_i)}{\partial w_i} > 0 \quad [3.11]$$

Where s^+ , and s^- are monotonically increasing functions of the weights, in the range $[0,1)$ and $(-1,0]$ respectively.

If the end period spread is assumed to be stochastic the model can be modified as follows:

$$s_{1,i}(w_i) = \begin{cases} s_{1,i}^+(w_i, \varepsilon_i), & w_i \geq 0 \\ s_{1,i}^-(w_i, \varepsilon_i), & w_i < 0 \end{cases} \quad [3.12]$$

$$\frac{\partial s_{1,i}^+(w_i, \varepsilon_i)}{\partial \varepsilon_i} > 0, \quad \frac{\partial s_{1,i}^-(w_i, \varepsilon_i)}{\partial \varepsilon_i} < 0 \quad [3.13]$$

$$E[\varepsilon_i] = 0, \quad E[\varepsilon_i \varepsilon_j] = \sigma_{\varepsilon_i \varepsilon_j}, \quad E[\varepsilon_i r_j] = 0 \quad \forall i, j \quad [3.14]$$

The random variable ε represents the uncertainty of future liquidity costs. We assume that ε has a 0 expected value and is uncorrelated with the gross rate of return⁵.

The model proposed can be adjusted to cope with negative weights, but, in order to simplify the analysis, we will impose the restriction $g \geq 0$. Given our assumptions the problem then becomes:

$$\text{Min}_g f(w) \quad [3.15]$$

s.t.

$$\sum_i g_i = 1 \quad [3.16]$$

$$w_i = \frac{g_i}{(1 + s_{0,i}(g_i))} \quad [3.17]$$

$$f(w) = VAR[W_1] = VAR\left[\sum_i w_i R_i (1 - s_{1,i}(w_i R_i))\right] \quad [3.18]$$

$$\begin{aligned} E[W_1] &= \sum_i w_i E[R_i (1 - s_{1,i}(w_i R_i))] = \\ &= \sum_i w_i E[(1 + r_i)(1 - s_{1,i}(w_i (1 + r_i)))] = 1 + r_{PN} = R_{PN} \end{aligned} \quad [3.19]$$

$$g_i \geq 0 \quad [3.20]$$

Equation [3.16] and [3.17] represent the budget constraint. In what follows we use a more concise notation as illustrated below.

$$m_i(g_i) = 1 + s_{0,i}(g_i) \quad [3.21]$$

$$n_i(w_i) = R_i (1 - s_{1,i}(w_i R_i)) = (1 + r_i)(1 - s_{1,i}(w_i (1 + r_i))) \quad [3.22]$$

Given that the spread is increasing in the transaction size, the elements in the budget constraint, $m_i(g_i)$, are increasing in g , while the elements in the constraint on the expected net return, $E[n_i(w_i)]$, are decreasing in w . Note also that both the budget constraint [3.16] - [3.17] and the constraint on the expected net return [3.19] are separable.

The covariance matrix Σ_N is no longer constant but depends on the portfolio net weights, w , therefore the objective function is generally not convex. In the simplest case, where the spread is a linear function of g and w , such that $s_{0,i}(g_i) = a_{0,i}g_i$ and $s_{1,i}(w_i) = a_{1,i}w_i(1+r_i)$, the weights net of buy transaction costs are represented by:

⁵ A more comprehensive model might consider the liquidity spread dependent on the volatility of the underlying asset. Therefore $s_i = f(w_i, \sigma_i, \varepsilon_i)$.

$$w_i = \frac{g_i}{m_i(g_i)} = \frac{g_i}{(1 + a_{0,i}g_i)} \quad [3.23]$$

While the holdings net of buy and sell transaction costs are given by the following formula.

$$w_i(1 - a_{1,i}w_i(1 + r_i)) = \frac{g_i}{(1 + a_{0,i}g_i)} - a_{1,i} \frac{g_i^2(1 + r_i)}{(1 + a_{0,i}g_i)^2} \quad [3.24]$$

Note that net holdings given by [3.24] are first increasing and then decreasing in g_i and therefore appropriate restrictions need to be imposed on the parameters a . In particular when short-selling is not allowed, and therefore $g_i \in [0,1]$, the liquidity coefficients need to be such that $1 + a_{0,i} - 2a_{1,i}(1 + r_i) \geq 0$ ⁶.

In this case the objective function is:

$$VAR(W_1) = \gamma^T C(r_i, r_j) \gamma + \beta^T C(r_i^2, r_j^2) \beta - 2\gamma^T C(r_i, r_j^2) \beta \quad [3.25]$$

Where C represents a covariance matrix of the appropriate elements, $\beta_i = a_{1,i}w_i^2$, and $\gamma_i = w_i - 2\beta_i$. The portfolio variance is therefore a fourth order polynomial in w .

Another appropriate function for the spread might be the sigmoid function. In this case,

⁶ The restrictions on the parameters imply that the holdings net of buy and sell transaction costs are a non-decreasing function of the initial gross investment. When the spread is linear in the weights such restrictions make a linear spread function adequate when the gross weights are in the range $[0,1]$ even though the spread does not tend to 1 as the weights tend to infinity.

$$s_{0,i} = \frac{2}{1 + e^{-\alpha_{0,i}g_i}} - 1 \quad [3.26]$$

Where $\alpha_{0,i}$ is a non-negative parameter. The function assumes real values in the range $(-1,+1)$, is monotonically increasing in g , and respects the restrictions [3.7] through [3.11].

In order to model the uncertainty of the future spread either a random parameter is explicitly included into the spread function, or the future spread is assumed to be a random variable with mean s_0 . In the first case we obtain:

$$s_{1,i} = \frac{2}{1 + e^{-\alpha_{1,i}w_i(1+r_i)(1+e_i)}} - 1 \quad [3.27]$$

Alternatively an appropriate distribution can be used. If we assume that s_1 follows a Beta distribution, $s_1 \sim \text{Beta}(0,1,C,D)$, with

$$\text{Density of } s = s^{C-1}(1-s)^{D-1}\Gamma(C+D)/[\Gamma(C)\Gamma(D)]$$

$$\text{Mean} = C/(C+D)$$

$$\text{Variance} = CD/[(C+D+1)(C+D)^2]$$

It is possible to obtain the parameters C and D by setting the mean to $\text{abs}(s_0)$, using an appropriate variance and solving the resulting system, and finally setting $\text{sign}(s_1)=\text{sign}(s_0)$.

3.3. Numerical Examples

To illustrate the effects of liquidity constraints on a traditional Mean-Variance optimisation, we consider the small four-asset example from Chang et al.

(2000)⁷, with the following gross expected rates of return, standard deviations and correlations:

Asset #	Correlations				Exp. RoR	Std. Dev.
	1	2	3	4		
1	1				0.000659	0.030586
2	0.164589	1			0.003174	0.030474
3	0.099763	0.083122	1		0.001377	0.035770
4	0.118368	0.143822	0.252213	1	0.004798	0.046351

Table 3.1: Four-asset example.

In what follows we analyse the simple case where the spread is linear in the weights and deterministic, such that $s_{0,i}(g_i) = a_{0,i}g_i$ and $s_{1,i}(w_i) = a_{1,i}w_i(1+r_i)$, as illustrated in the previous paragraph. We call a_i the liquidity coefficient, interpreted as the spread paid when 100% of the portfolio is invested in asset i .

Note that in order to gain a better insight into the problem structure, we will often use extremely high liquidity coefficients. Such coefficients imply high transaction costs compared to the gross expected rates of return and therefore we can expect net expected rates of return to be negative. This is an obvious violation of the assumption of investors being rational. In what follows we will therefore assume that investors are forced to select a portfolio even when the rate of return is negative.

Let us first consider a portfolio of two assets. We use the first two assets from Table 3.1. Let us also assume that the liquidity coefficient is 3% when we purchase the assets and 5% when we sell them at the end of the period, for each asset. The expected rate of return and standard deviations for two portfolios (the first one with 100% invested in asset 1, and the second one with 50% invested in each asset) are shown in Table 3.2.

⁷ Weekly data drawn from the FTSE data set over the period March 1992 - September 1997, as in Chang et al. (2000).

	Weights			Exp. RoR		Std. Dev.	
	Gross	Net of Buy	Net of Buy and Sell	Gross	Net	Gross	Net
Asset 1	100.00%	97.09%	92.37%	0.07%	-7.57%	-	-
Asset 2	0.00%	0.00%	0.00%	-	-	-	-
Portfolio 1	100.00%	97.09%	92.37%	0.07%	-7.57%	3.06%	2.68%
Asset 1	50.00%	49.26%	48.05%	0.07%	-3.84%	-	-
Asset 2	50.00%	49.26%	48.04%	0.32%	-3.61%	-	-
Portfolio 2	100.00%	98.52%	96.09%	0.19%	-3.72%	2.33%	2.18%

Table 3.2: Two-asset portfolios.

The weights net of initial transaction costs (buy costs) are calculated according to [3.23], while for the weights net of buy and sell costs [3.24] is used. Since the future spread is assumed to be deterministic, the standard deviation is calculated using [3.25]. Note that transaction costs affect both the standard deviation and the expected rate of return. Also note that the standard deviation of the portfolio is smaller with non-zero transaction costs. This is because the future spread is assumed to be deterministic and the effect on the standard deviation is only due to a reduction in the net holdings (see [3.25]). If the future spread is assumed to be stochastic there are two opposite effects, one due to a reduction in the net holdings and one due to a change of the net covariance matrix. The overall effect on the portfolio standard deviation depends in this case on the specific assumptions. Figure 3.4-Figure 3.8 show the results for feasible combinations of two assets. In Figure 3.8 the effect on the net expected rate of return is clearly dominant, since, by reducing extreme weights, the impact of transaction costs is attenuated.

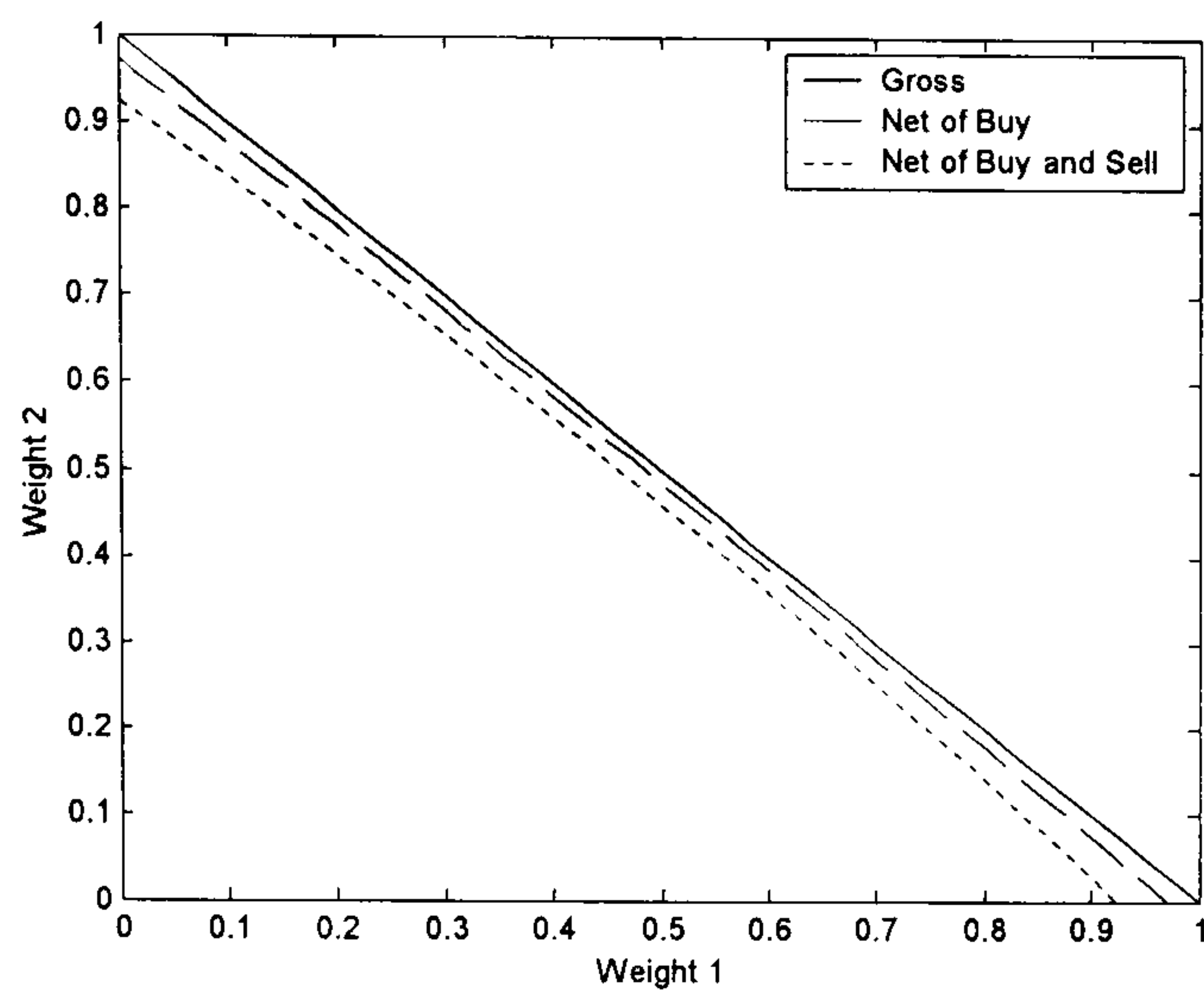


Figure 3.4: Two-asset portfolio weights.

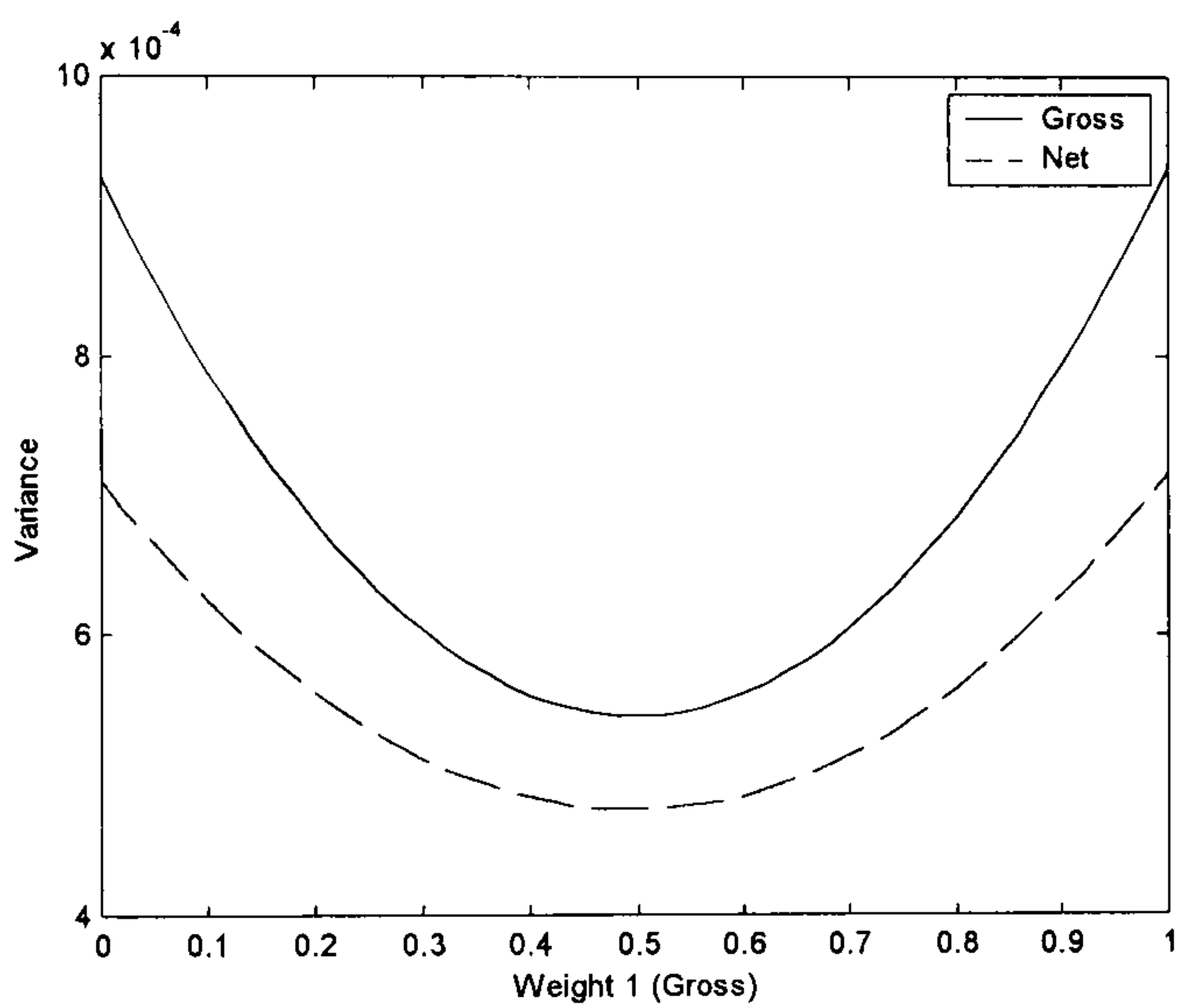


Figure 3.5: Variance.

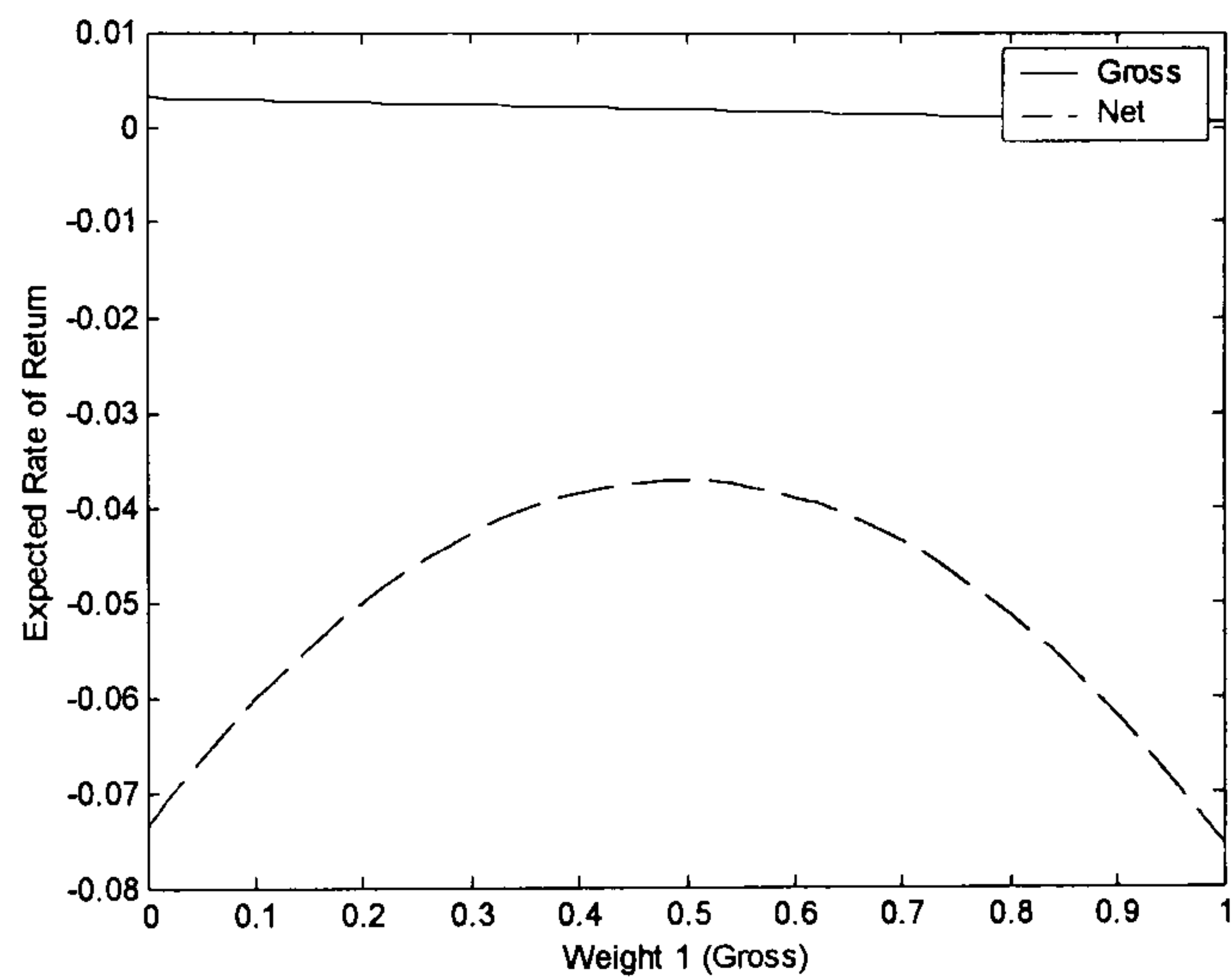


Figure 3.6: Expected rate of return.

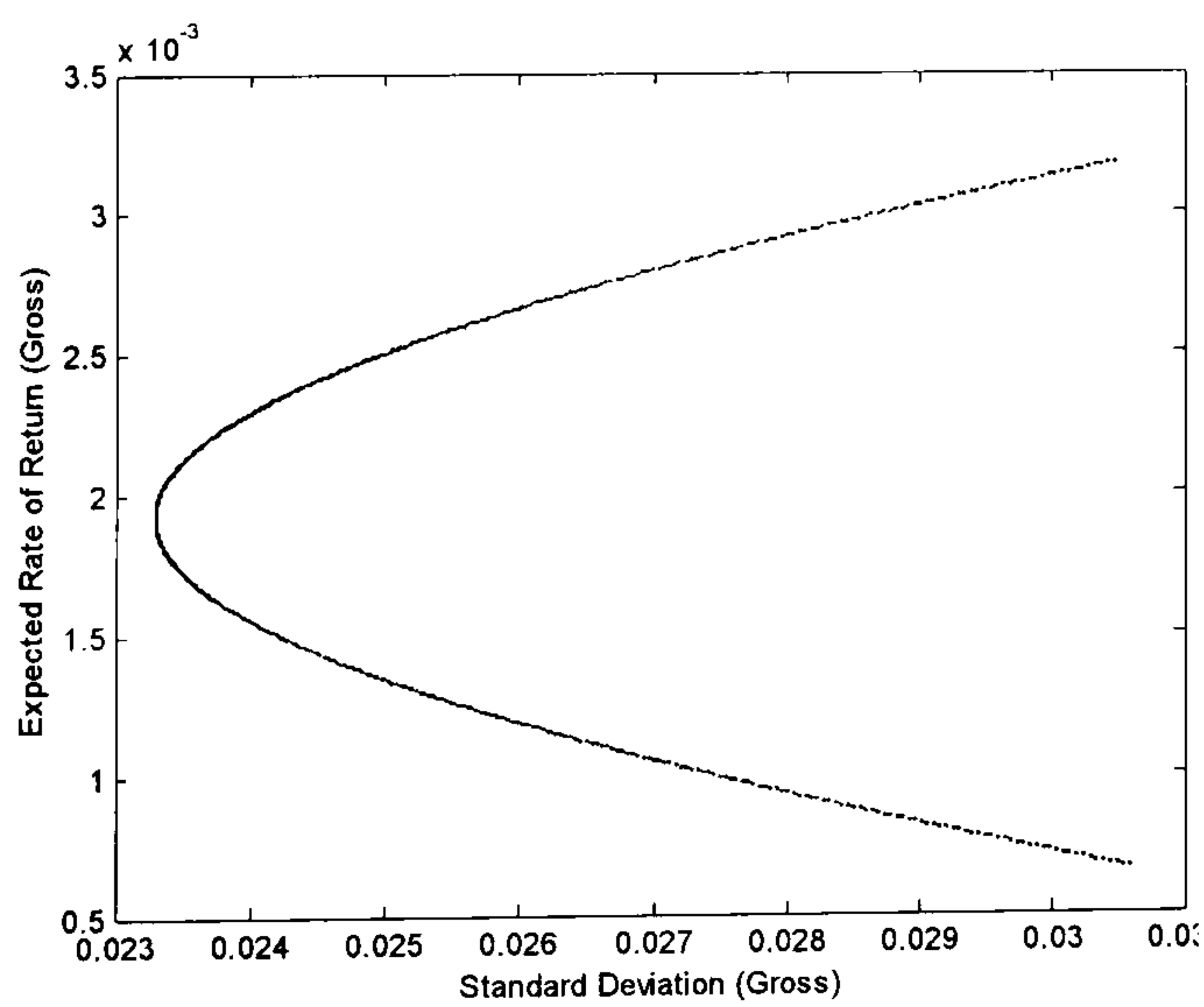


Figure 3.7: Feasible combinations of two assets (gross).

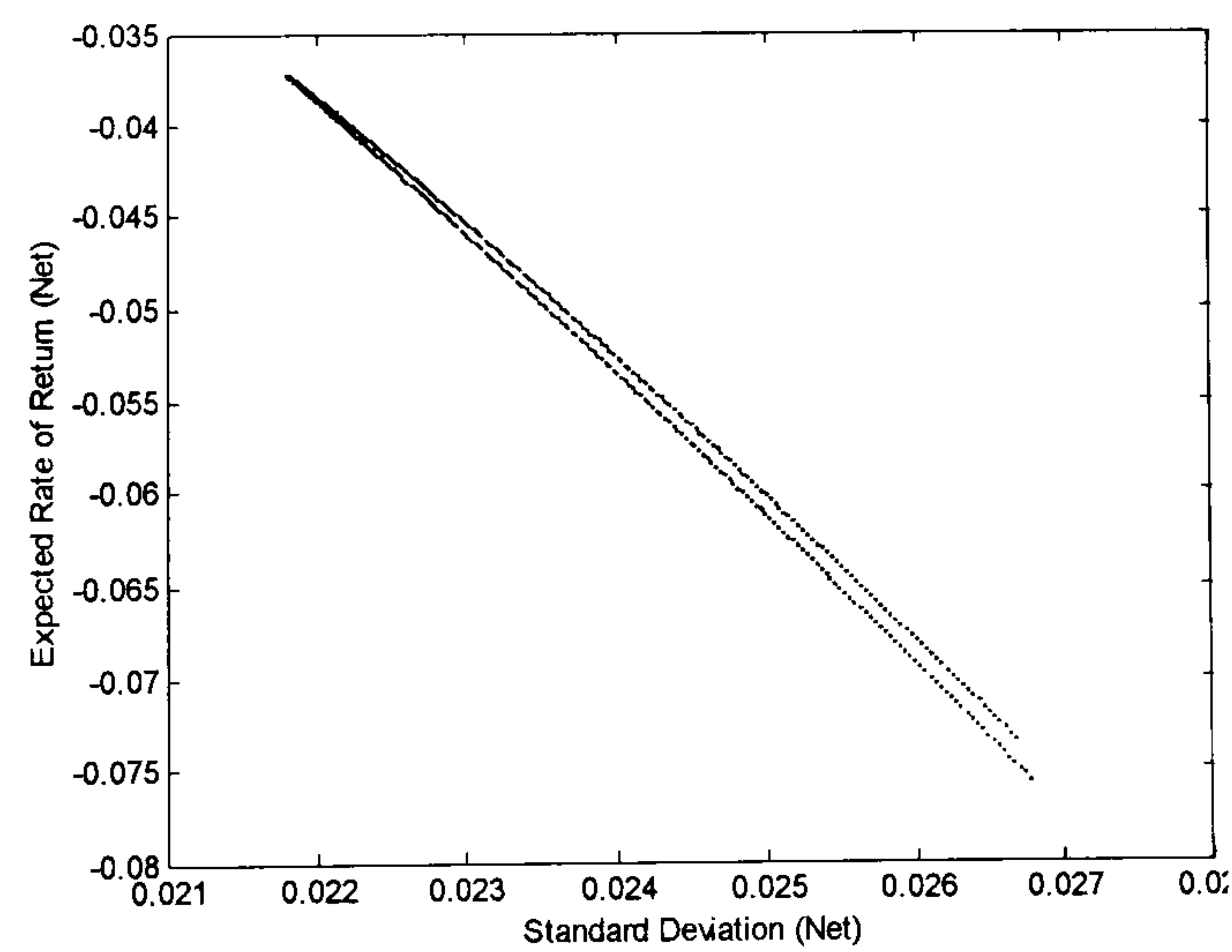


Figure 3.8: Feasible combinations of two assets (net).

We now consider the extreme case where the initial liquidity coefficient is 10% and the liquidity coefficient for the end period is 25% (therefore less than the maximum value allowed given our restrictions on the weights). The impact of such a liquidity structure for two-asset portfolios is illustrated in Figure 3.9-Figure 3.10.

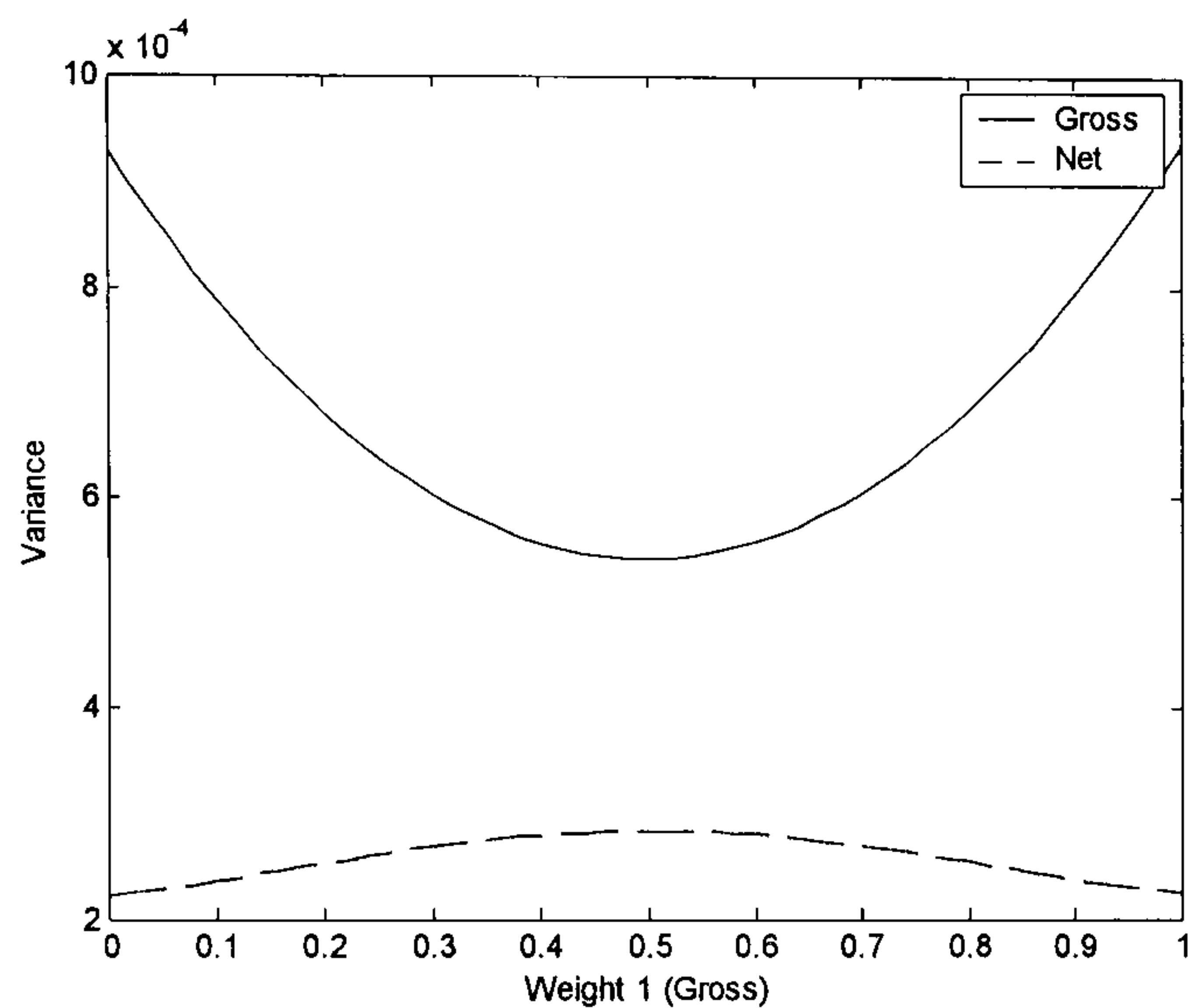


Figure 3.9: Variance.

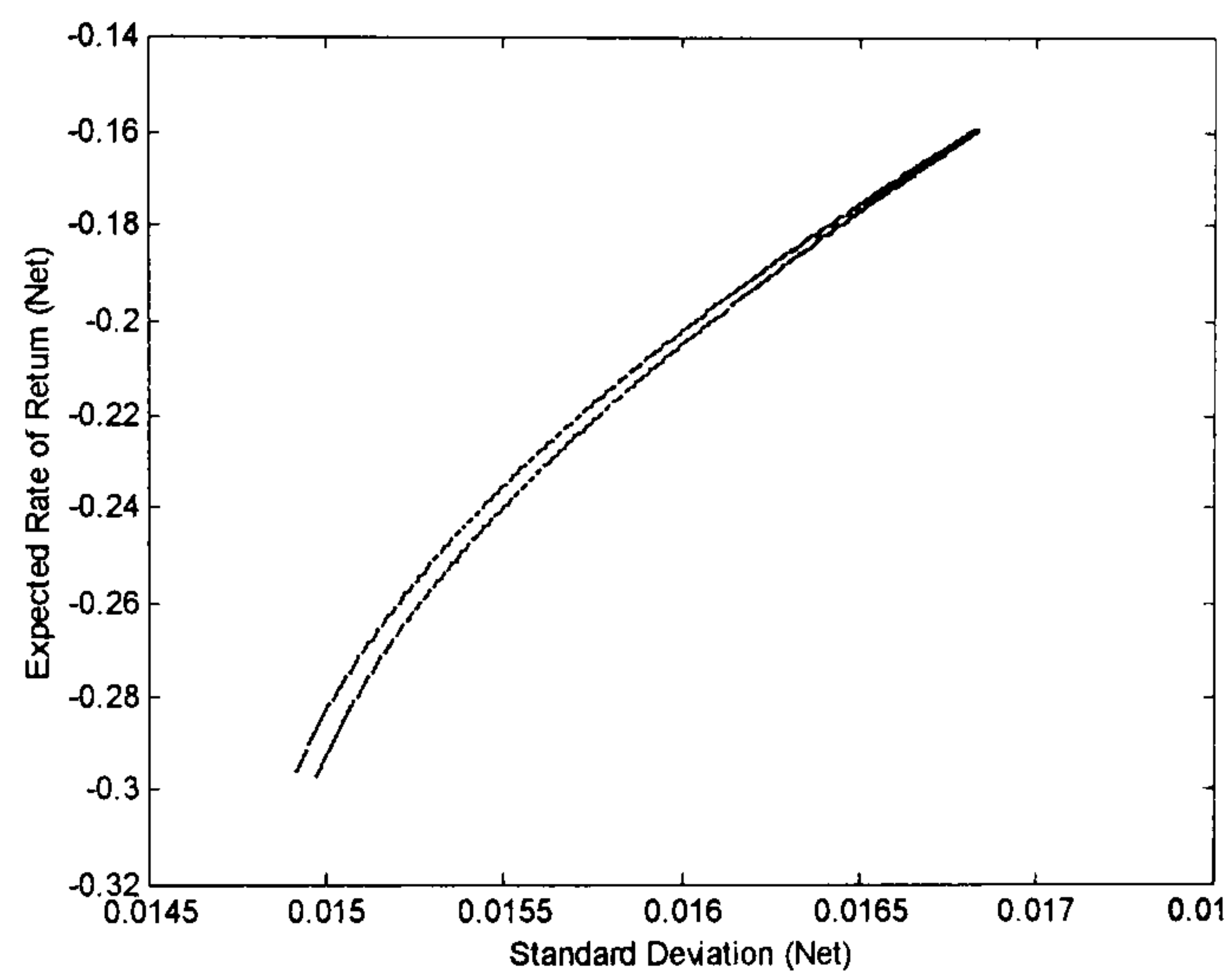


Figure 3.10: Feasible combinations of two assets (net).

The results might appear counterintuitive. The net expected rate of return of a portfolio is not a simple convex combination of the expected rates of return of the underlying assets, but there is an additional non-negative term given by the reduction of transaction costs⁸. Consequently, with fixed positive liquidity coefficients, the net expected rate of return of a portfolio is a concave function. At the same time, since, in this case, the effect of transaction costs on the variance is only due to the reduction in net holdings, by reducing extreme weights and therefore the transaction costs, net holdings are increased and this can in turn increase the variance of the portfolio.

The results for an liquidity coefficients of 10 and 25% using the first three assets from Table 3.1 are illustrated in Figure 3.11-Figure 3.16.

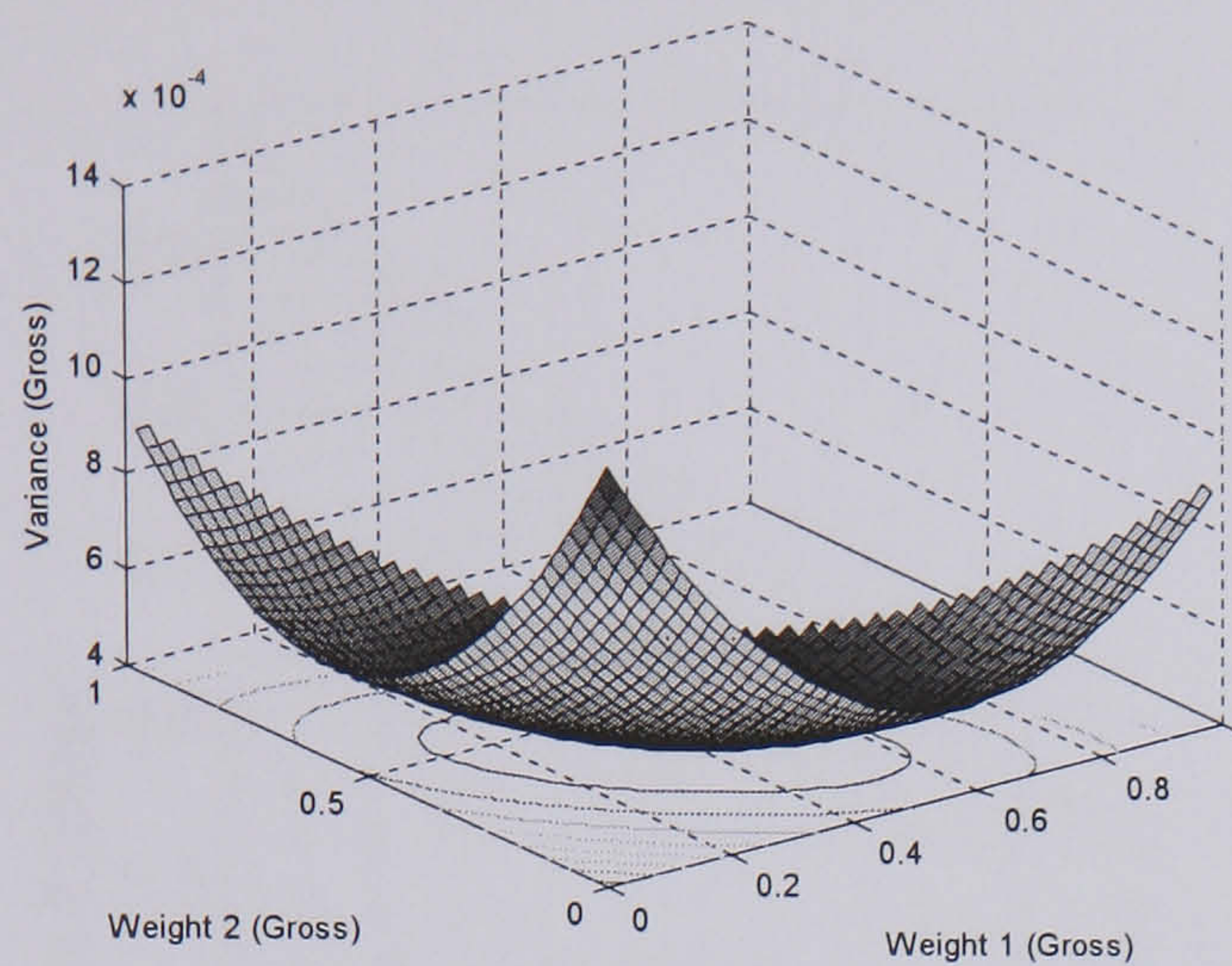


Figure 3.11: Variance (gross).

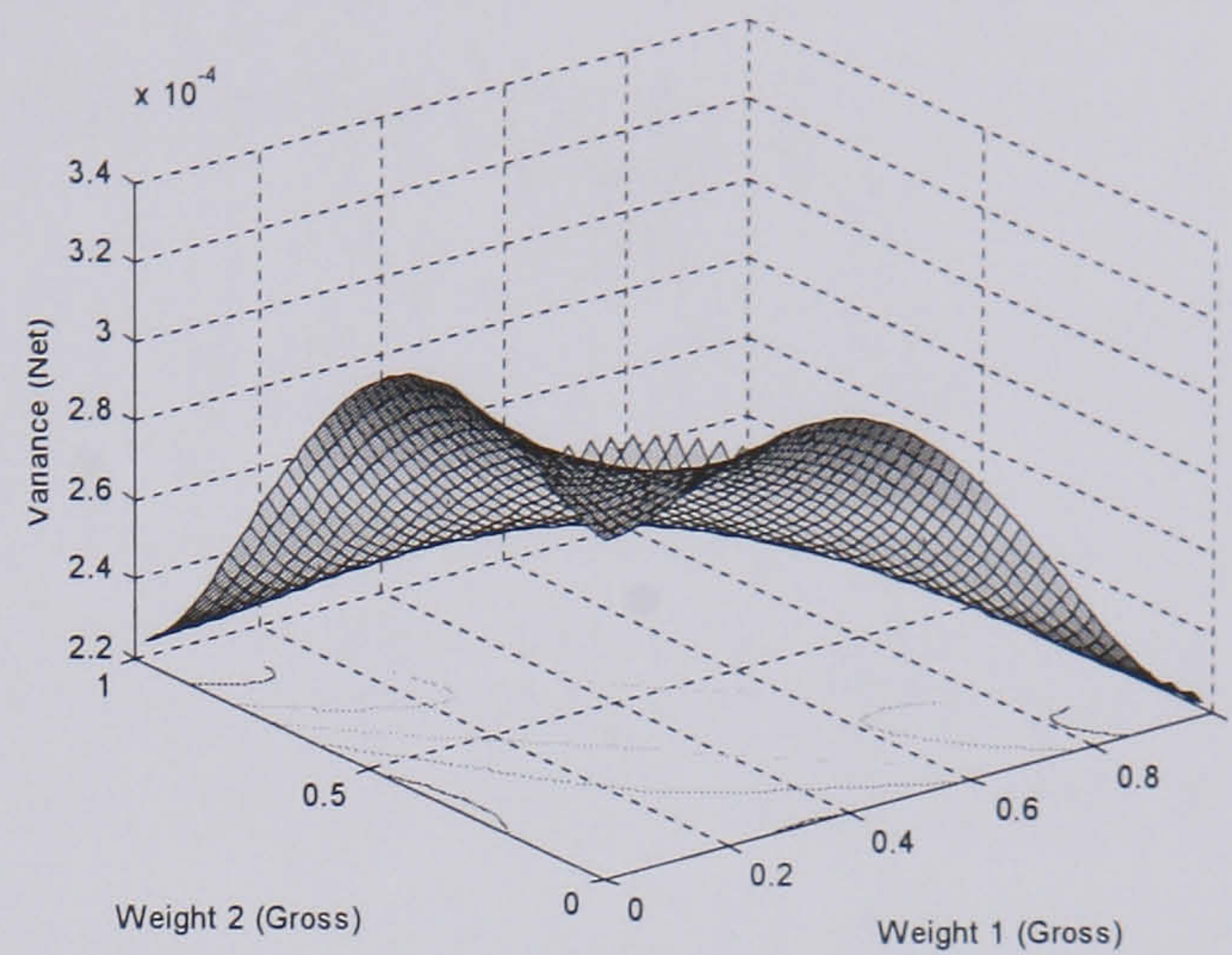


Figure 3.12: Variance (net).

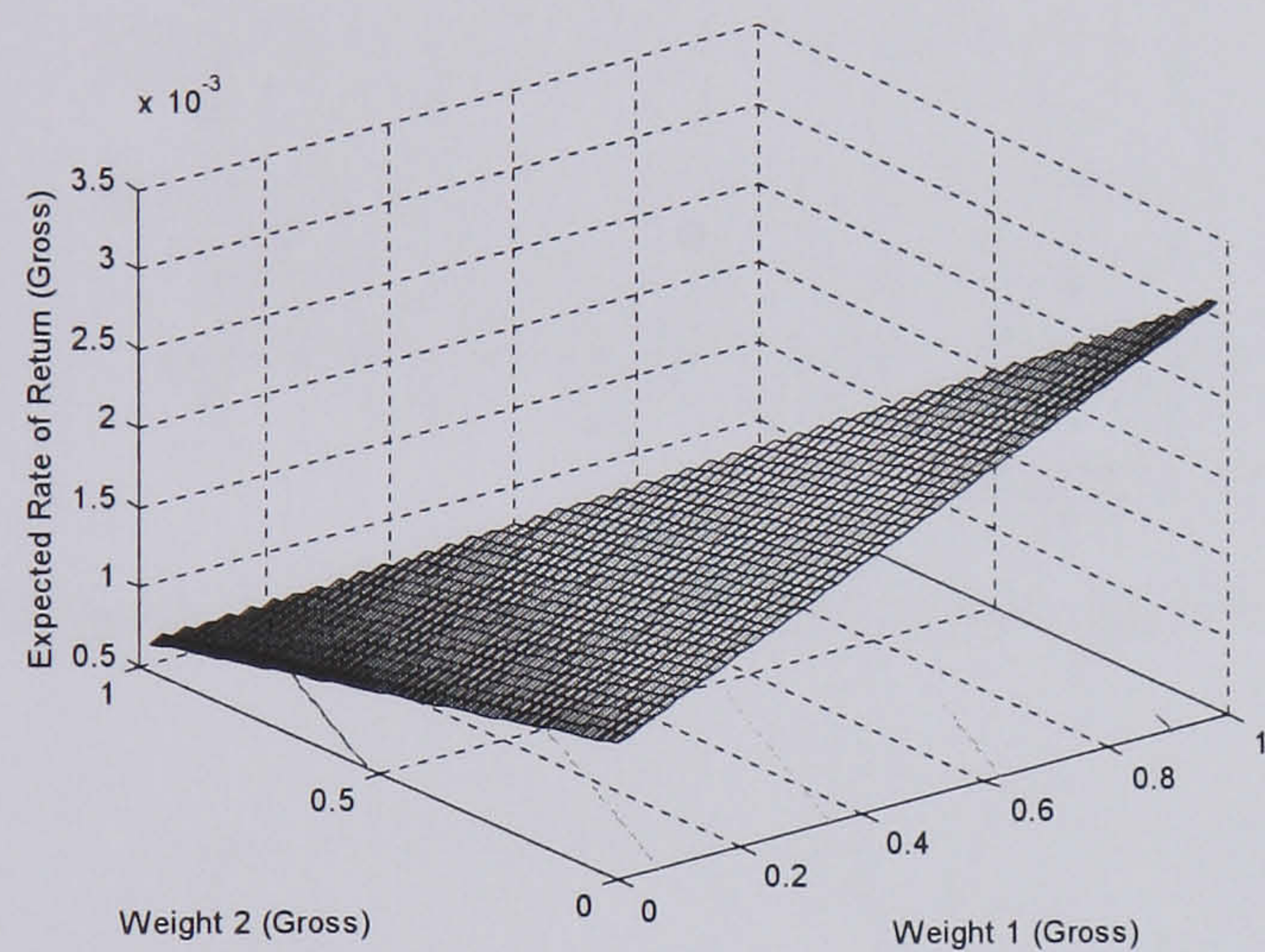


Figure 3.13: Expected rate of return (gross).

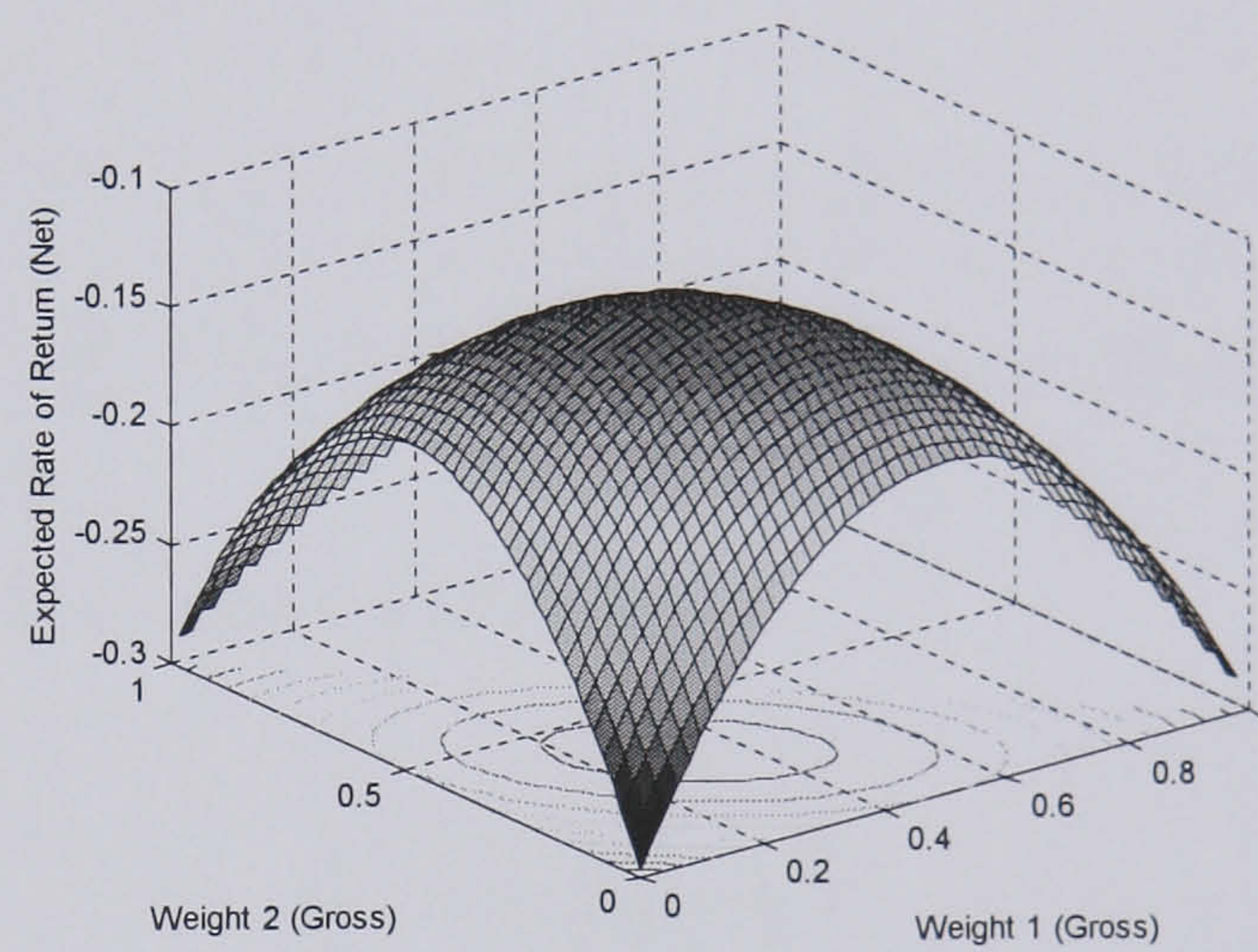


Figure 3.14: Expected rate of return (net).

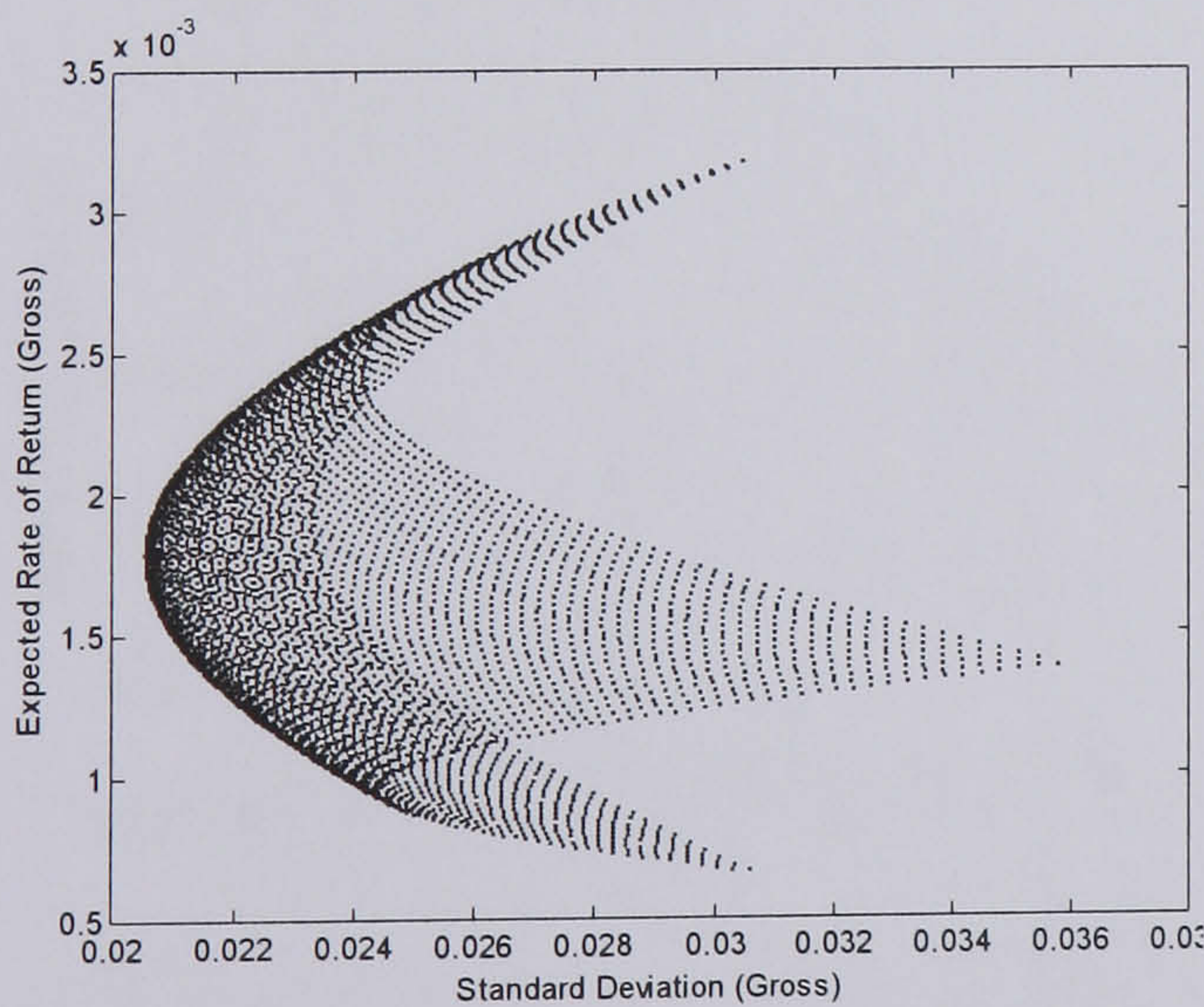


Figure 3.15: Feasible combinations of three assets (gross).

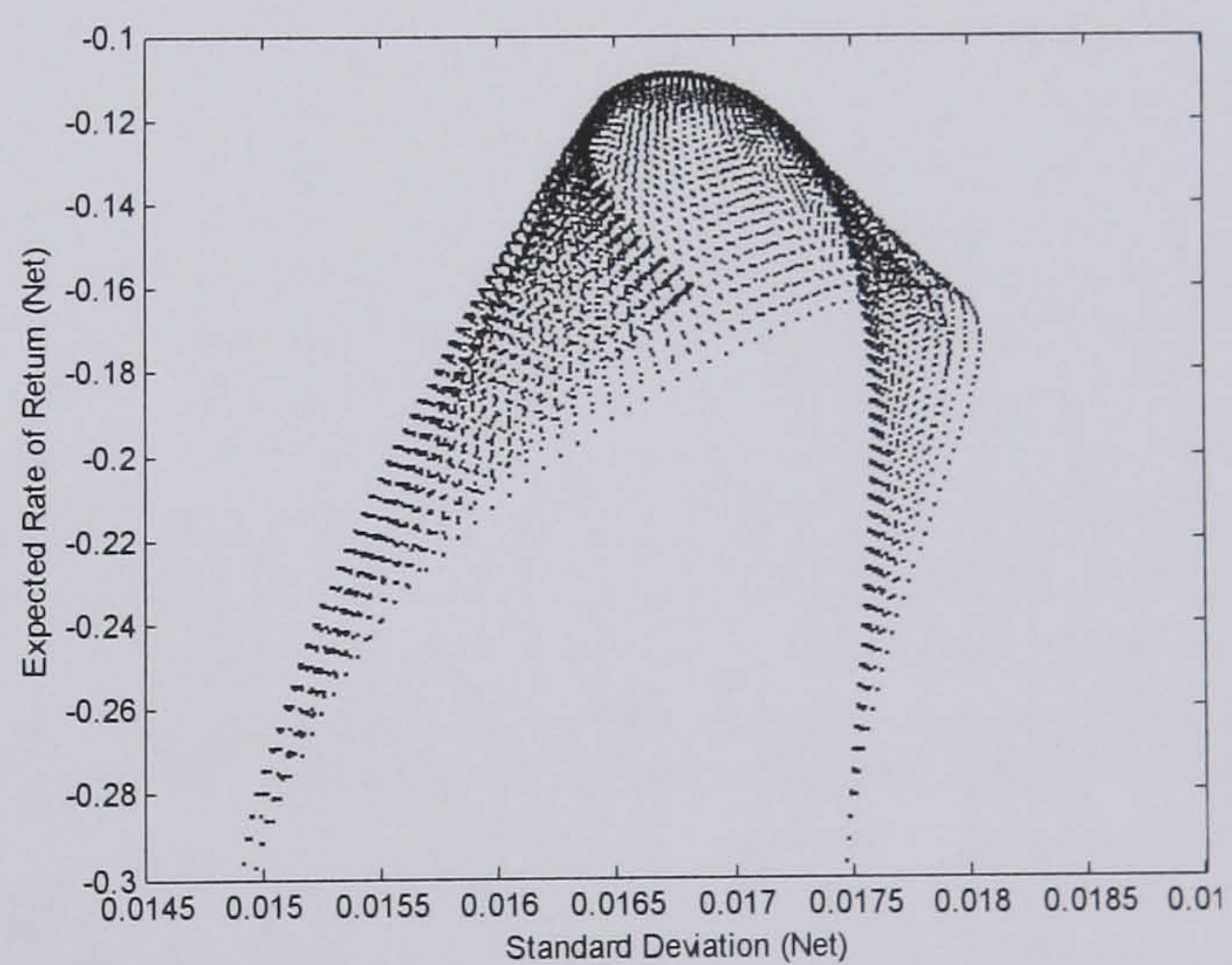


Figure 3.16: Feasible combinations of three assets (net).

⁸ In the problem considered, since the transaction costs are much larger than the gross expected return the impact of such an additional term is dominating.

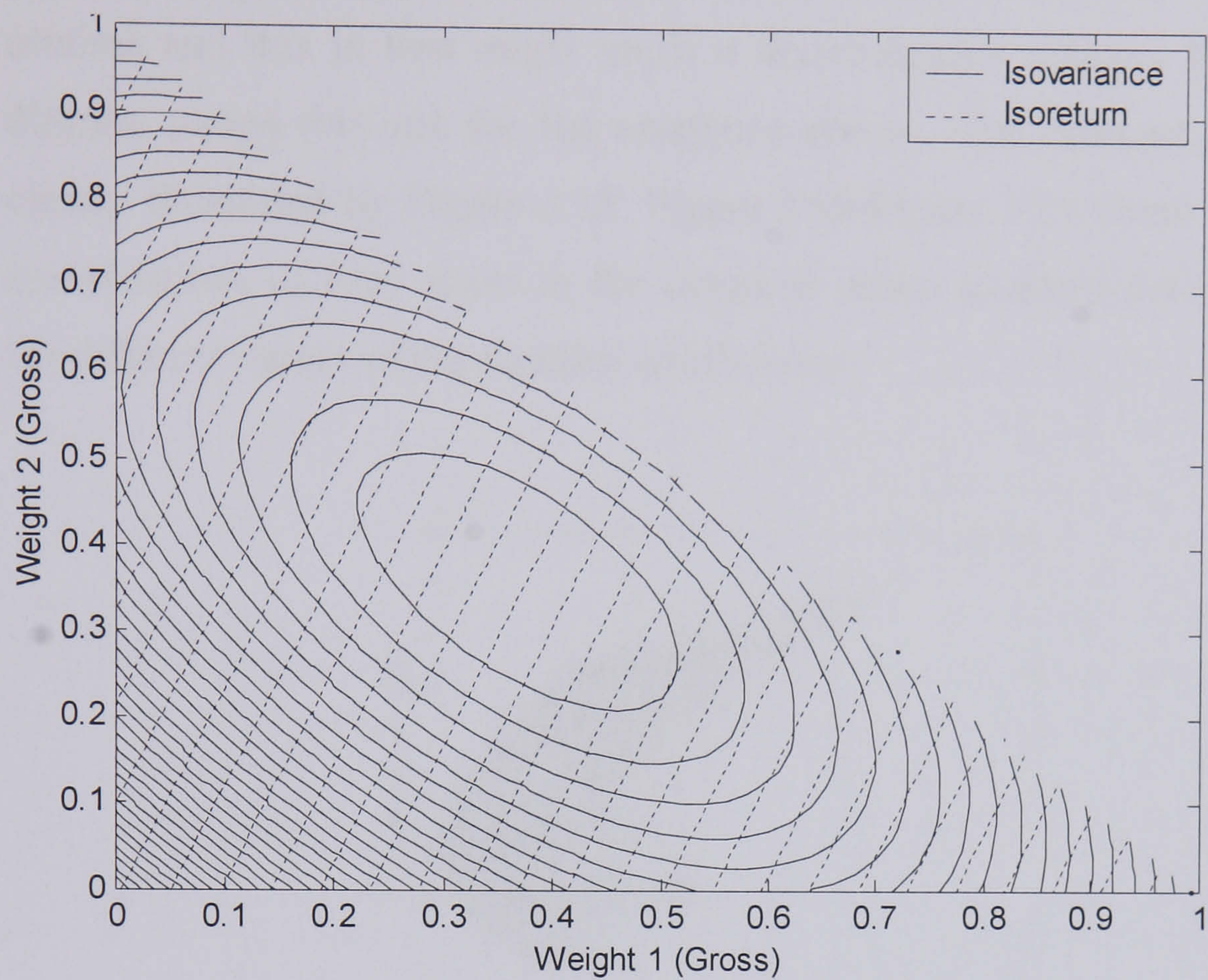


Figure 3.17: Isovariance and isoreturn lines for three-asset portfolios (gross).

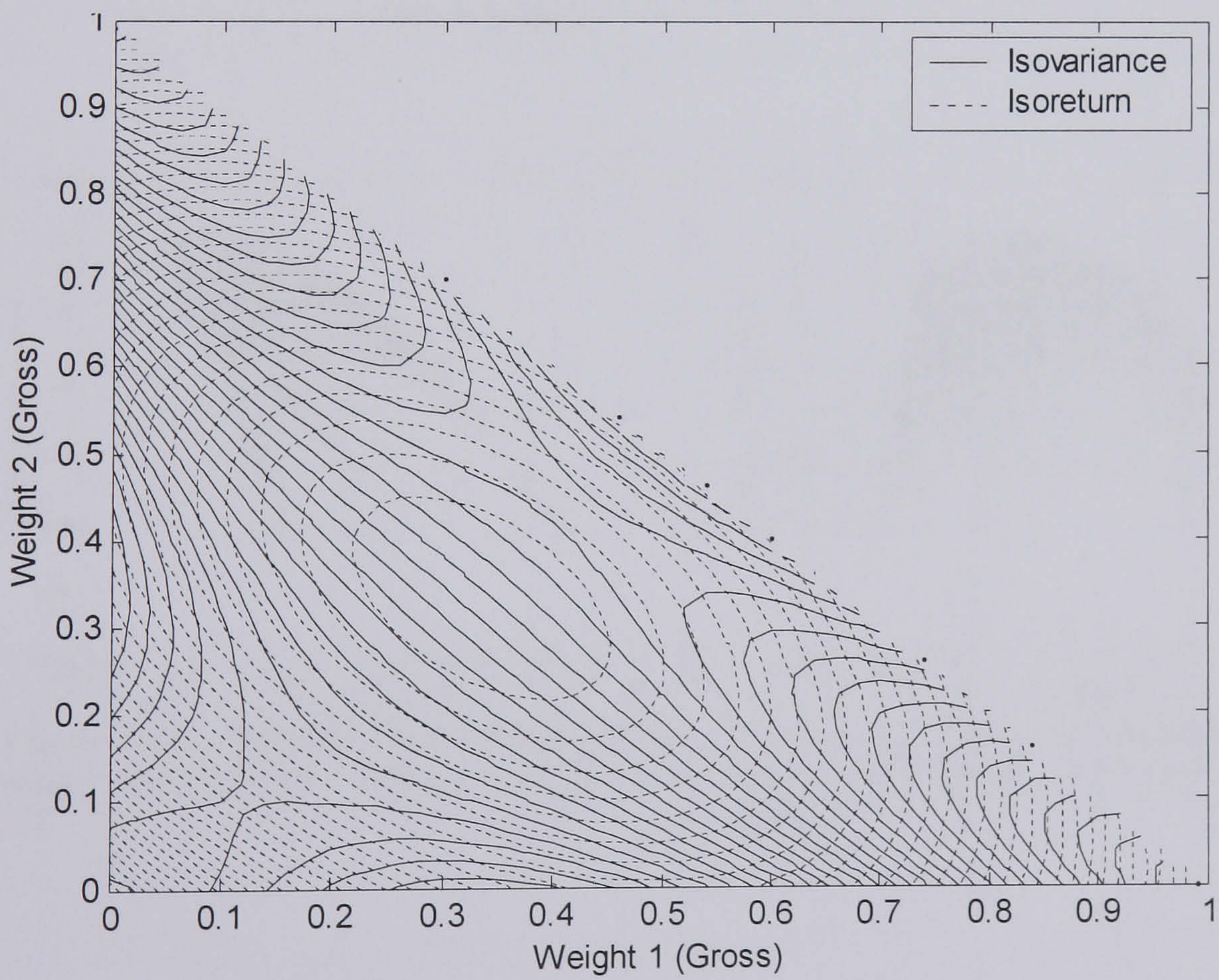


Figure 3.18: Isovariance and isoreturn lines for three-asset portfolios (net).

In this case the distortions in the variance structure imply multiple local minima and this in turn might imply a discontinuous efficient frontier with discrete jumps (but not for the examples above). The problem structure is clearly illustrated by Figure 3.18. Figure 3.19-Figure 3.21 show the feasible combinations of four assets in the expected return-standard deviation plane, for different values of the liquidity coefficients.

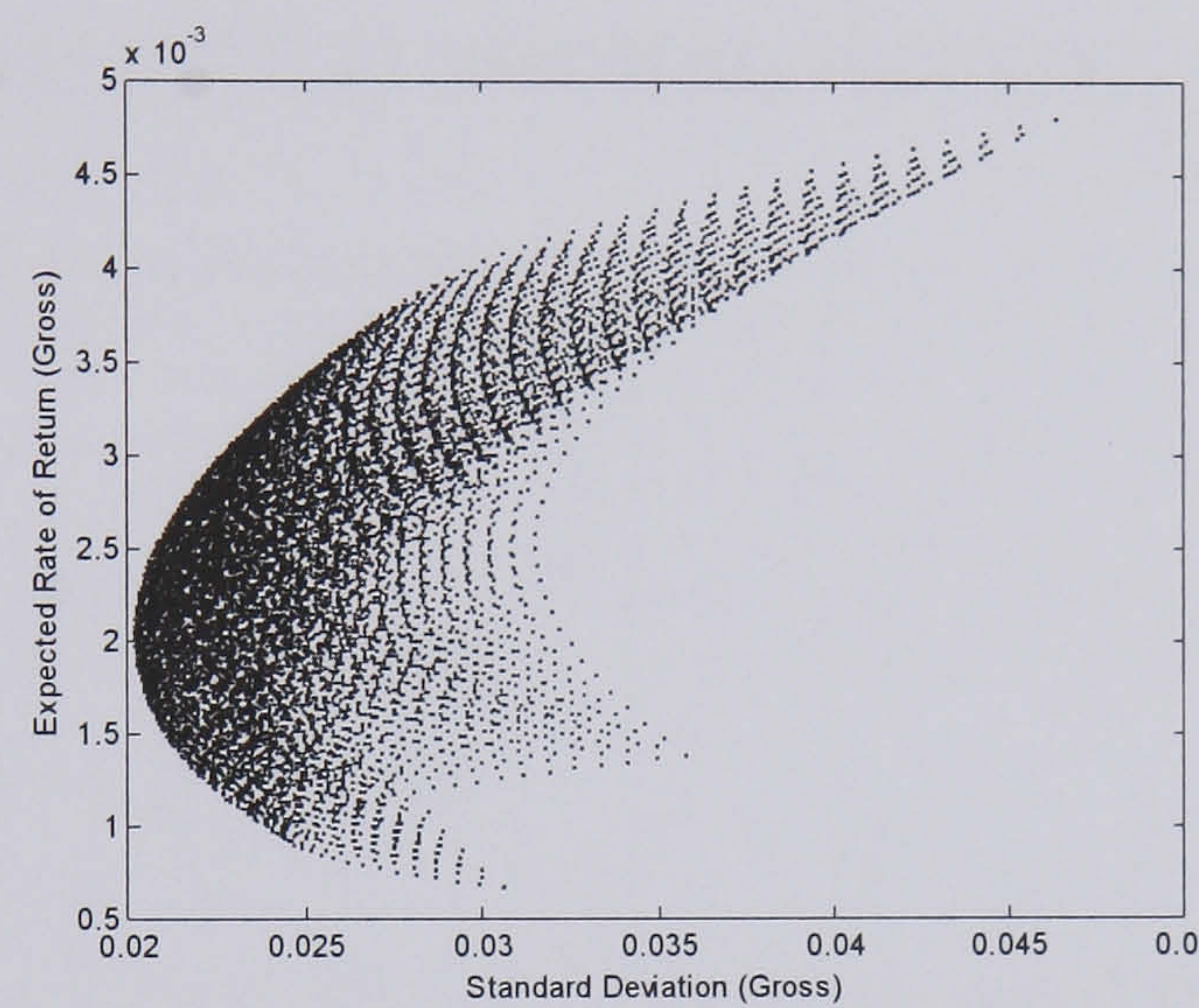


Figure 3.19: Feasible combinations of four assets (gross).

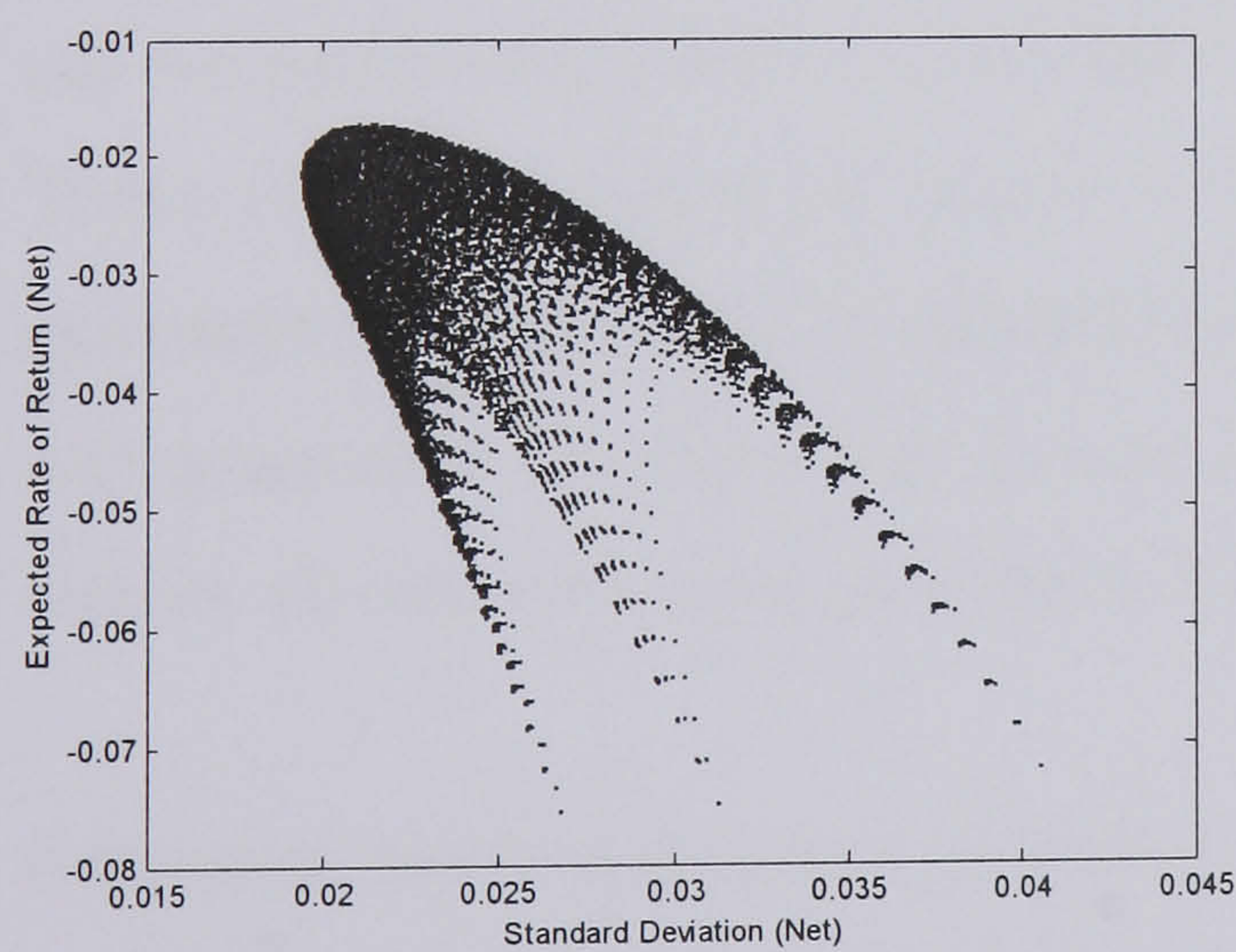


Figure 3.20: Feasible combinations of four assets with liquidity coefficients of 3% and 5%.

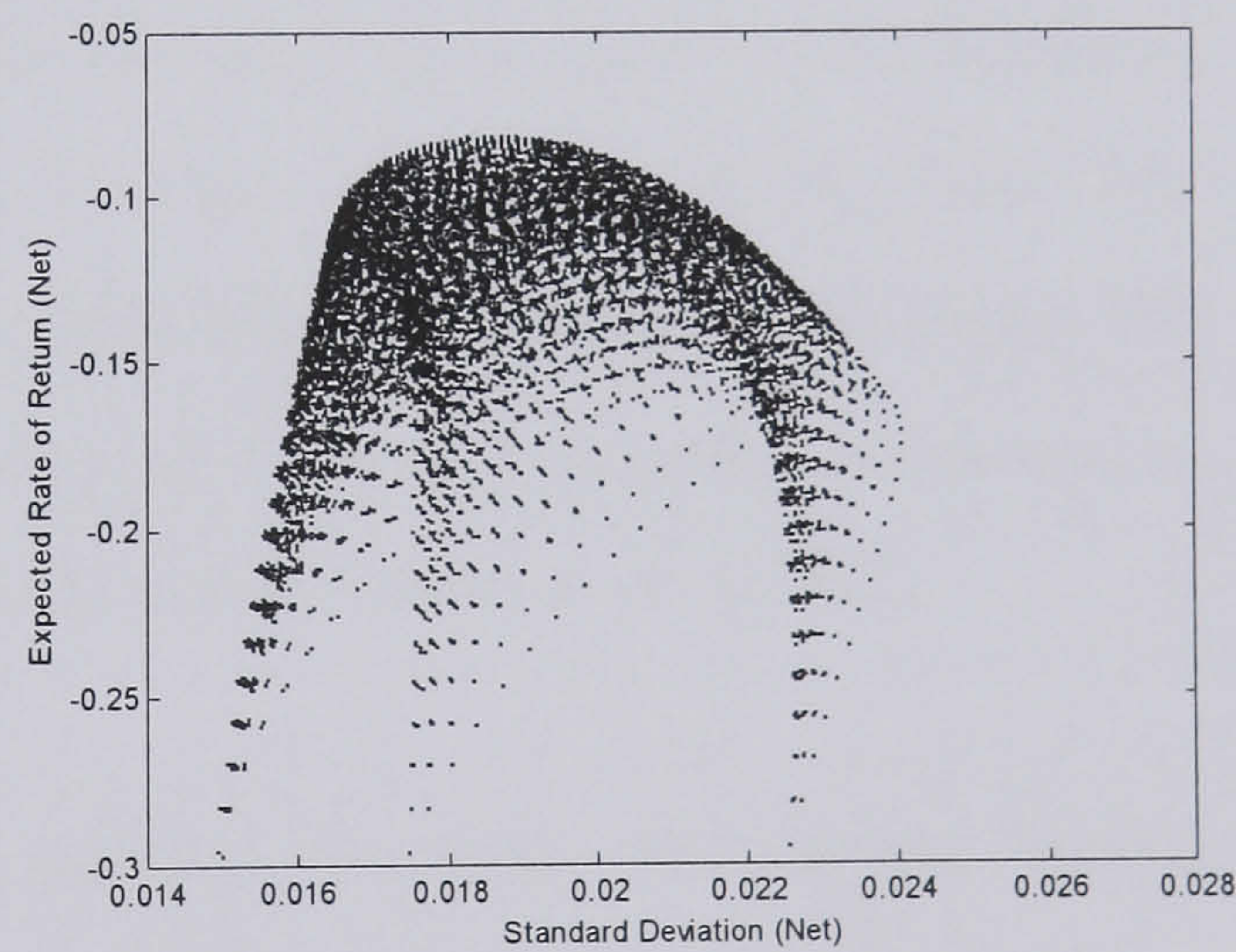


Figure 3.21: Feasible combinations of four assets with liquidity coefficients of 10% and 25%.

3.4. Fixed Transaction Costs

Fixed transaction costs better describe real world situations where a fixed charge for any nonzero trade is applied. Considering fixed costs also discourages trading small amounts of large number of assets. In the general case they imply that some sections of the transaction cost function are concave and lead to a hard combinatorial problem. In particular if we consider only the budget constraint as in [3.2]⁹, by introducing fixed costs, we need to modify it as follows:

$$\sum_i t_{0,i}(w_i) = 1 \quad [3.28]$$

$$t_{0,i}(w_i) = \begin{cases} 0, & w_i = 0 \\ F_i + w_i(1 + s_{0,i}) & w_i \neq 0 \end{cases} \quad [3.29]$$

Where [3.28] is the new budget constraint, $t_{0,i}$ represents the total cost of purchasing asset i , and F_i represents the fixed cost. Similarly for the constraint on the expected return. To compute the exact solution, an exhaustive search can be performed. Each w_i can either be set to 0, or assumed to be nonzero. These two alternatives for each of the n assets yield 2^n combinations. The exhaustive search can be improved by using branch and bound methods. We will propose a heuristic that can be used to find approximate solutions. A similar algorithm was independently developed by Lobo et al. (2002).

Assuming lower and upper bounds (l_i and u_i) for each variable are known (alternatively, $-\infty$ and $+\infty$ can be used) we obtain a convex envelope of $t_{0,i}$.

⁹ For simplicity [3.2] is used, which does not include variable spreads. The example above can easily be extended to include the case where the spread depends on the transaction size.

$$t_{0,i}^{CE}(w_i) = \begin{cases} w_i \left(\frac{F_i}{u_i} + 1 + s_{0,i}^+ \right), & w_i \geq 0 \\ w_i \left(-\frac{F_i}{-l_i} + 1 + s_{0,i}^- \right), & w_i < 0 \end{cases} \quad [3.30]$$

The convex envelope t^{CE} relaxes the budget constraint, therefore the optimal value of the resulting problem is a lower bound on the optimal value of the unmodified problem. Using the solution of this problem, w^0 , an iterative procedure can be designed such that at each step, k , the budget constraint is replaced with [3.31] and the resulting problem is solved:

$$t_{0,i}^{\{k\}}(w_i) = \begin{cases} w_i \left(\frac{F_i}{w_i^{\{k-1\}} + \delta} + 1 + s_{0,i}^+ \right), & w_i \geq 0 \\ w_i \left(-\frac{F_i}{-w_i^{\{k-1\}} + \delta} + 1 + s_{0,i}^- \right), & w_i < 0 \end{cases} \quad [3.31]$$

If the solution of two consecutive iterations, $w^{\{k\}}$ and $w^{\{k-1\}}$, are approximately equal, the algorithm stops and returns $w^* = w^{\{k\}}$, otherwise a new optimisation is performed.

An interpretation of the algorithm is that the fixed costs are amortised over the transaction amount of the previous iteration. At each iteration small transactions in the solution are driven to 0. As w_i^* becomes small, the modified $t^{\{k\}}$ has an increased derivative, pushing w_i^* to 0. The parameter δ defines a soft threshold for deciding whether w_i^* should be 0¹⁰. The solution w^* will be nearly feasible for the original problem.

For $w_i^* = 0$

$$t_{0,i}^{\{k\}}(0) = t_{0,i}(0) = 0 \quad [3.32]$$

And for $w_i^* \gg \delta$

$$t_{0,i}^{\{k\}}(w_i^*) = w_i^* \left(\frac{F_i}{w_i^* + \delta} + 1 + s_{0,i} \right) \approx F_i + w_i^* (1 + s_{0,i}) = t_{0,i}(w_i^*) \quad [3.33]$$

At each iteration the nonconvex transaction costs are replaced by convex ones, that agree with the nonconvex costs at the current iterate.

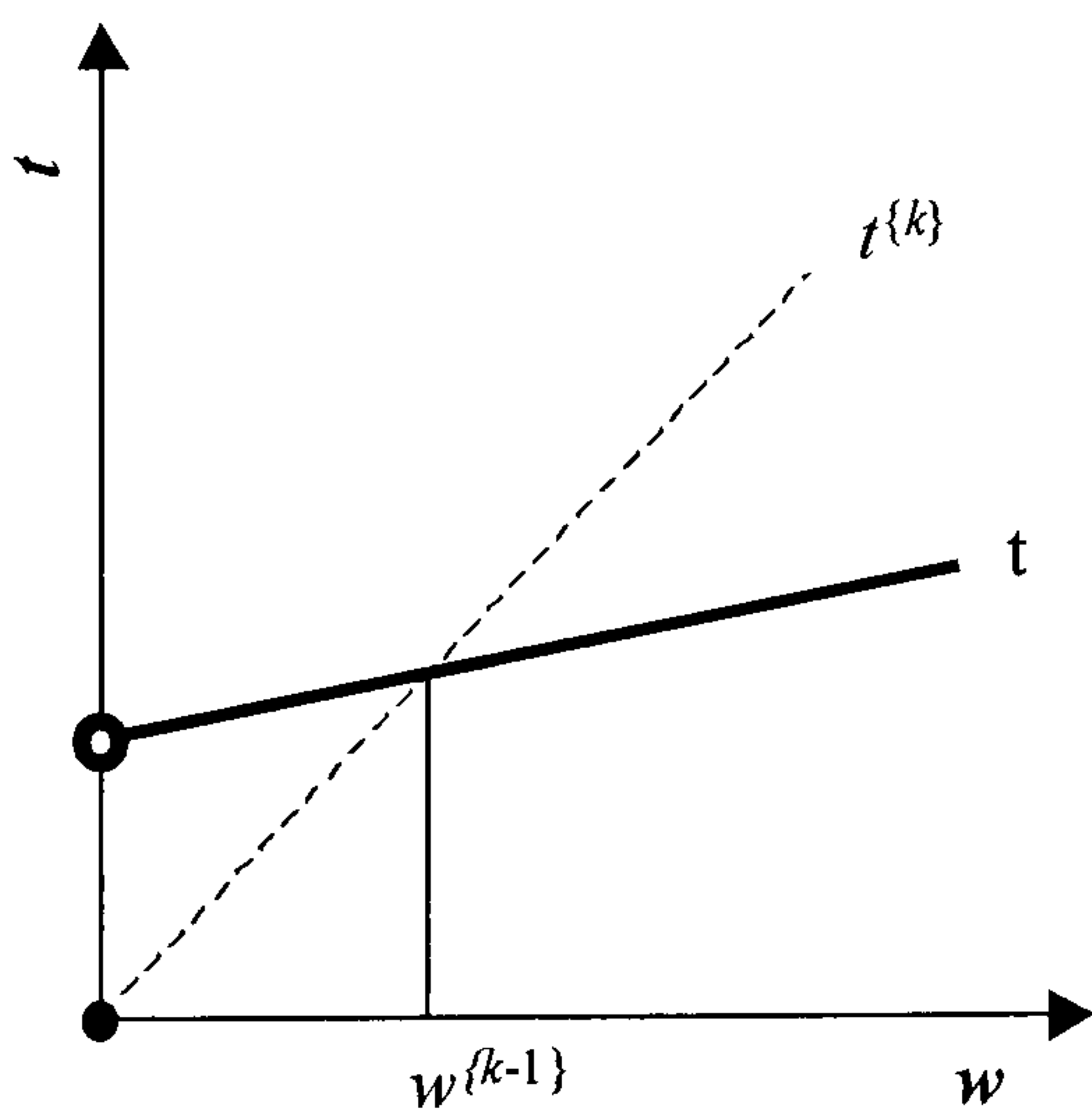


Figure 3.22: Approximation of the fixed cost function¹¹.

At each iteration, the problem to be solved is analogous to the one without fixed costs. Therefore, in what follows we will assume that fixed costs are not included.

¹⁰ When a solution w^* is in the order of δ (e.g. when $\delta < w^*$, but not $\delta \ll w^*$), feasibility problems may arise, with a non-negligible violation of the budget constraint. In practice terms in the order of δ will seldom appear in the solution.

¹¹ At each step, k , the nonconvex transaction cost function t is replaced by the convex transaction cost function $t^{\{k\}}$. If two successive iterates are the same, then the iterates are almost feasible solutions of the original problem.

3.5. Portfolio Rebalancing Problem

If we assume that the investor has some initial positions, represented by the variable x_i ¹², the problem becomes:

$$\underset{g}{\text{Min}} f(w + x) = \underset{g}{\text{Min}} \text{VAR} \left[\sum_i (w_i + x_i) n_i (w_i + x_i) \right] \quad [3.34]$$

s.t.

$$\sum_i g_i = 0 \quad [3.35]$$

$$w_i = \frac{g_i}{m_i(g_i)} \quad [3.36]$$

$$\sum_i (w_i + x_i) E[n_i(w_i + x_i)] = R_{PN} \quad [3.37]$$

$$x_i + w_i \geq 0 \quad [3.38]$$

Note that in this case g , and therefore w , can also be negative, but overall short positions are not allowed. The model assumes that the portfolio will be liquidated at the end of the period.

In practice, it is unlikely that all the positions will be closed at horizon and the formulation proposed might be too conservative. On the other hand, by completely disregarding the impact of liquidity, implicitly assuming that no asset will be liquidated in the next period, investors might underestimate the risks involved. An ad hoc solution might be to multiply the expected future spread by an amortisation factor between 0 and 1. The historical or expected portfolio turnover might be used to calculate this parameter. A way to think of the amortisation process is to pretend that in order to cover the liquidity costs

¹² One tricky aspect of the portfolio rebalancing problem is that in practice it is a multi-period problem. The mean-variance model assumes a one-period world and it is inadequate when solving a multi-period optimisation. Nevertheless many market practitioners use a rolling Markowitz model, rebalancing their positions at given time intervals, by solving each time a mean-variance optimisation.

paid when the position is closed, we borrow funds using our investment assets as collateral for the loan. We can infinitely defer the payment of the liquidation costs as long as the collateral is sufficient. As such the cost of the loan interest becomes the true liquidity cost. If trades will improve the expected return or reduce risk sufficiently to justify this interest cost, the transactions are a net improvement on the portfolio. In this case the future spread s_1 in the objective function and in the expected return constraint is replaced by $\alpha s_1(w)$, where α is the amortisation parameter.

4. Network Flows and Sequential Convex Programming

In this chapter we present an algorithm for the problem [3.15]-[3.20]. Considering that the constraints are separable it is possible to approximate the constraints using piecewise linear functions and express a feasible solution as a network flow problem. Unfortunately the objective function is neither separable nor convex. Given the fundamental network structure of the problem we therefore propose to use a sequence of approximations of the objective function in order to find a local minimum. We use a technique called sequential convex programming (a generalisation of the sequential quadratic programming algorithm), which yields a sequence of convex and separable approximations. Each subproblem can be modelled as a generalised network flow and extremely efficient algorithms can be used to solve it. In section 4.1, we illustrate the sequential convex programming approach. In section 4.2, we show how the resulting subproblem is linearised. In section 4.3, the network formulation of the problem is presented. In section 4.4, we describe the primal simplex algorithm used to solve each subproblem. In section 4.5, we show how parallel arcs can be handled using the primal simplex algorithm. In section 4.6, we illustrate the computational results.

4.1. Sequential Convex Programming

To reduce the computational cost of complex non-linear problems, the general procedure is to generate a sequence of convex, explicit subproblems and solve them in an iterative fashion, that is, a Sequential Convex Programming (SCP) approach is used (Fleury, 1993). Various approximation schemes have been developed for this purpose.

The approximation schemes of interest to us are local function approximations (Barthelemy and Haftka, 1993), which generate an approximated formulation of the problem in the vicinity of the current point. One of the earliest such schemes is Sequential Linear Programming (SLP). In SLP, a linear approximation of the function is formulated using the first order derivative term of the Taylor series expansion. Let us consider the k^{th} iteration of the optimisation process, with starting parameters approximation $w^{\{k\}}$.

$$\tilde{f}^{\{k\}}(w) = f(w^{\{k\}}) + \sum_i (w_i - w_i^{\{k\}}) \left. \frac{\partial f}{\partial w_i} \right|_{w^{\{k\}}} \quad [4.1]$$

The Convex Linearisation (CONLIN) method (Fleury and Braibant, 1986) uses reciprocals of the variables to formulate the approximation. This method linearises each function using a properly selected mix of direct and reciprocal variables. The selection of the variables is made based on the signs of the first partial derivatives, that is, direct variables for positive first derivatives and reciprocal for negative first derivatives. It is of the form

$$\tilde{f}^{\{k\}}(w) = r^{\{k\}} + \sum_+ w_i \left. \frac{\partial f}{\partial w_i} \right|_{w^{\{k\}}} - \sum_- w_i^{\{k\}2} \frac{1}{w_i} \left. \frac{\partial f}{\partial w_i} \right|_{w^{\{k\}}} \quad [4.2]$$

$$\frac{\partial \tilde{f}^{\{k\}}}{\partial w_i} = \begin{cases} \left. \frac{\partial f}{\partial w_i} \right|_{w^{\{k\}}} \geq 0, & \left. \frac{\partial f}{\partial w_i} \right|_{w^{\{k\}}} \geq 0 \\ \frac{w_i^{\{k\}2}}{w_i^2} \left. \frac{\partial f}{\partial w_i} \right|_{w^{\{k\}}} \leq 0, & \left. \frac{\partial f}{\partial w_i} \right|_{w^{\{k\}}} < 0 \end{cases} \quad [4.3]$$

$$\frac{\partial^2 \tilde{f}^{\{k\}}}{\partial w_i^2} = \begin{cases} 0, & \left. \frac{\partial f}{\partial w_i} \right|_{w^{\{k\}}} \geq 0 \\ -\frac{2w_i^{\{k\}2}}{w_i^3} \left. \frac{\partial f}{\partial w_i} \right|_{w^{\{k\}}}, & \left. \frac{\partial f}{\partial w_i} \right|_{w^{\{k\}}} < 0 \end{cases} \quad [4.4]$$

Where $r^{\{k\}}$ is such that $f(w^{\{k\}}) = \tilde{f}^{\{k\}}(w^{\{k\}})$. The symbol $\sum_+ (\sum_-)$ means summation over the positive (negative) first derivatives. This method yields convex (for $w \geq 0$) and separable approximations. CONLIN employs more conservative¹ approximations and has shown good convergence properties in dealing with structural optimisation problems. In some cases, this convex approximation is either too conservative or not sufficiently conservative in which case oscillations occur.

Svanberg (1987) proposed a modification of CONLIN called the Method of Moving Asymptotes (MMA). In this method, the linearisation variables can be used to adjust the degree of convexity and conservativeness depending on the problem. The variables are of the form $1/(w_i - l_i)$ and $1/(u_i - w_i)$ where u_i and l_i are user selected variables called the moving asymptotes, such that $l_i \leq w_i \leq u_i$. The approximation is of the form:

$$\tilde{f}^{\{k\}}(w) = r^{\{k\}} + \sum_+ \frac{(u_i - w_i^{\{k\}})^2}{u_i - w_i} \frac{\partial f}{\partial w_i} \bigg|_{w^{\{k\}}} - \sum_- \frac{(w_i^{\{k\}} - l_i)^2}{w_i - l_i} \frac{\partial f}{\partial w_i} \bigg|_{w^{\{k\}}} \quad [4.5]$$

$$\frac{\partial \tilde{f}^{\{k\}}}{\partial w_i} = \begin{cases} \frac{(u_i - w_i^{\{k\}})^2}{(u_i - w_i)^2} \frac{\partial f}{\partial w_i} \bigg|_{w^{\{k\}}} \geq 0, & \frac{\partial f}{\partial w_i} \bigg|_{w^{\{k\}}} \geq 0 \\ \frac{(w_i^{\{k\}} - l_i)^2}{(w_i - l_i)^2} \frac{\partial f}{\partial w_i} \bigg|_{w^{\{k\}}} \leq 0, & \frac{\partial f}{\partial w_i} \bigg|_{w^{\{k\}}} < 0 \end{cases} \quad [4.6]$$

$$\frac{\partial^2 \tilde{f}^{\{k\}}}{\partial w_i^2} = \begin{cases} \frac{2(u_i - w_i^{\{k\}})^2}{(u_i - w_i)^3} \frac{\partial f}{\partial w_i} \bigg|_{w^{\{k\}}} \geq 0, & \frac{\partial f}{\partial w_i} \bigg|_{w^{\{k\}}} \geq 0 \\ -\frac{2(w_i^{\{k\}} - l_i)^2}{(w_i - l_i)^3} \frac{\partial f}{\partial w_i} \bigg|_{w^{\{k\}}} \geq 0, & \frac{\partial f}{\partial w_i} \bigg|_{w^{\{k\}}} < 0 \end{cases} \quad [4.7]$$

¹ The approximation of the feasible domain is conservative if it tends to underestimate the size of the domain. This is attractive because it approaches the optimum through a sequence of feasible problems. SLP approximations are not

The moving asymptotes u_i and l_i can be used to control the optimisation process. If the process oscillates, then by moving the asymptotes closer to the current iteration point it can be stabilised, and if it converges slowly the asymptotes are moved away. MMA offers a great deal of flexibility in matching the curvature of the approximated function through the choice of l_i and u_i . However, empirical techniques have to be used to determine their values after each iteration.

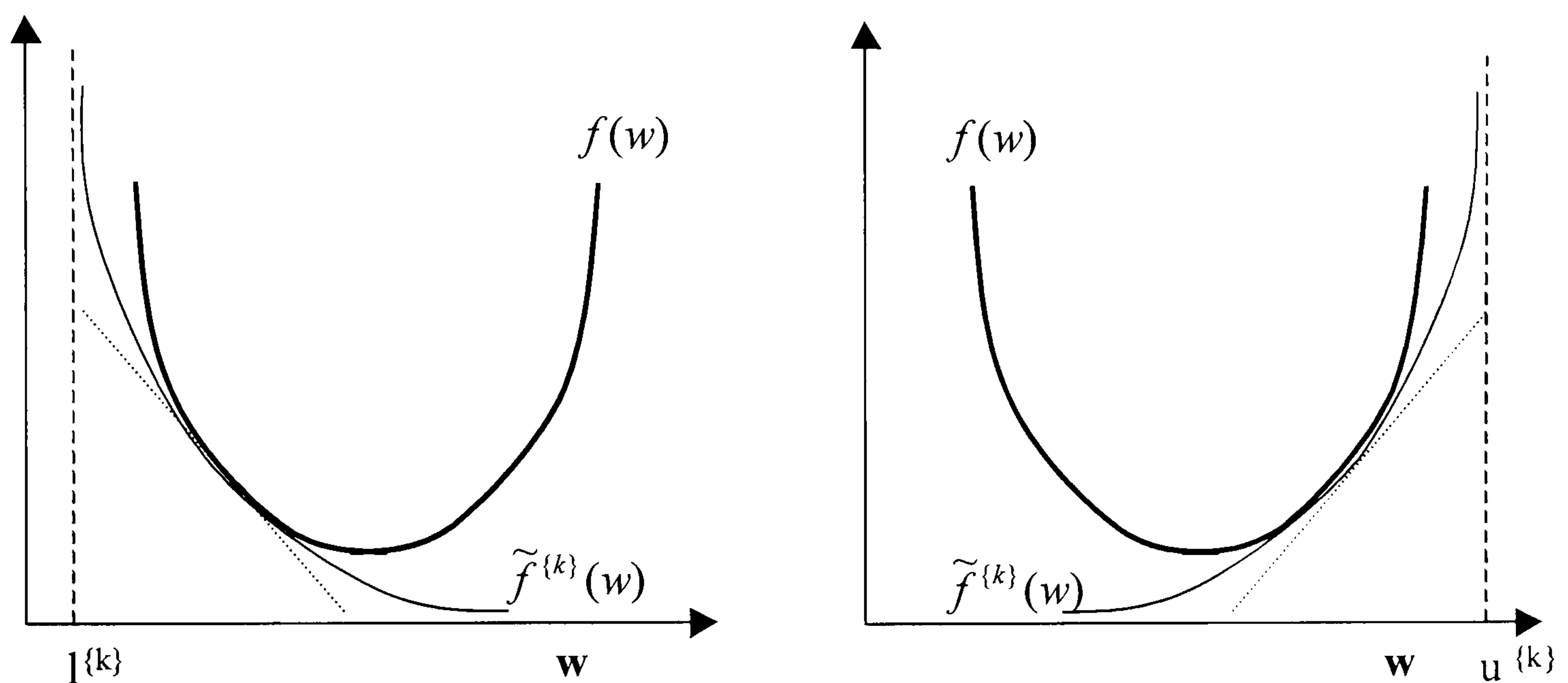


Figure 4.1: Method of the moving asymptotes.

Let us point out a major property of both CONLIN and MMA methods. The objective function is approximated by a monotonous increasing function if the function derivative is positive, or by a monotonous decreasing function if the derivative is negative. Hence, if constraints are not approximated (e.g. because of their simple explicit form), this causes a non-convergence of both methods if the minimum point is located strictly inside the feasible domain (no active constraints).

conservative.

Nevertheless, Svanberg (1995) proposed a very simple modification of MMA to make it globally convergent. This method, called "Globally Convergent Method of Moving Asymptotes" (GCMMA), still creates convex approximations of functions in the problem (objective function and constraints if necessary), but they present a "U" shape with a minimum between two vertical asymptotes (the moving asymptotes defined in MMA) for each optimisation variable. A new convex approximation parameter appears, called ρ . It is modified from one iteration to another similarly to moving asymptotes updating.

Let us suppose that $u^{\{k\}}$ and $l^{\{k\}}$, the moving asymptotes at the k^{th} iteration, and $\rho^{\{k\}}$, are known (see the updating method described later). After computing the derivative of $f(w)$ with respect to each optimisation variable, the subproblem objective function is given by:

$$\tilde{f}^{\{k\}}(w) = r^{\{k\}} + \sum_i \left(\frac{p_i^{\{k\}}}{u_i^{\{k\}} - w_i} + \frac{q_i^{\{k\}}}{w_i - l_i^{\{k\}}} \right) \quad [4.8]$$

$$p_i^{\{k\}} = (u_i^{\{k\}} - w_i^{\{k\}})^2 \left(\left(\frac{\partial f}{\partial w_i} \Big|_{w_i^{\{k\}}} \right)_+ + \frac{\rho^{\{k\}}}{2} (u_i^{\{k\}} - l_i^{\{k\}}) \right) \quad [4.9]$$

$$q_i^{\{k\}} = (w_i^{\{k\}} - l_i^{\{k\}})^2 \left(\left(\frac{\partial f}{\partial w_i} \Big|_{w_i^{\{k\}}} \right)_- + \frac{\rho^{\{k\}}}{2} (u_i^{\{k\}} - l_i^{\{k\}}) \right) \quad [4.10]$$

$$r^{\{k\}} = f(w^{\{k\}}) - \sum_i \left(\frac{p_i^{\{k\}}}{u_i^{\{k\}} - w_i^{\{k\}}} + \frac{q_i^{\{k\}}}{w_i^{\{k\}} - l_i^{\{k\}}} \right) \quad [4.11]$$

Where the symbols $()_+$ and $()_-$ respectively mean:

$$(y)_+ = \begin{cases} y, & y > 0 \\ 0, & y \leq 0 \end{cases} \quad (y)_- = \begin{cases} 0, & y > 0 \\ y, & y \leq 0 \end{cases}$$

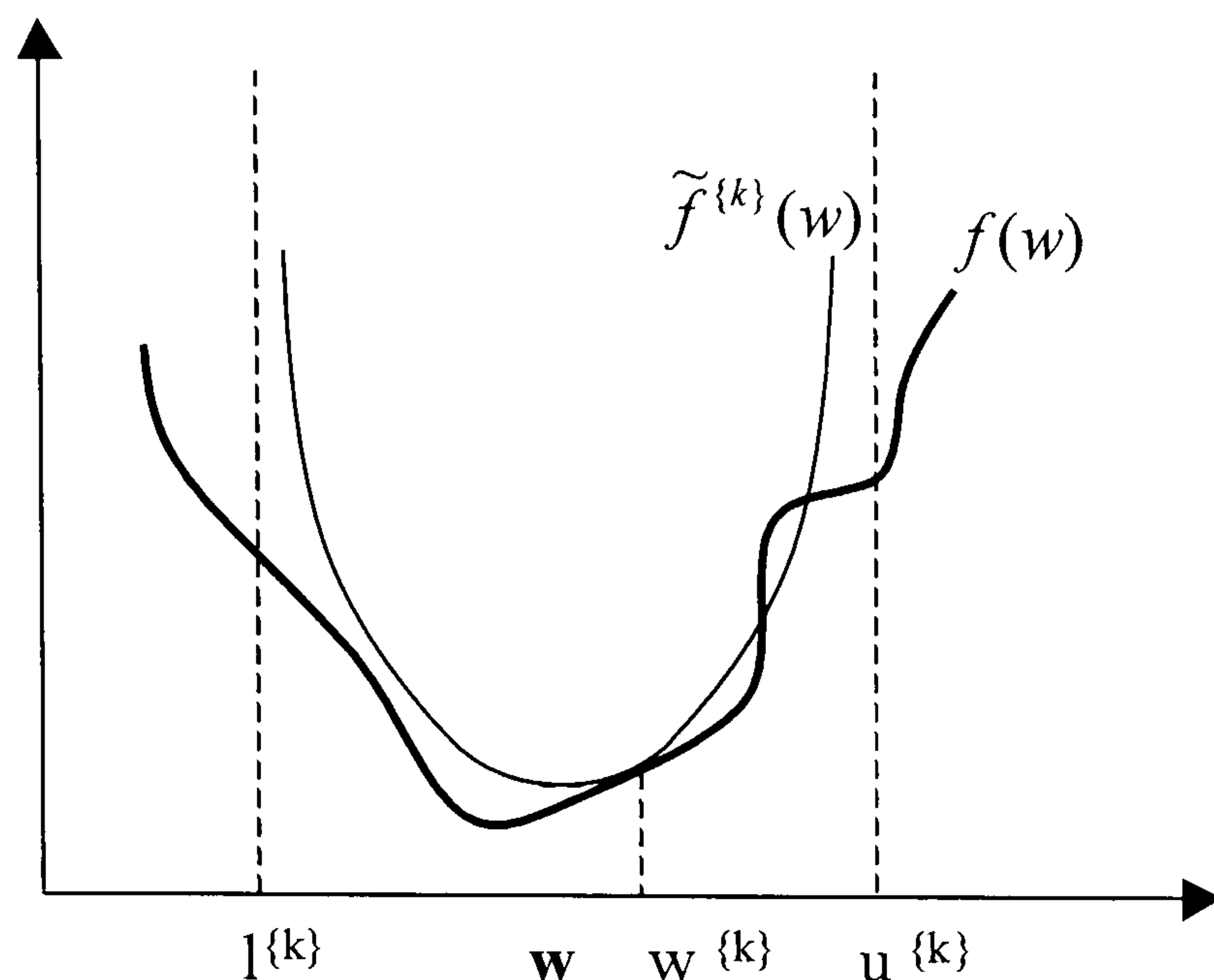


Figure 4.2: Globally Convergent Method of Moving Asymptotes².

Moving asymptotes updating is similar to MMA updating (see below). Regarding ρ , Svanberg proposed the following updating law that has already proven its efficiency in structural optimisation. From equation [4.8], the approximated function can be expressed as:

$$\tilde{f}^{\{k\}}(w) = h^{\{k\}}(w) + \rho^{\{k\}}(w)d^{\{k\}}(w) \quad [4.12]$$

Where $h^{\{k\}}(w)$ and $d^{\{k\}}(w)$ do not depend on $\rho^{\{k\}}$. Let us define

$$\gamma^{\{k\}} = \frac{f(w^{k+1}) - \tilde{f}^{\{k\}}(w^{k+1})}{d^{\{k\}}(w^{k+1})} \quad [4.13]$$

The ρ updating rule proposed by Svanberg is:

² The figure shows the objective function (bold line) and its convex approximation (thin line).

$$\rho^{\{k+1\}} = \begin{cases} \rho^{\{k\}} \frac{6\rho^{\{k\}} + 4\gamma^{\{k\}}}{6\rho^{\{k\}} + \gamma^{\{k\}}}, & \gamma^{\{k\}} \geq 0 \\ \rho^{\{k\}} \frac{2\rho^{\{k\}} - \gamma^{\{k\}}}{2\rho^{\{k\}} - 2\gamma^{\{k\}}}, & \gamma^{\{k\}} < 0 \end{cases} \quad [4.14]$$

In order to ensure the quality of convex approximations with MMA and GCMMA, it is crucial to use an efficient moving asymptotes updating law. Following Zhang and Fleury (2002), the updating rule we suggest adjusts automatically moving asymptotes in order to ensure that the k^{th} convex approximation will fit the exact function at the preceding iteration point $w^{\{k-1\}}$. In this case:

$$l_i^{\{k\}}(w) = w_i^{\{k\}} - \delta(w_i^{\{k-1\}} - l_i^{\{k-1\}}) \quad [4.15]$$

$$u_i^{\{k\}}(w) = w_i^{\{k\}} + \delta(u_i^{\{k-1\}} - w_i^{\{k-1\}}) \quad [4.16]$$

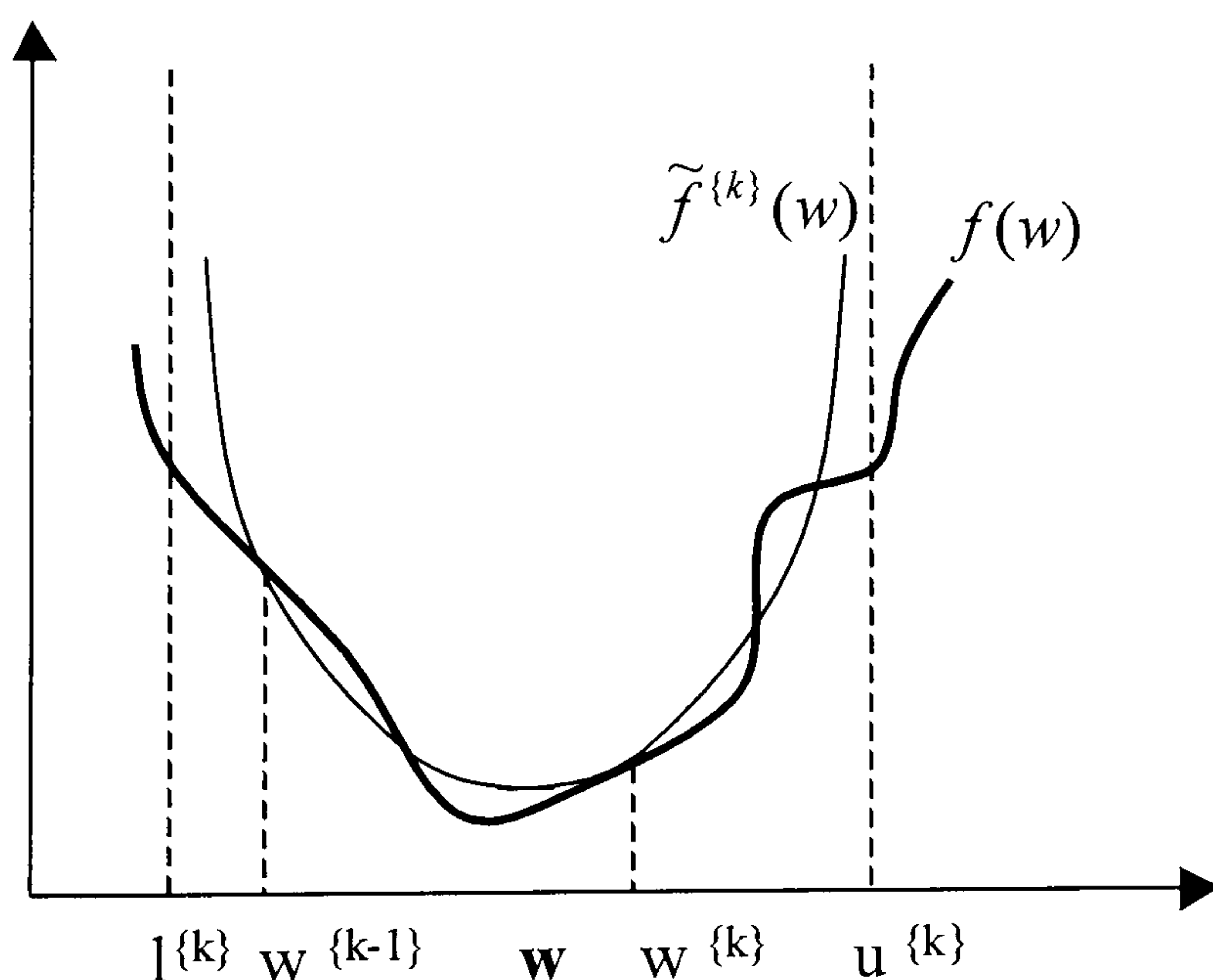


Figure 4.3: Moving asymptotes updating.

Where the parameter δ is computed in order to ensure that $f(w^{\{k-1\}}) = \tilde{f}^{\{k\}}(w^{\{k-1\}}, \delta)$. This non-linear equation is solved with a Newton-Raphson iterative scheme. When this parameter is computed, new moving asymptotes are evaluated.

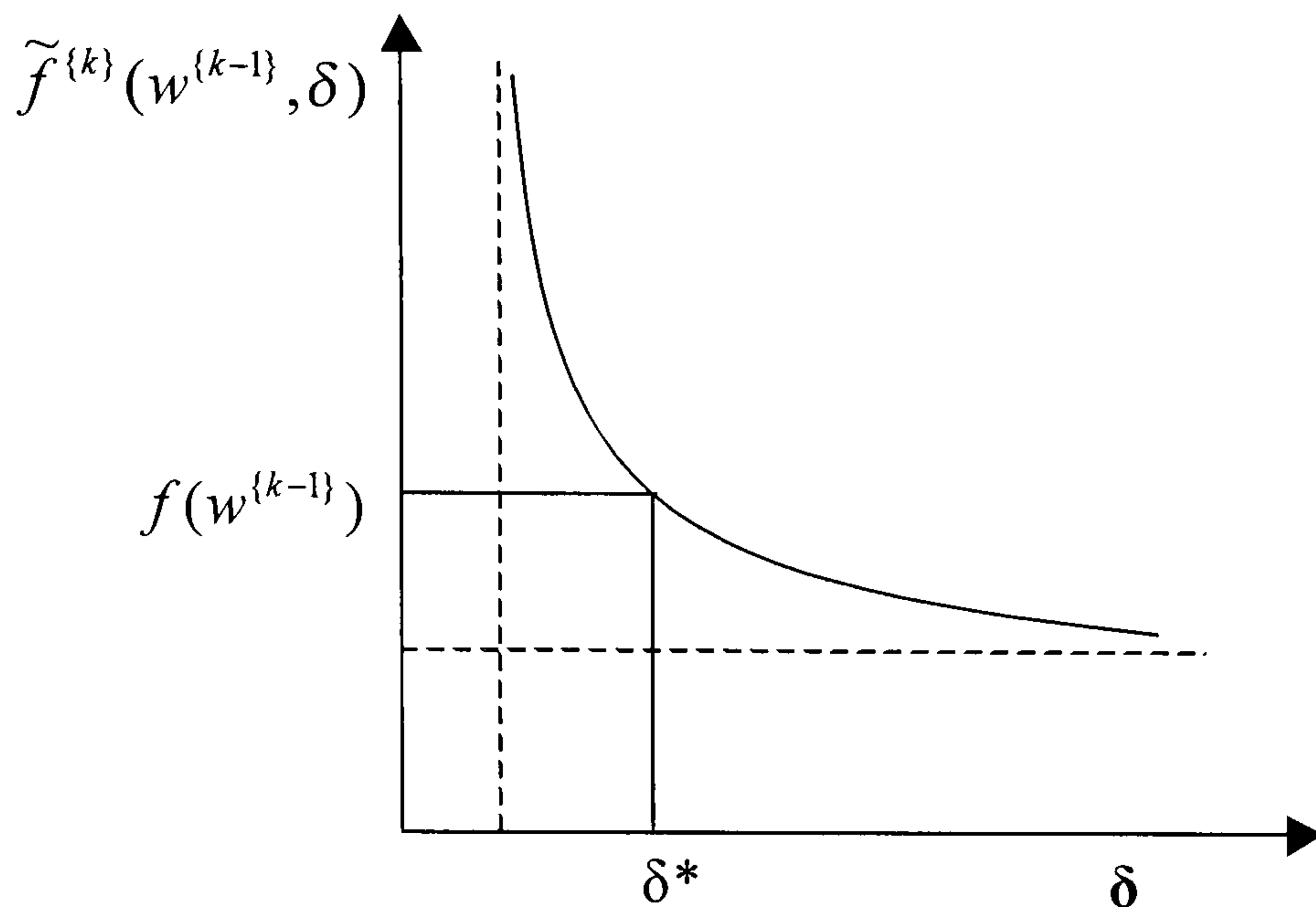


Figure 4.4: Convex approximation evolution with parameter δ .

4.2. Piecewise-Linear Approximation of the Subproblem

Depending on the spread function used convex approximations of both the objective functions and the constraints might be required. The primal simplex algorithm requires constant cost coefficients and arc gains. Non-linear functions can in this case be expressed as parallel arcs with appropriate flow bounds. In order to simplify the notation, we will assume that all the PL approximations use the same set of breakpoints.

The following notation is used in the approximation of a given function,

$f_i(w_i)$:

$$\hat{f}_i(w_i) = \sum_{p=1}^q \hat{f}_i^{\{p\}} w_i^{\{p\}} \quad [4.17]$$

With:

$$w_i^{\{p\}} = \begin{cases} 0 & \text{if } w_i < b_i^{\{p-1\}} \\ w_i - b_i^{\{p-1\}} & \text{if } b_i^{\{p-1\}} \leq w_i \leq b_i^{\{p\}} \\ b_i^{\{p\}} - b_i^{\{p-1\}} & \text{if } w_i > b_i^{\{p\}} \end{cases} . \quad [4.18]$$

$$0 = b_i^{\{0\}} < b_i^{\{1\}} \dots < b_i^{\{p-1\}} < b_i^{\{p\}} \dots < b_i^{\{q\}} \quad [4.19]$$

Where $b_i^{\{p\}}$ denotes the p^{th} breakpoint of the PL function $\hat{f}_i(w_i)$. By definition, $w_i = \sum_{p=1}^q w_i^{\{p\}}$. Furthermore, $\hat{f}_i^{\{p\}}$ is a scalar representing the slope of the p^{th} segment. If $f_i(w_i)$ is convex, then the sequence $\{\hat{f}_i^{\{p\}}\}_p$ will be monotonically increasing; this implies that the solution of a network flow representation is contiguous. As can be noted from [4.17], we have assumed that the PL approximation starts at 0. This is not a restrictive assumption and can easily be relaxed because we can always satisfy it by defining $\hat{f}_i^{\{p\}}(w_i^{\{p\}}) = \hat{f}_i^{\{p\}}(w_i^{\{p\}}) - \hat{f}_i^{\{p\}}(0)$.

Several algorithms are available to calculate the specific set of breakpoints $b_i^{\{p\}}$ and the set of corresponding slopes $\hat{f}_i(w_i)$. An important characteristic of such approximations is that they should be conservative. This means that approximations involving the objective function should be greater than or equal to the original function and approximations of the constraints should produce a feasible set which is not larger than the original feasible set. A common approach is to select a discrete set of points in the domain and use linear interpolations of the images of adjacent pairs of the selected points.

This procedure in our case produces conservative approximations. The objective function is in fact a sum of convex functions and therefore the approximations obtained will in no case underestimate the risk of the portfolio. The constraints [3.17]-[3.18] and [3.19] are non-linear functions, where their

specific functional form depends on the choice of the spread function $s(w)$. Since, for fixed liquidity coefficients, the spread is a monotonically increasing function of the weights, $n(w)$ is decreasing in w and $m(g)$ is increasing in g . As we will illustrate in the following sections the budget constraint implies a flow from the cash node to the various assets, therefore the multipliers are reciprocals of $m(g)$. In both cases the approximated constraints produce a conservative feasible set.

The simplest method to select an appropriate set of breakpoints is uniform decomposition, where the domain is divided into equal intervals. Since it is difficult to choose an adequate interval size, uniform decomposition typically produce approximations that oversample regions of low curvature and undersample regions of high curvature. A better alternative is adaptive decomposition, in which the sampling rate varies across the domain according to the complexity of the surface as measured by the variation of its curvature. However finding the best approximation with the exact minimum number of breakpoints required for a given accuracy is an NP-hard problem and we must instead rely on heuristics. In a typical adaptive decomposition algorithm, the domain is recursively subdivided until the approximation is within a tolerance region. We propose a simple implementation of the above method.

1. Initially the domain is divided into 2 equal segments.
2. The first order derivatives at each of the three points are calculated.
3. If the difference of the derivatives of two consecutive breakpoints is greater than a given threshold the interval is subdivided, and the procedure is repeated for each of the resulting two intervals.

Given that all the functions are either concave or convex the procedure above produces accurate approximations. The algorithm can be further improved by

requiring a finer grid in the region of the current iterate. In our computational results we will however use only uniform decomposition.

Once the constraints are linearised, solving either [3.16]-[3.17] or [3.19] is equivalent to solving a system of n^q linear constraints, although, exploiting the contiguity of the resulting flows and the fact that each liquidity function does not depend on the transaction size of the other assets, it is possible to reduce the dimension of the system considerably.

If we now consider the original formulation of the subproblem, and use the PL approximations of the non-linear functions as illustrated in [4.17]-[4.19], the linearised problem becomes³:

$$\min_w \sum_{i=1}^n \sum_{p=1}^q w_i^{\{p\}} \hat{f}_i^{\{p\}} \quad [4.20]$$

s.t.

$$\sum_{i=1}^n \sum_{p=1}^q w_i^{\{p\}} \hat{m}_i^{\{p\}} = 1 \quad [4.21]$$

$$\sum_{i=1}^n \sum_{p=1}^q w_i^{\{p\}} \hat{n}_i^{\{p\}} = R_{PN} \quad [4.22]$$

$$w_i^{\{p\}} = \begin{cases} 0 & \text{if } w_i < b_i^{\{p-1\}} \\ w_i - b_i^{\{p-1\}} & \text{if } b_i^{\{p-1\}} \leq w_i \leq b_i^{\{p\}} \\ b_i^{\{p\}} - b_i^{\{p-1\}} & \text{if } w_i > b_i^{\{p\}} \end{cases} \quad [4.23]$$

In particular, since $\hat{m}_i^{\{p\}}$ depends on the gross weight, g_i , only through the breakpoints, we can simplify the notation by considering only the net weights, w . Interpreted as a network flow model, the budget constraint [4.21] is equivalent to imposing a restriction on the inverted flow on the directed arcs

³ In order to simplify the notation we assumed that all PL approximations use the same set of breakpoints. This does not need to be the case. If a different set of breakpoints is used for each non-linear function the constraint [4.23] needs to be adjusted accordingly.

emanating from the cash node. Since the breakpoints are expressed with respect to the inverted flow they need to be adjusted accordingly⁴.

4.3. Network Flow Formulation of the Subproblem

The preceding discussion implies that we can find approximate solutions to the generalised convex cost flow problem by solving an associated generalised minimum cost flow problem. In this section we will show how the subproblems can be represented as generalised network flow models. This representation allows us to use efficient algorithms that have been developed to exploit the particular structure of the netform.

Specialised software for this formulation is often more than an order of magnitude faster than using general purpose code. The terminology "generalised" networks results because of the presence of arc multipliers. It is well known (Dantzig 1963) that a generalised network is a special case of a linear programming problem for which the basis B can be written in the form:

$$B = \begin{bmatrix} B_1 & & & \\ & \ddots & & \\ & & B_i & \\ & & & \ddots \\ & & & & B_k \end{bmatrix}$$

Where each B_i is either upper triangular or nearly upper triangular with only one element below the diagonal. Using techniques from graph theory we associate rows of B_i with nodes and columns with arcs. Then every triangular

⁴ A flow bound with respect to the inverted flow, w , is equivalent to a flow bound on the directed flow, g , divided by the arc multiplier on the directed graph.

submatrix of B corresponds to a spanning tree with a root node and every submatrix B_i that is nearly triangular corresponds to a tree with a loop. Spanning trees with root nodes or with loops are defined as augmented trees. Thus the basis is a forest of augmented trees. Generalised networks may be viewed as an extension of the class of pure networks where all the multipliers are identically equal to $+1$. From a numerical point of view, however, the distinction between pure and generalised networks is an important one. The basis of a pure network is a spanning tree, not a forest (see the following section), and this makes the relevant computations very efficient. It appears from empirical tests (Brown and McBride, 1984) that generalised network codes are only marginally slower than their pure network counterparts. These codes in turn can achieve a hundredfold improvement in efficiency over general purpose linear programming software. The highly sparse nature of the problem allows for the solution of very large networks.

A major shortcoming of the network flow formulation of the problem is that additional constraints can not generally be expressed within the network formulation. In this case Lagrangian relaxation might be used. Alternatively, Glover and Klingman (1981) proposed an efficient algorithm that can cope with side constraints.

The network representation of the subproblem is illustrated in Figure 4.5. The arcs in bold represent the objective function while the thin arcs represent the constraints. All the nodes in the network have zero demand and are therefore transshipment nodes except the first node, which is a supply node with supply of 1 and the last node, which is a demand node with demand of R_{PN} . Each couple of curved arcs represents the set of parallel arcs corresponding to the PL approximation of a non-linear function. Only two assets, the first and the last, and only the first and the last arc of each PL approximation are graphed. Note that each constraint arc (thin arrows) has zero cost and a multiplier not

necessarily equal to 1 and each cost arc (bold arrows) has a unit multiplier and a positive cost. Finally, the notation of the upper flow bound, $b_i^{\{p\}}$, should be interpreted as specific to the particular PL approximation.

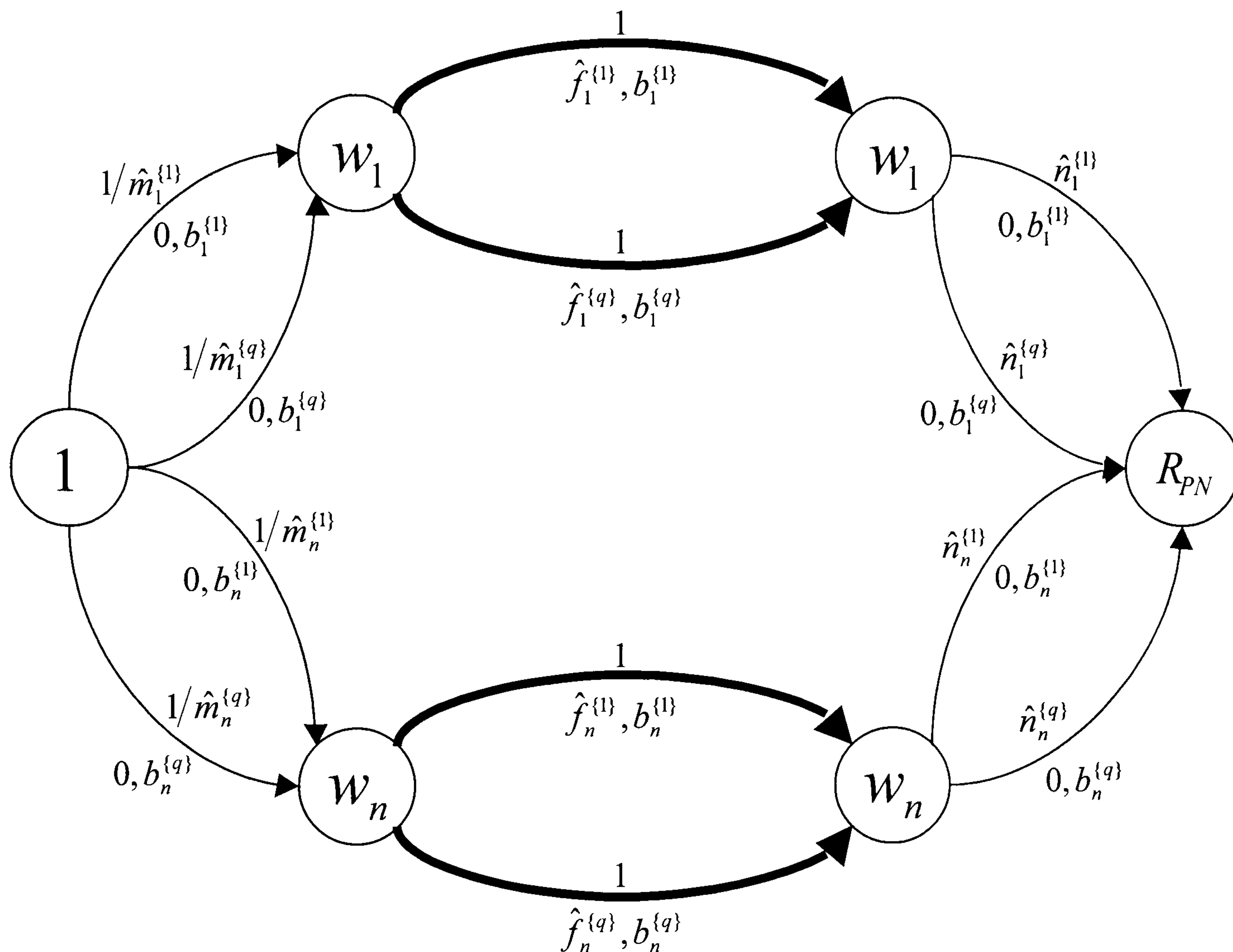


Figure 4.5: Network representation of the generalised minimum cost flow problem when the non-linear functions are approximated by PL functions⁵.

The network can be divided into three subgraphs.

⁵ The expression above each arc represents its multiplier, the expression below the arc includes information about the arc cost and its upper bound.

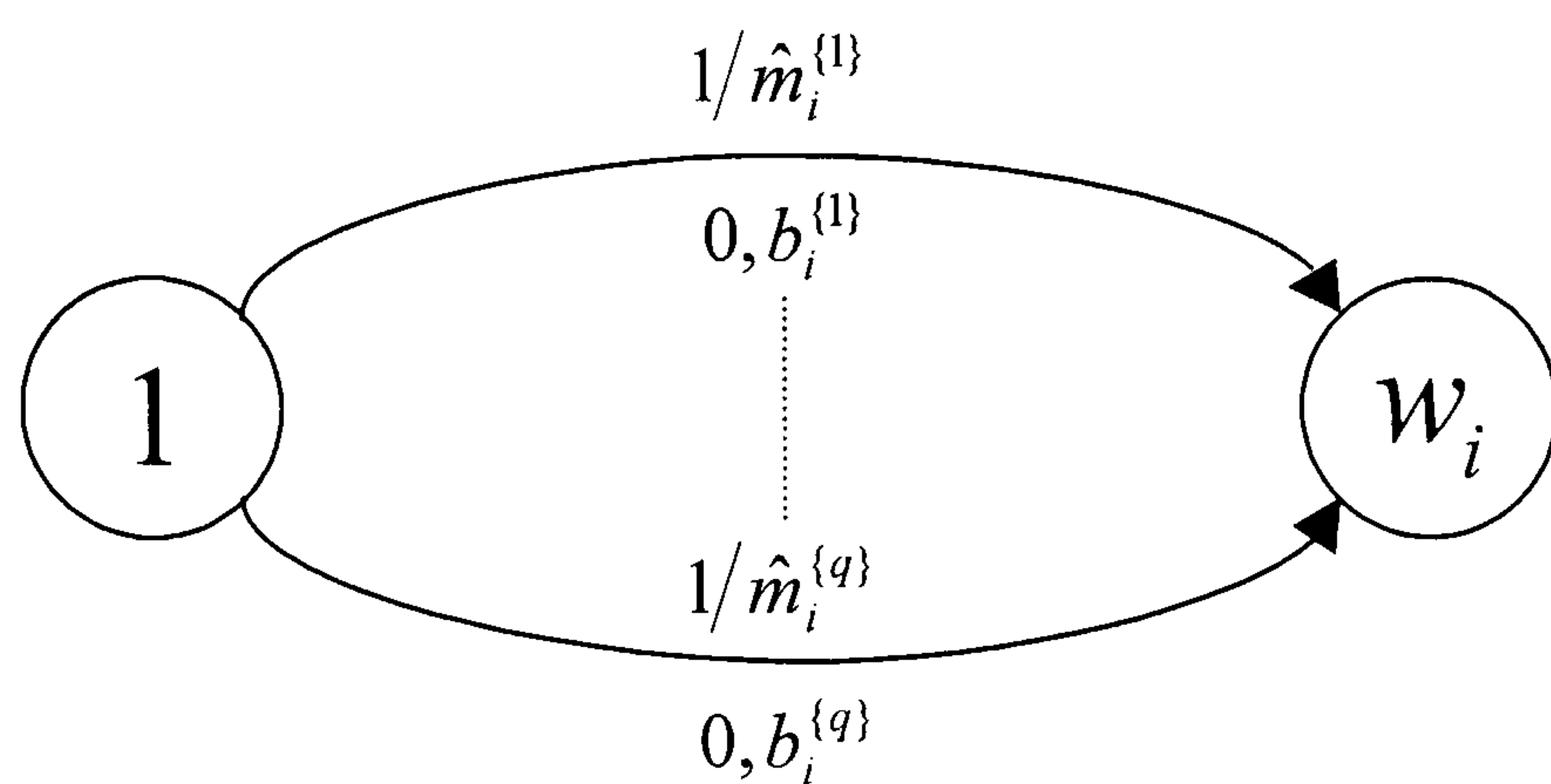


Figure 4.6: Network representation of the budget constraint.

The first part is represented by the set of thin arcs on the left of the figure and by the set of the respective nodes. This subgraph represents the budget constraint [4.21]. Note that the arc multipliers are the reciprocals of the liquidity cost function $m(g)$. This is because when one unit is invested into asset i at the beginning of the period, after considering the liquidity cost, only $1/\hat{m}_i^{\{p\}}(g_i)$ units are actually purchased. Since $m(g)$ was assumed to be increasing in w , then the sequence of multipliers, $\{1/\hat{m}_i^{\{p\}}\}_p$, associated with each segment of the PL approximation will be decreasing in p ⁶.

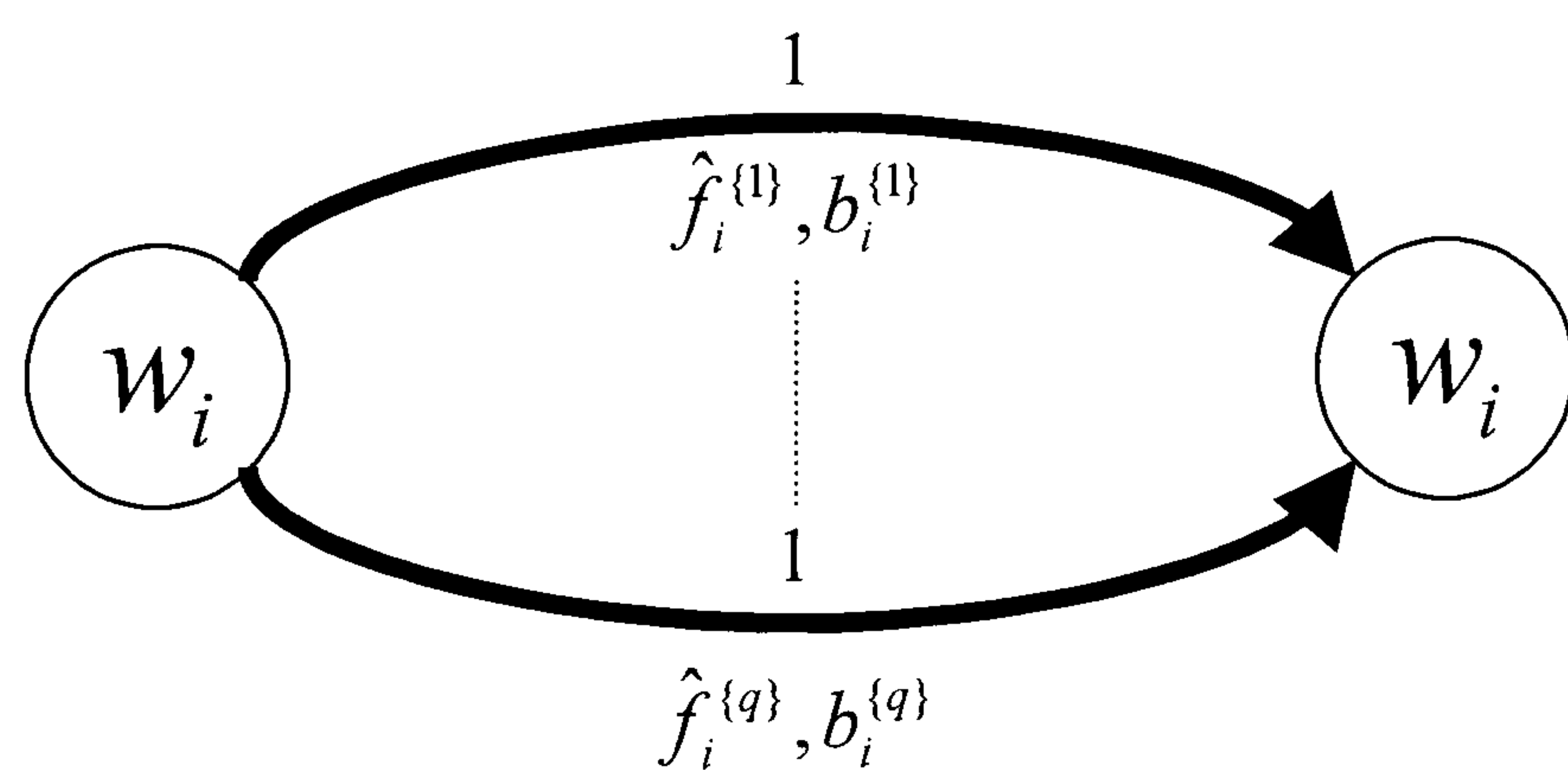


Figure 4.7: Network representation of the objective function.

The central part of the figure illustrates the objective function. All the arcs have unit multiplier and the cost associated with each arc is positive.

Furthermore, the series of cost coefficients $\{\hat{f}_i^{\{p\}}\}_p$ associated with the PL approximation is increasing in p , since the original cost function is convex.

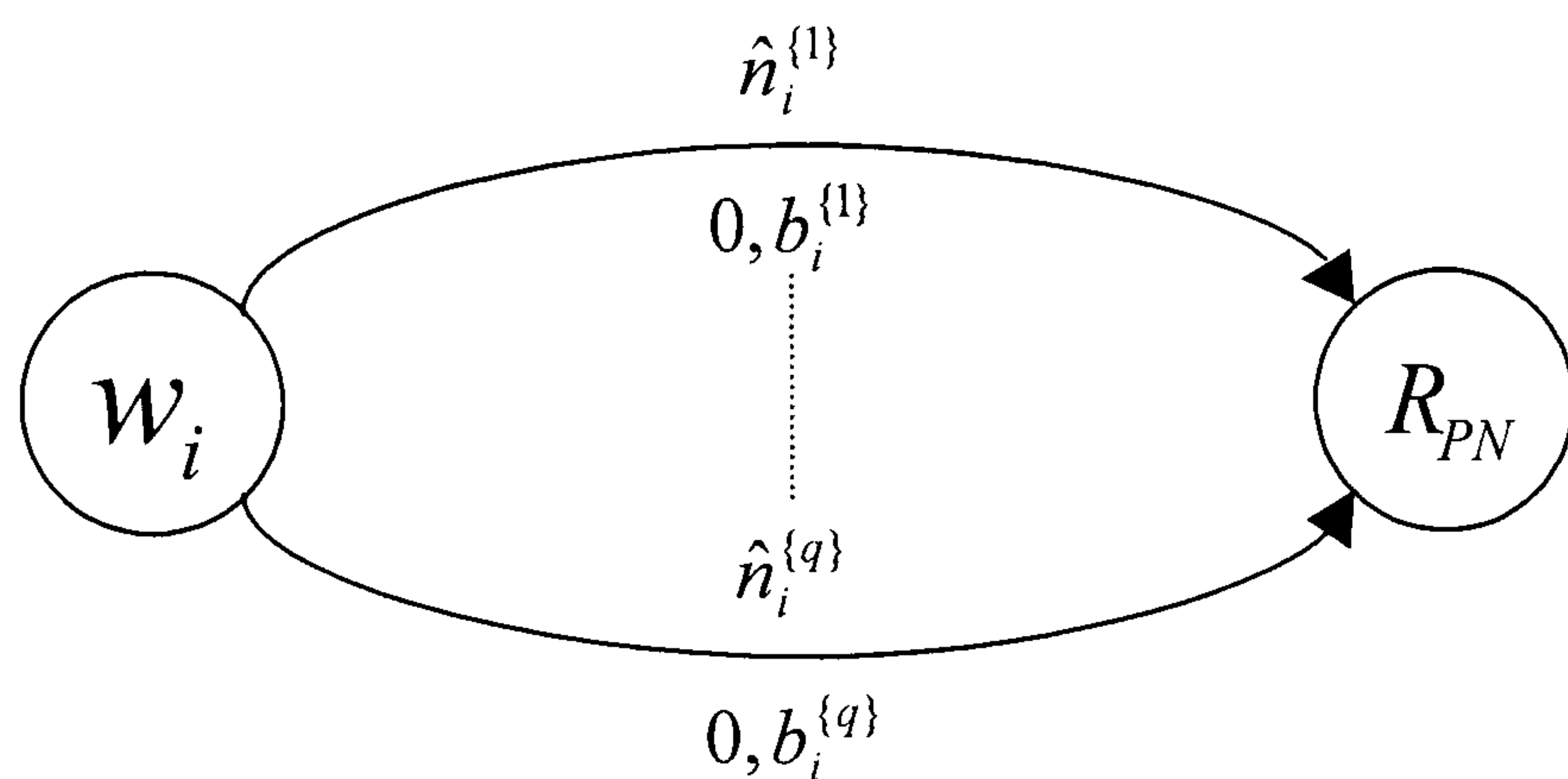


Figure 4.8: Network representation of the constraint on the expected portfolio return.

Finally the last part of the network represents the constraint on the expected return of the portfolio. All the arc costs are zero and the sequence, $\{\hat{n}^{\{p\}}\}_p$, is decreasing in p ⁷.

When the non-linear functions in the constraints and in the cost function are approximated using the technique illustrated in [4.17]-[4.19], each non-linear arc is substituted by q parallel arcs. This procedure basically converts a convex generalised network flow problem into a generalised minimum cost flow problem but it expands the underlying network structure considerably. There will be $3n$ arcs in the non-linear formulation of the problem and $3nq$ arcs in the linear formulation of the problem.

⁶ $\hat{m}_i^{\{p\}}$ is the p -th slope of the linear approximation of $m_i(g_i) = (1 + s_{0,i}(g_i))$, which is increasing in g .

⁷ $\hat{n}_i^{\{p\}}$ is the p -th slope of the linear approximation of $n_i(w_i) = (1 + r_i)(1 - s_{1,i}(w_i(1 + r_i)))$, which is decreasing in w .

Note though that we can use the contiguity of the solution in order to reduce the number of arcs that must be maintained by the algorithm. A solution is said to be contiguous if

$$\begin{cases} w_i^{\{p\}} > 0 \Rightarrow w_i^{\{p-j\}} = b^{\{p-j\}} & \forall j = 1, 2, \dots, p-1 \\ w_i^{\{p\}} = 0 \Rightarrow w_i^{\{p+j\}} = 0 & \forall j = 1, 2, \dots, p-q \end{cases}$$

The contiguity property applies to both the cost arcs, where the non-linearity derives from the cost function, and the constraint arcs, where there are non-linearities in the flow multipliers. The network can be divided into three subgraphs, by adding appropriate collector nodes and additional synthetic arcs, to obtain two subgraphs consisting of the constraint arcs, which are maximum flow problems, and a subgraph consisting of the cost arcs, which is a minimum cost flow problem. In the first case the costs are zero and the arc multipliers are a decreasing sequence. In the optimal solutions therefore all the possible flow will be directed to the arcs with the highest multipliers until their respective upper flow bounds are reached. Similarly, in the second case the costs are an increasing sequence and all the multipliers are equal to +1. Therefore the flow will be directed to the arcs with the lowest cost up to the level specified by the upper flow bounds. For a more formal proof consider a non-contiguous solution for which $w_i^{\{l\}} > 0$, and, for some $p < l$, $w_i^{\{p\}}$ is strictly less than its upper bound. Since $\hat{f}_i^{\{p\}} < \hat{f}_i^{\{l\}}$, by the convexity of the objective function, we can improve the solution by adding a small number ϵ to $w_i^{\{p\}}$ and subtracting it from $w_i^{\{l\}}$. Therefore a non-contiguous solution can not be an optimal solution.

Another important characteristic of the network structure is that the graph is directed from the source node to the end node. As we will illustrate in the

following sections this property allows us to simplify the selection of the leaving arc in the primal simplex algorithm.

The above model is the network formulation of a one period asset allocation problem where the investor has only cash at the beginning of the period. If the problem relates to portfolio rebalancing, the following arcs and nodes need to be added to the left of the graph.

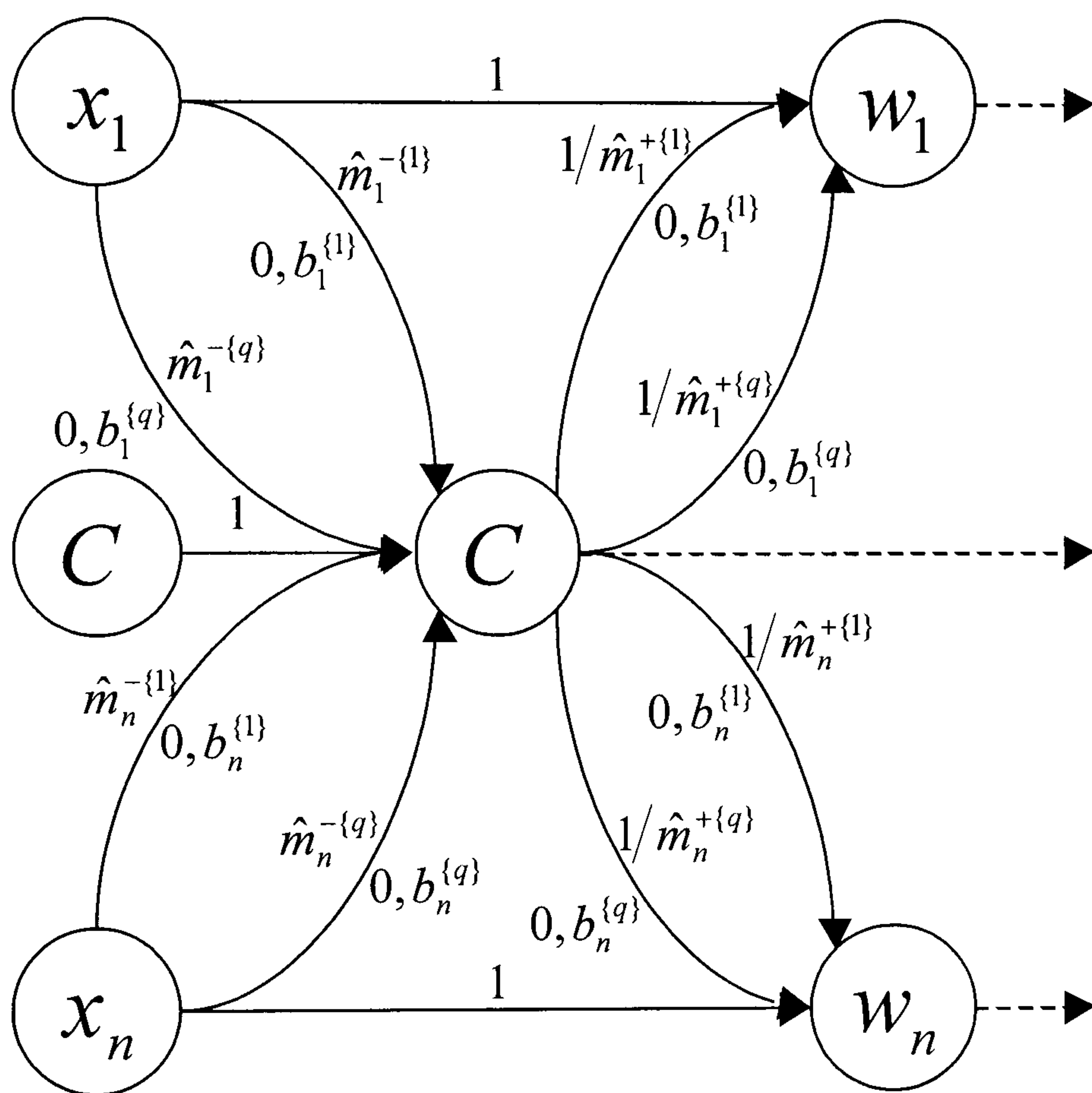


Figure 4.9: Subgraph representing portion of the portfolio rebalancing problem.

In the graph above the nodes on the left represent the supply nodes: the nodes denoted by x_i are the amounts already invested in asset i , the nodes denoted by C represent the amount invested in cash. The node C in the middle of the figure is a transshipment node as are the nodes w_i . Note that the liquidity cost function applied to the arcs on the left of the figure is $m_i^-(g_i) = 1 - s_{0,i}(g_i)$ which implies a decreasing sequence of arc multipliers. In a similar way it is possible to extend the network representation of the problem to a dynamic

multi-period asset allocation. The full network for the portfolio rebalancing problem is illustrated in Figure 4.10.

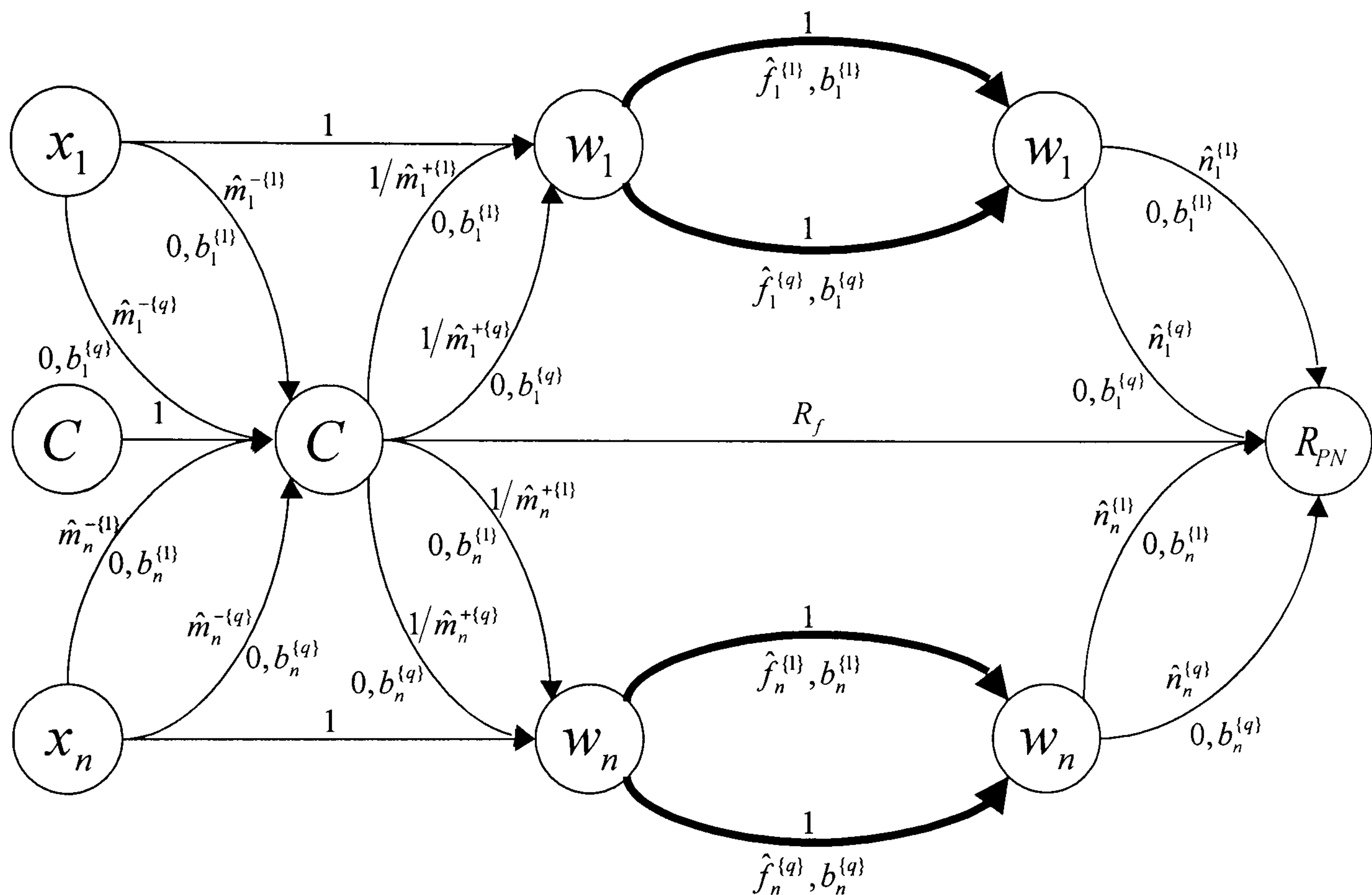


Figure 4.10: Network representation of the portfolio rebalancing problem.

4.4. Generalised Network Simplex Algorithm

In this section we will introduce the generalised network simplex algorithm, one of the fastest available algorithms for solving generalised network flow problems in practice. The algorithm terminates finitely, even though the number of iterations it performs can not be bounded by a polynomial or pseudo-polynomial function of the input size parameters. Its worst-case running time is exponential, but in practice its computational time grows much slower than $O(nm)$, where n is in this case the number of nodes, and m is the number of arcs. Empirical tests have found the generalised network simplex algorithm to be about two to three times slower than the network simplex algorithm.

The generalised simplex algorithm is an adaptation of the linear programming simplex method. This adaptation is possible because of the special topological structure of the basis.

Let $G=(N,A)$ denote a graph consisting of the set N of nodes and the set A of arcs. A typical minimum cost generalised network flow problem is represented below. In this section we will denote each arc by the ordered pair of integers representing the source node and the end node, even though this approach is inappropriate when parallel arcs are included in the network. The algorithms illustrated below can easily handle parallel arcs as well.

$$\min_{x_{ij}} \sum_{(ij) \in A} c_{ij} x_{ij} \quad [4.24]$$

s.t.

$$\sum_{\{j:(i,j) \in A\}} x_{ij} - \sum_{\{j:(j,i) \in A\}} \mu_{ji} x_{ji} = b_i \quad \forall i \in N \quad [4.25]$$

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad \forall (i,j) \in A \quad [4.26]$$

Or, using matrix notation

$$\min_x c^T x \quad [4.27]$$

s.t.

$$Mx = b \quad [4.28]$$

$$l \leq x \leq u \quad [4.29]$$

Where:

x_{ij} is the flow on arc (i,j) .

c_{ij} is the cost associated to arc (i,j) .

μ_{ij} is the flow multiplier on arc (i,j) .

l_{ij} and u_{ij} are the lower and upper bounds on the flow on arc (i,j) .

b_i is the demand/supply of node i , such that for $b_i > 0$ node i is a supply node, for $b_i < 0$ is a demand node, and for $b_i = 0$ is a transshipment node.

M is the node-arc incidence matrix where each column represents an arc, and each row a node. M is such that, if arc (i,j) is represented by the n^{th} column, $M(i,n)=1$, $M(j,n)=-\mu_{ij}$, and all other elements are 0.

The expressions [4.25] and [4.28] are the mass balance constraints. They require that the total inflow minus the total outflow of each node is equal to the node supply/demand.

In our problem (see Figure 4.5), the cost vector, c , has zero elements for the "constraint" arcs and a value given by $f(w)$ for the "cost" arcs. The flow multipliers, μ , are 1 for the "cost" arcs and are either $1/m(w)$ or $n(w)$ for the "constraint" arcs. The demand/supply vector, b , is 1 at the root node, $-R_{PN}$ at the terminal node and 0 for all the other nodes. The vector of optimal weights, which is the solution to the problem, is given by the subset of flows on the "cost" arcs.

Considering the Lagrangian for [4.24]-[4.25]

$$\begin{aligned}
 L(x, \pi) &= \sum_{(i,j) \in A} c_{ij} x_{ij} + \sum_{i \in N} \pi_i \left(b_i - \sum_{\{j: (i,j) \in A\}} x_{ij} + \sum_{\{j: (j,i) \in A\}} \mu_{ji} x_{ji} \right) = \\
 &= \sum_{(i,j) \in A} x_{ij} (c_{ij} - \pi_i + \mu_{ij} \pi_j) + \sum_{i \in N} \pi_i b_i
 \end{aligned}
 \tag{4.30}$$

The dual problem is:

$$\underset{\pi}{\text{Max}} \underset{x}{\text{Min}} L(x, \pi) \quad [4.31]$$

As we have seen in the previous section a (good) augmented forest is the equivalent of a basis for a generalised network flow problem. An augmented forest is a subgraph $G^F(N, F)$, with $F \subseteq A$, defined as a collection of node-disjoint augmented trees that span all the nodes of the graph. An augmented tree is a subgraph $G^A(N^A, T^A)$ of G so that $N^A \subseteq N$ and $T^A \subseteq A$, where T^A is a spanning tree of the node set N^A together with an additional arc called the extra arc. The extra arc produces exactly one cycle called the extra cycle. A good augmented tree is an augmented tree whose extra cycle is not a breakeven cycle (i.e. the multiplier of the cycle is different from one). A good augmented forest is an augmented forest where all the components are good augmented trees.

An augmented forest defines a partition of the arcs of the graph into three sets (F, L, U) , where F is an augmented forest, L is the set of the arcs for which the flow x_{ij} is at the lower bound, l_{ij} , and U is the set of arcs for which the flow x_{ij} is at the upper bound, u_{ij} . There is a one-to-one relation between an augmented forest partition (F, L, U) and a solution of the generalised network flow problem. A number π_i is associated with each node i of the graph and is called the node potential. The variables π_i are equivalent to the dual variables in a LP problem. With respect to the node potentials a reduced cost is obtained for each arc in the graph of the form:

$$c_{ij}^\pi = c_{ij} - \pi_i + \mu_{ij} \pi_j \quad [4.32]$$

Where μ_{ij} denotes the multiplier of the arc ij . A flow x^* is optimal if for some vector π the following conditions are satisfied

$$\begin{cases} l_{ij} < x_{ij} < u_{ij} & \Rightarrow & c_{ij}^{\pi} = 0 \\ x_{ij} = l_{ij} & \Rightarrow & c_{ij}^{\pi} \geq 0 \\ x_{ij} = u_{ij} & \Rightarrow & c_{ij}^{\pi} \leq 0 \end{cases} \quad [4.33]$$

The above conditions are equivalent to the complementary slackness conditions in a LP problem. The generalised network simplex algorithm maintains a feasible good augmented forest structure at every iteration and successively transforms it into an improved augmented forest structure until it becomes optimal.

An initial feasible augmented forest can be attained by adding artificial arcs (i,i) to each node i in the network, with a very large cost K and a multiplier of 0.5, if the node is a supply node, and of 2 otherwise. Therefore it is possible to consume the excess supply (demand) of each node by sending flow along the arc (i,i) , that will create a cycle with a gain smaller (greater) than 1. The flow on each of these loops will be $x_{ii} = e_i / (1 - \mu_{ii})$, where e_i represents the excess supply (demand). Moreover, since each artificial arc has a sufficiently large cost K , no solution with a positive flow on any artificial arc will be optimal unless the generalised network flow problem is infeasible.

The initial node potentials are determined from the fact that the reduced cost of each arc in F must be zero. At each iteration, the algorithm adds one arc to the spanning tree in place of one of its current arcs. The entering arc is a non-tree arc violating its optimality condition. The algorithm can be summarised in the following steps.

- (1) Add a non-tree arc (an arc in L for which $c_{ij}^{\pi} < 0$ or an arc of U for which $c_{ij}^{\pi} > 0$) to the spanning tree, creating a spanning tree with a double loop.

- (2) Send the maximum possible flow into the loops until the flow on at least one arc in the cycle reaches its lower or upper bound.
- (3) Drop an arc whose flow has reached its lower or upper bound, giving a new spanning tree structure.
- (4) Update the node potentials for those nodes that belong to the augmented tree(s) involved in the arc substitution.

The network simplex algorithm maintains a feasible basis structure at each iteration and successively modifies the basis structure via pivots until it becomes an optimum basis structure. Because of its relationship to the primal simplex algorithm for linear programming, this operation of moving from one spanning tree structure to another is known as a pivot operation. The special structure of the basis enables the simplex computations to be performed efficiently. The algorithms involved in the flow calculation and in the computation of the node potentials of an augmented tree have a complexity of $O(\text{number of arcs})$ and $O(\text{number of nodes})$ respectively. Following Ahuja et al. (1993), the pseudo-code for the algorithm is as follows:

Algorithm Generalised Network Simplex

Begin

Determine an initial feasible augmented forest structure (F, L, U)

While Some arc in L or in U violates the optimality conditions [4.33]

 Select an entering arc (k, l) violating its optimality condition⁸

 Add arc (k, l) to the augmented forest F

Call Find Leaving Arc

Call Update Flows

Call Update Node Potentials

 Update Tree Indices

End

End

Algorithm Find Leaving Arc

Begin

Set the flow on the entering arc $(k, l) = 1$

Call Update Flows, and obtain the flows y_{ij} that clear the flow imbalances created by the entering arc

Compute the maximum flow multiplier, δ , for which the flow of each arc remains between the bounds, such that

$$0 \leq x_{ij} + \delta y_{ij} \leq u_{ij} \quad \forall (i, j) \in F \cup (k, l)$$

Eliminate the arc, which reaches its bounds

End

Algorithm Update Flows

Begin

Send a flow of θ on the extra arc and calculate node imbalances using all the arcs $\in U$

For Each Leaf Node (moving from the leaf nodes to the root node)

 Compute flow (as a function of θ) to clear node imbalance.

End

Calculate θ that clears the imbalance of the root node created by the flow on the extra arc

Calculate all the other flows substituting θ

End

Algorithm Update Node Potentials

Begin

Set the node potential of the root node $= \theta$

For Each Child Node (moving from the root node to the leaf nodes)

 Calculate node potentials as a function of θ such that

$$c_{ij}^{\pi} = c_{ij} - \pi_i + \mu_{ij} \pi_j = 0$$

End

Calculate θ that solves $c_{ij}^{\pi} = 0$ for the extra arc

Calculate all the other node potentials substituting θ

End

⁸ There are several rules to select the entering arc. Dantzig's pivot rule selects the arc with the maximum violation, represented by $|c_{ij}^{\pi}|$. The first eligible arc pivot rule selects the first arc with positive violation. The candidate list pivot rule maintains a list of arcs with positive violations, and selects the arc with the maximum violation from the list.

4.5. Parallel Arcs

Several algorithms other than the primal simplex are available for solving network flow problems with piecewise linear functions. In a study from Marins et al. (1996), comparing different routines, it turns out that the Primal method is significantly faster than the Out-of-Kilter method which in turn is faster than the Cost Scaling method and the Dual method. Details of the algorithms are in the paper of Marins et al. (1996).

The parallel arcs resulting from a piecewise linear problem can either be considered explicitly or implicitly by the algorithm. Because of the contiguity of the solution there is a one-to-one relation between the solution to a problem with concave piecewise linear functions and the solution to the problem with parallel arcs.

If arcs are considered explicitly no modification is required to the algorithm, but it is possible to further improve the performance by considering the specific problem structure. As we have seen previously, if the functions underlying the arc multipliers are concave and the cost functions are convex, an optimal solution will always be a contiguous solution. This implies that it is not necessary to maintain all the parallel arcs between each pair of nodes.

At each iteration only the segments immediately preceding and following the last segment with a positive flow need to be maintained. Therefore for each arc (i,j) whose residual capacity is different from zero, three parallel arcs will be maintained between node i and node j . Two arcs will be maintained between each contiguous pair of nodes. At each iteration, the algorithm sends the maximum possible amount of flow along the loops created by introducing

the entering arc. The flow augmentation operation will make some of the flows with non-zero residual capacity on the loops reach their respective flow bounds. For the pair of nodes connected by these arcs only two parallel arcs need to be maintained, therefore some arcs will be eliminated from the network. Similarly, some flows that were at their lower or upper bound before the flow augmentation might attain a non-zero residual capacity, and some arcs will be added to the network. Given that the solution is contiguous all the arcs that are added to or deleted from the network belong to either the set U or the set L . Therefore the augmented forest structure is unchanged after the arcs are updated.

It is possible that the new entering arc will connect the same nodes as the leaving arc. If so, we augment flow along the same loop until a different leaving arc is found. We could identify the amount of flow we could send along the loops in one of two ways: 1) by sending flow repeatedly along the loops, or 2) by performing a binary search in the interval $[0, u_{MAX}]$, where u_{MAX} is the maximum capacity allowed on the augmented tree when the last segments are considered.

Every cycle consisting only of the cost nodes is a breakeven cycle, since all the multipliers are 1. Therefore no good augmented tree will contain such a cycle. If an entering arc is a cost arc then the leaving arc will be a cost arc on the same loop. Also, all the augmented trees have to include either the start node or the end node or both. Since an augmented forest spans all the nodes in the network, the forest will be composed of either two augmented trees or of a single augmented tree spanning the full network.

When there are two spanning trees, the only possible cycle is the one created by two parallel "constraint" arcs (we can not have a cycle using the "cost" arcs). Since the solution is contiguous, one of these two arcs will be either at

its lower limit or at its upper limit, and the augmented forest is degenerate. Therefore, the only non-degenerate solution consists of a single spanning tree with the addition of one extra arc, which creates a large cycle that includes both the root node and the terminal node.

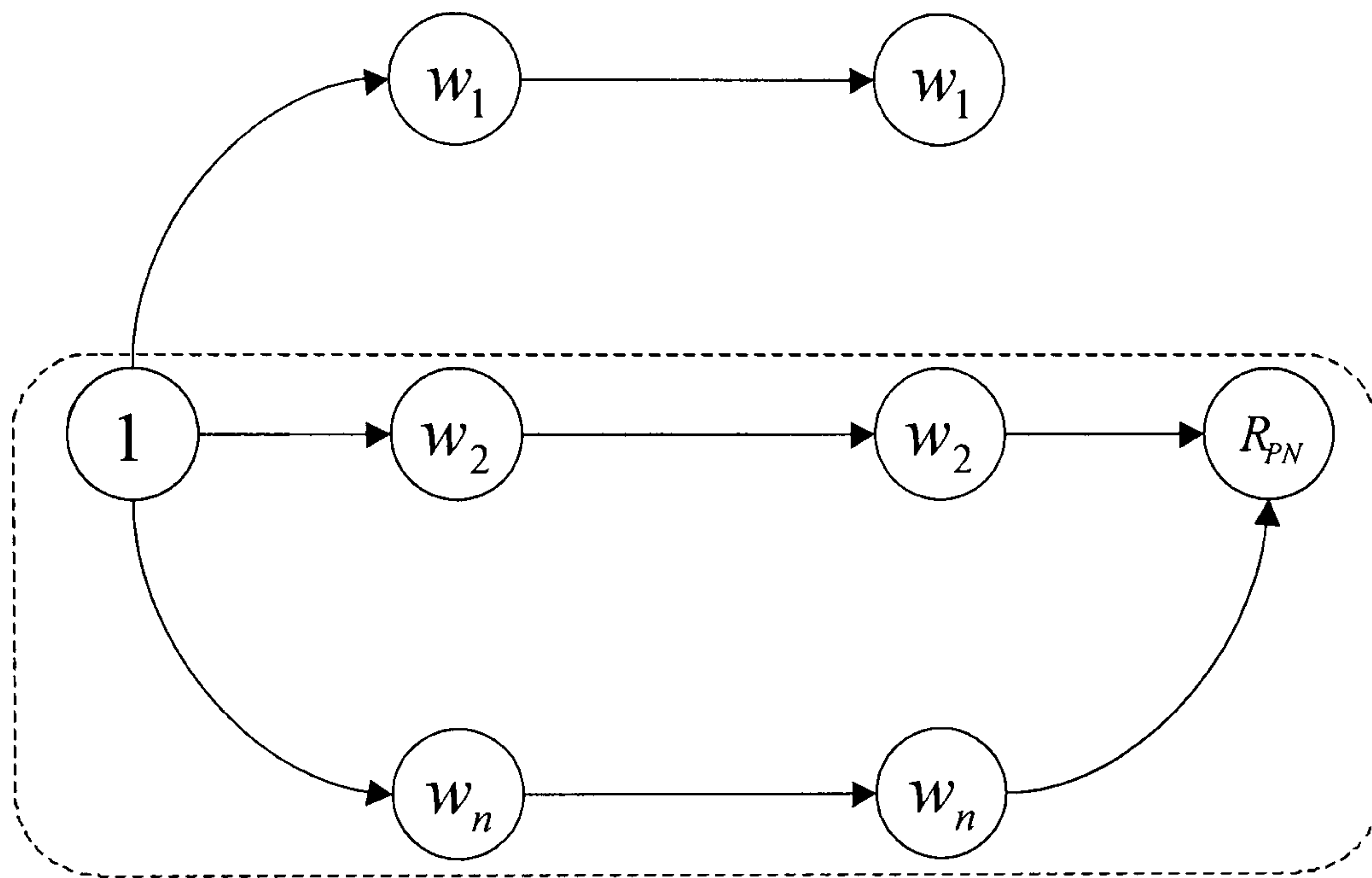


Figure 4.11: Non degenerate augmented forest⁹.

No loop can involve two parallel arcs. As a consequence, the pivoting will either exchange one parallel arc for another (which corresponds to increasing or decreasing the flow between the pair of nodes involved by moving along the next or previous segments of the PL approximation), or it will change one of the arcs creating the large loop between the root node and the terminal node. In both cases it is simple to identify the leaving arc.

When parallel arcs are considered implicitly, the algorithm needs to be modified slightly. We keep a single arc between each pair of nodes but two reduced costs are associated with the non-forest arcs: a left reduced cost,

⁹ A basis for the problem can only consist of a single spanning tree with an extra arc creating a large loop between the root node and the terminal node.

relating to a reduction in the flow on arc (i,j) , c_{ij}^- , and a right reduced cost, relating to an increase in the flow on the arc (i,j) , c_{ij}^+ .

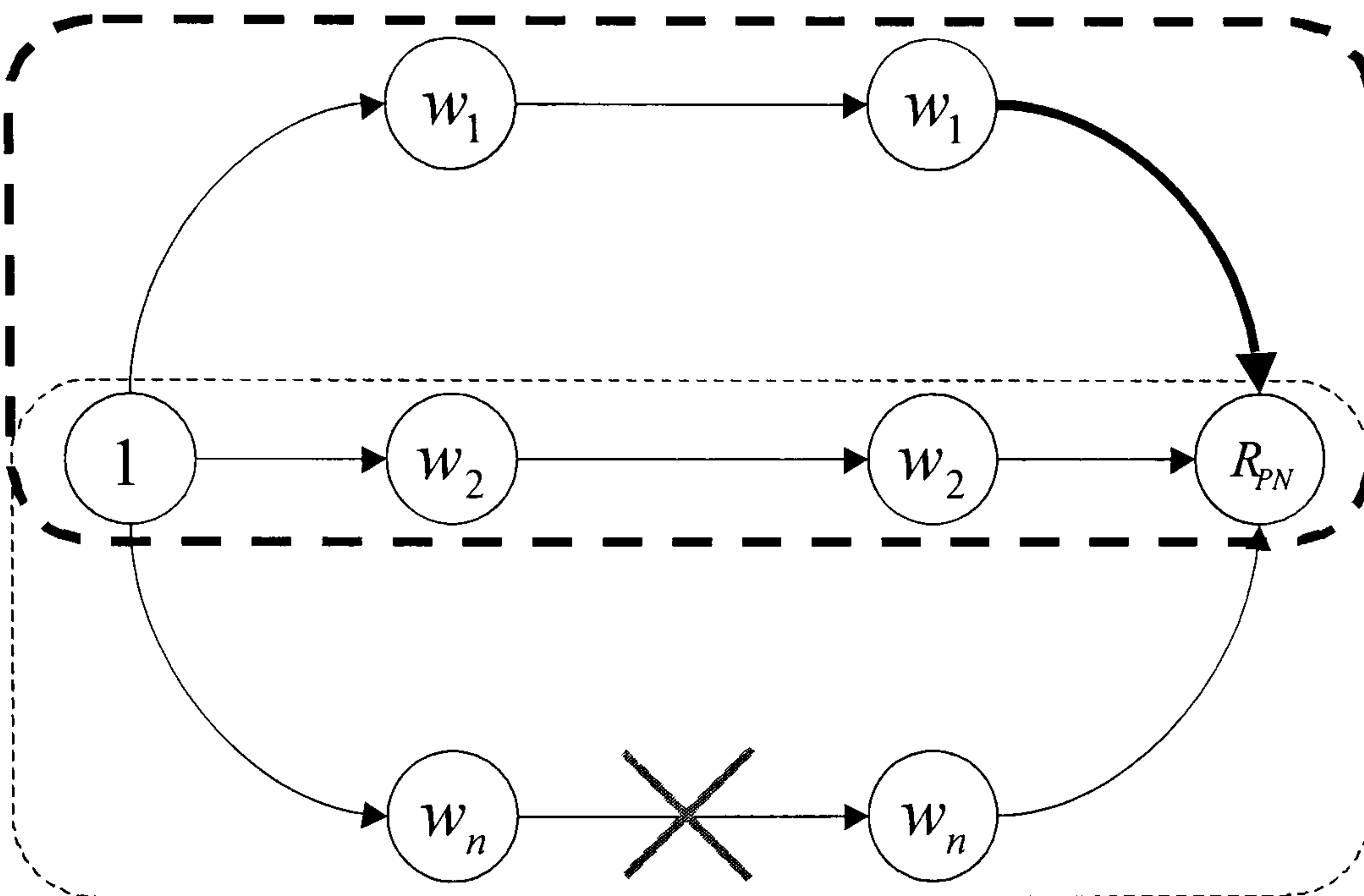


Figure 4.12: Pivot operation¹⁰.

From the complementary slackness conditions it follows that a feasible basic solution, associated with an augmented forest, is optimal if and only if, for every non-basic arc $(i,j) \notin F$, $c_{ij}^- \leq 0$ or $c_{ij}^+ \geq 0$. Note that $c_{ij}^- \leq 0$ and $c_{ij}^+ \geq 0$ can not occur simultaneously given the contiguity of the solution. Let us define the total reduced cost as $c_{ij}^\pi = c_{ij}^+$ if $c_{ij}^+ < 0$; $c_{ij}^\pi = c_{ij}^-$ if $c_{ij}^- > 0$. Non-zero values of the reduced costs belong to non-basic arcs, which are candidates to improve the current solution. If no reduced cost is different from zero the algorithm terminates. Otherwise, we select an arc with non-zero reduced cost, and augment unit flow in the appropriate direction. In fact, as in the previous case, we might perform a binary search to augment more flow in a single step.

¹⁰ The entering arc (bold) creates a new large cycle (bold). The leaving arc has to be one of the 9 arcs connecting the root node to the terminal node.

4.6. Computational Results

The algorithms presented in this and other chapters were coded in Matlab and the tests were run on an AMD Athlon 1.4 GHz with 256MB of memory, under Windows NT. We used three data sets. The first set is the 4-asset example illustrated in the previous chapter with data from the FTSE100, the second set consists of 10 assets from the S&P500, and the third set of 30 assets from the Bloomberg Euro 500 index. The assets have been selected randomly from the above indexes, dropping stocks with missing values. We used Bloomberg to obtain weekly prices from March 1992 to September 1997. For each data set we considered two different liquidity structures: the first one, with liquidity coefficients of 3% for the initial period and 5% for the end period, and the second one with coefficients of 10% and 25%. We examined 50 equally spaced points on the efficient frontier by varying the required rate of return. Each optimisation is initialised using always the same starting point, represented by the portfolio with equally weighted assets.

In this section we compare three different optimisers. The first one, *fmincon*, is the non-linear optimiser from the Matlab Optimisation Toolbox. This optimiser is used as a benchmark. The algorithm tries to find a solution of the Karush-Kuhn-Tucker equations associated with the problem using a Sequential Quadratic Programming method. An estimate of the Hessian of the Lagrangian is updated at each iteration using the BFGS formula. A line search is performed using a merit function similar to that proposed by Powell (1978). The QP subproblem is solved using an active set strategy similar to that described in Gill (1991). The second one, *mma*, is based on the GCMMA method described in section 4.1, where each convex separable subproblem is solved using a primal-dual interior-point approach, in which a sequence of relaxed Karush-Kuhn-Tucker conditions are solved by Newton's method. The third optimiser, *mma pla*, uses GCMMA, but the subproblems are first

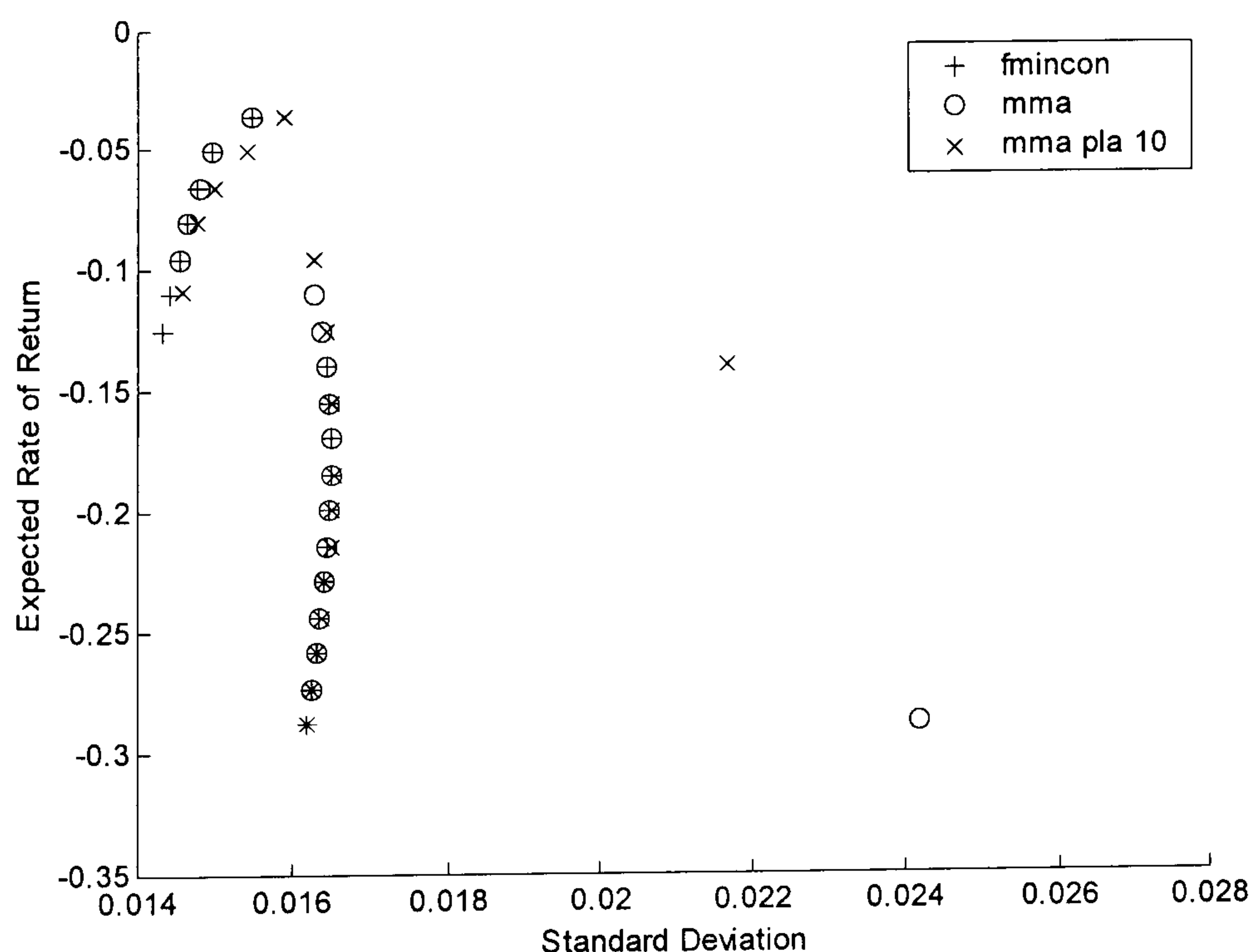
linearised and then solved using the network flow algorithm illustrated above. In the tests we used the standard version of the generalised network simplex algorithm without the adaptations considered in the previous section. We consider two different linearisations, one using 10 segments for each non-linear function and one using 30 segments.

Errors are measured by considering the percentage distance in the standard deviation for a given return from the minimum standard deviation achieved by all the algorithms examined. Note that the errors provide us only with a relative accuracy measure, useful to compare different routines, but they do not give any indication regarding the distance from the exact efficient frontier. The results are shown in Table 4.1.

		FMINCON	MMA	MMA PLA 10 Segments	MMA PLA 30 Segments
4 Assets LC0=0.03 LC1=0.05	Mean percentage error	0.013%	0.034%	1.049%	0.040%
	Median percentage error	0.001%	0.016%	0.028%	0.003%
	Time (s)	15.7	21.8	4.4	7.41
	Missed solutions	0	0	0	0
4 Assets LC0=0.1 LC1=0.25	Mean percentage error	0.000%	0.037%	0.208%	0.022%
	Median percentage error	0.000%	0.030%	0.060%	0.010%
	Time (s)	11.9	15.9	4.9	6.86
	Missed solutions	0	0	1	1
10 Assets LC0=0.03 LC1=0.05	Mean percentage error	0.022%	2.177%	3.273%	1.906%
	Median percentage error	0.001%	0.038%	0.109%	0.012%
	Time (s)	60.6	129.9	22.7	31.06
	Missed solutions	0	0	1	1
10 Assets LC0=0.1 LC1=0.25	Mean percentage error	0.000%	1.433%	1.356%	0.936%
	Median percentage error	0.000%	0.026%	0.374%	0.021%
	Time (s)	42.6	69.8	18.0	28.64
	Missed solutions	0	0	1	0
30 Assets LC0=0.03 LC1=0.05	Mean percentage error	0.463%	2.032%	1.810%	1.508%
	Median percentage error	0.000%	0.033%	0.491%	0.016%
	Time (s)	737.0	1127.4	188.6	367.60
	Missed solutions	0	0	3	2
30 Assets LC0=0.1 LC1=0.25	Mean percentage error	2.509%	0.631%	3.090%	3.176%
	Median percentage error	0.000%	0.000%	0.419%	0.111%
	Time (s)	750.4	2043.1	223.9	312.08
	Missed solutions	0	0	2	1

Table 4.1: Computational results for MMA and MMA PLA.

Solving the linearised subproblems with the network flow algorithm described in this chapter results in an improvement in performance of 5 to 10 times. As expected there is a slight loss in accuracy, but less than what we expected. This is because, when a solution converges, the asymptotes get closer to one another. This, in turn, restricts the domain of the subproblem and since the number of breakpoints is always the same it results in piecewise linear approximations which are progressively better. However, using piecewise linear approximations, it might happen that the MMA algorithm oscillates, and, in general, it requires more iterations in order to converge. Moreover, since the approximations are conservative it might happen that a feasible solution exists for the original problem but it will not be feasible in the approximated problem. Both issues are attenuated by using more segments. Figure 4.13 shows the solutions of the three algorithms for 10 assets with liquidity coefficients of 0.1 and 0.25. Figure 4.14 considers the problem with 30 assets and liquidity coefficients of 0.1 and 0.24. The solutions for the problem with 10 assets and liquidity coefficients of 0.1 and 0.25 are illustrated in Figure 4.13.



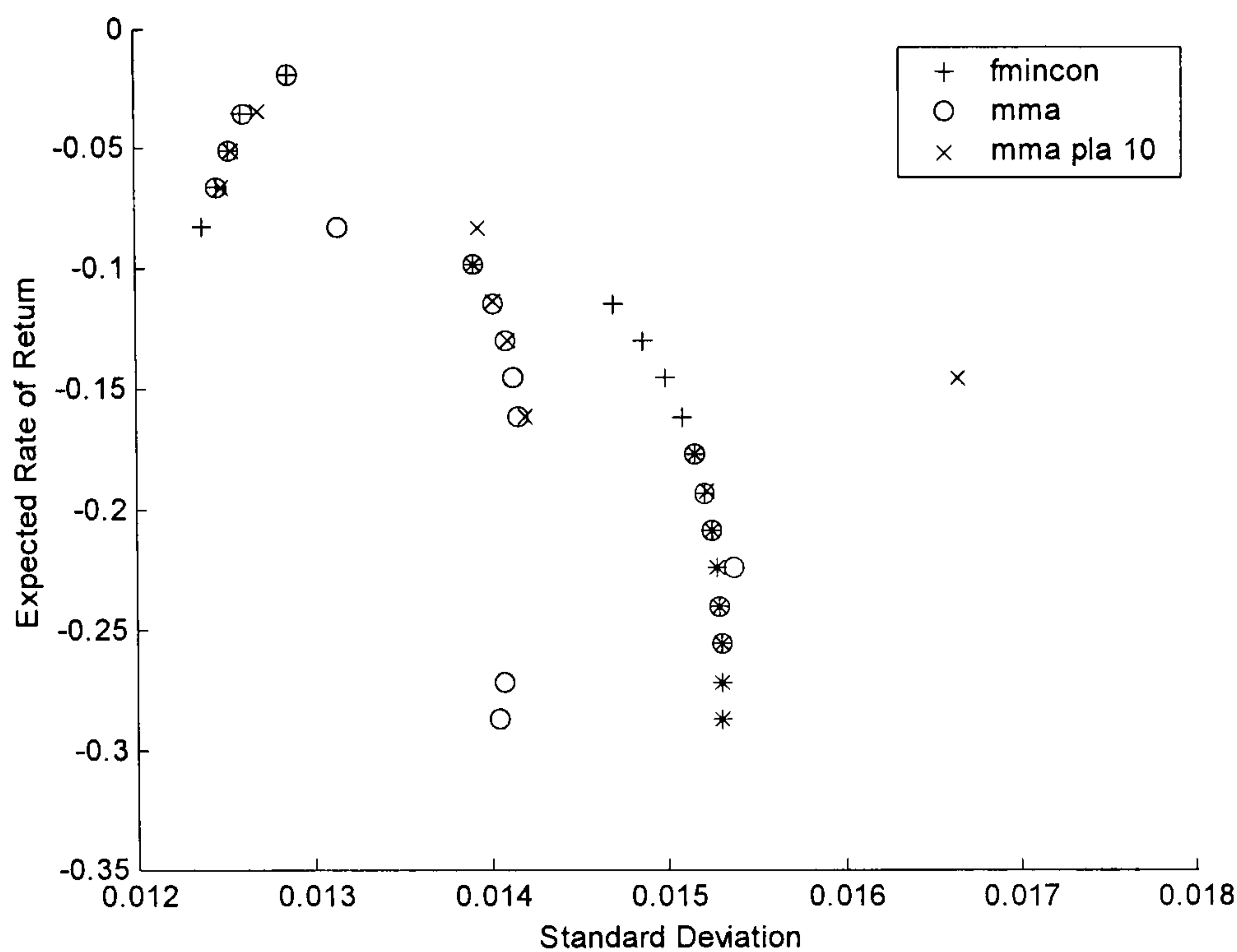


Figure 4.14: 30 Assets (LC 0.1 and 0.25).

When there is a single global optimum our results show that *mma* is equivalent to *fmincon* even though in our implementation it is slightly slower. With multiple local minima (i.e. high liquidity coefficients), as expected, both algorithms are trapped in the local minimum closer to the starting point. However, *mma* can occasionally find a better solution than *fmincon* (see Figure 4.14). Compared to *fmincon*, *mma* converges slowly in the proximity of the optimal point. Using piecewise linear approximations, we observed an improvement in terms of efficiency (with respect to *mma*), even though such gains were often offset by a loss of accuracy. The performance improvement is less evident when comparing *mma pla* to *fmincon*, and unless the algorithm is further optimised¹¹, the marginal improvement in efficiency does not compensate for the higher mean and median errors.

¹¹ It must be noted that the primal simplex algorithm makes heavy use of loops. Such implementation is not very efficient in Matlab, which is a matrix-orientated language.

5. Space Mapping

In this chapter we present an extension of the Space Mapping algorithm in order to find a sequence of quadratic subproblems that can be solved using available algorithms. The original model was presented by Bandler et al. in 1994, with the aim of establishing a link between coarse and fine models of complex physical phenomena, directing the bulk of computer evaluations to the coarse model while preserving the accuracy offered by the fine model. In section 5.1, we expose the original formulation of the Space Mapping method. In section 5.2, we show how the algorithm can be adapted in order to solve our non-linear optimisation problem. In section 5.3, we show the computational results for the algorithm.

5.1. Space Mapping Algorithm

Let's assume that two models of the same phenomenon are available, a fine-model, $f_F(x_F)$, and a coarse-model $f_C(x_C)$. The fine-model is accurate but it is difficult to solve, the coarse-model is not as accurate but more efficient computationally. The problem is solved in the coarse-model space, obtaining the solution x_C^* , and then a mapping is sought between the coarse-model and the fine-model so that the optimal coarse solution can be mapped into the fine-model space. The algorithm terminates once the coarse-model response matches a given fine-model response. Parameter extraction is crucial to the algorithm. Non-uniqueness of the extracted parameters may lead to divergence or oscillation of the iterations. To solve this problem a trust-region aggressive space mapping algorithm has been introduced by the author. It must be noted that the optimality of the fine-model can not be guaranteed. This is because the final response matches the optimal coarse-model response, which may be

different from the optimal fine-model response, obtained by solving the original problem directly.

SM aims at establishing a mapping, P , from the fine model parameter space, x_F , to the coarse model parameter space x_C , such that

$$x_C = P(x_F) \quad [5.1]$$

$$\|f_C(P(x_F)) - f_F(x_F)\| < \varepsilon \quad [5.2]$$

Where $\| \cdot \|$ is a suitable norm and ε is an arbitrary scalar number. The error function is defined as

$$d(x_F) = P(x_F) - x_C^* \quad [5.3]$$

Where x_C^* is the solution of the coarse model. The final fine-model design is obtained and the mapping established by solving the non-linear system

$$d(x_F) = 0 \quad [5.4]$$

Let $x_F^{\{k\}}$, be the k^{th} iterate in the solution of [5.4]. In SM, the next iterate is found by a quasi-Newtonian iteration

$$x_F^{\{k+1\}} = x_F^{\{k\}} + h^{\{k\}} \quad [5.5]$$

Where $h^{\{k\}}$ is obtained from

$$B^{\{k\}} h^{\{k\}} = -d(x_F^{\{k\}}) \quad [5.6]$$

$B^{\{k\}}$ is an approximation to the Jacobian of the vector d with respect to $x_F^{\{k\}}$ at the k^{th} iteration.

$$B^{\{k\}} \approx J(x_F^{\{k\}}) = \left[\frac{\partial d(x_F)}{\partial x_F} \right]_{x_F = x_F^{\{k\}}} \quad [5.7]$$

The matrix B is updated at each iteration using Broyden's update.

$$B^{\{k+1\}} = B^{\{k\}} + \frac{d(x_F^{\{k\}} + h^{\{k\}}) - d(x_F^{\{k\}}) - B^{\{k\}} h^{\{k\}}}{h^{\{k\}T} h^{\{k\}}} h^{\{k\}T} \quad [5.8]$$

Given [5.6] we get:

$$B^{\{k+1\}} = B^{\{k\}} + \frac{d(x_F^{\{k+1\}}) h^{\{k\}T}}{h^{\{k\}T} h^{\{k\}}} \quad [5.9]$$

$B^{\{k\}}$ approximates the mapping between the two spaces. In the first iteration B is initialised to the identity matrix. The vector d is obtained by evaluating $P(x_F)$, which is done indirectly through parameter extraction, so that the following problem is solved:

$$\underset{x_C^{\{k\}}}{\text{Min}} \|f_C(x_C^{\{k\}}) - f_F(x_F^{\{k\}})\| \quad [5.10]$$

Where $\| \cdot \|$ is a suitable norm¹. A prediction of the extracted parameters in the k^{th} iteration is given by:

¹ The authors suggest the Huber norm, which is robust against large errors and flexible with respect to small variations in the data.

$$x_C^{\{k\}} = x_C^{\{k-1\}} + B^{\{k-1\}}(x_F^{\{k\}} - x_F^{\{k-1\}}) \approx P(x_F^{\{k\}}) \quad [5.11]$$

This predicted point is taken as a starting point for the parameter-extraction optimisation problem, [5.10]. In the first iteration there is no available information about the mapping between the two spaces. In this case x_C^* can be used as a starting point.

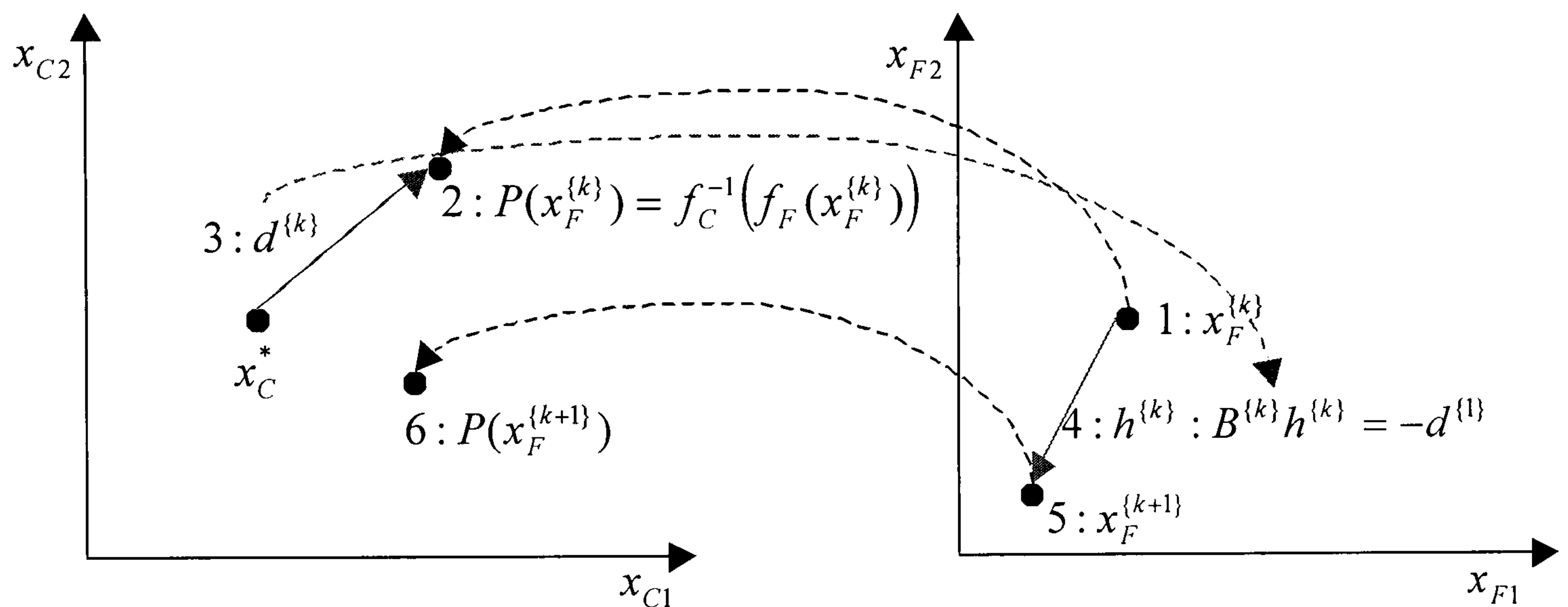


Figure 5.1: Space Mapping Algorithm.

The process is illustrated in Figure 5.1.

1. We initialise $x_F^{\{1\}}$, so that $x_F^{\{1\}} = x_C^*$, where x_C^* is the optimal point obtained by optimising the coarse model.
2. The point in the parameter space of the coarse function, $P(x_F^{\{k\}})$, corresponding to $x_F^{\{k\}}$ is found via parameter extraction².
3. The error function $d^{\{k\}}$ is calculated using the difference between the points $P(x_F^{\{k\}})$ and x_C^* . If $d^{\{k\}} \approx 0$ the algorithm returns $x_F^{\{k\}}$ and exits.

² The notation used in the figure, $f_C^{-1}(f_F(x_F^{\{k\}}))$, does not represent the inverse function itself but the parameter extraction process, due to the non-uniqueness of the solution.

4. The error function $d^{\{k\}}$ is mapped back into the parameter space of the fine function according to the equation $B^{\{k\}}h^{\{k\}} = -d(x_F^{\{k\}})$.
5. A new iterate $x_F^{\{k+1\}}$ is calculated using $x_F^{\{k+1\}} = x_F^{\{k\}} + h^{\{k\}}$.
6. The algorithm then iterates from (2), using the new point $x_F^{\{k+1\}}$.

The algorithm allows us to perform an optimisation using a simplified model and to map the result into the parameter space of the more complex model. At the end of the process, if the algorithm converges, a point x_F^* is obtained which is equivalent to x_C^* . The point x_F^* might not be optimal since a different result might have been obtained by performing the optimisation using the fine model directly.

The non-uniqueness of the parameter extraction problem may lead to divergence or oscillation. To overcome this difficulty a modification of the algorithm is proposed by Bandler et al. (1998). At the k^{th} iteration, the residual vector $d^{\{k\}} = P(x_F^{\{k\}}) - x_C^*$ defines the difference between the vector of extracted coarse-model parameters and the optimal coarse-model solution. The value of $\|d^{\{k\}}\|$ serves as a measure of the misalignment between the two spaces in the k^{th} iteration. Minimising $\|d^{\{k\}}\|$, the k^{th} iteration of the algorithm is given by:

$$(B^{\{k\}T} B^{\{k\}} + \lambda I)h^{\{k\}} = -B^{\{k\}T} d^{\{k\}} \quad [5.12]$$

The parameter λ is such that the step obtained satisfies $\|h^{\{k\}}\| \leq \delta$, where δ is the size of the trust region.

5.2. Space Mapping Algorithm and Asset Allocation

To create a coarse model for the asset allocation problem we proceed as follows³. In chapter 3 we noted that the objective function is of the form:

$$VAR(W_1) = \gamma^T C(r_i, r_j) \gamma + \beta^T C(r_i^2, r_j^2) \beta - 2\gamma^T C(r_i, r_j^2) \beta \quad [5.13]$$

The last two terms on the right hand side of [5.13] are higher order terms (third and fourth order) and can be omitted in an approximation. Furthermore the equation for γ illustrated in Chapter 3 implies that the portfolio weights have a direct effect on the variance and also, by changing the liquidity spread, an indirect effect. Similarly the constraints are in the form $g^T m$ and $g^T n$, where m and n are functions of the gross weights. Only a relatively small part of the variation of both the objective function and the constraint is determined by the indirect effect.

Exploiting this peculiarity, a reasonable approximation scheme might be to fix the spread using the value of the previous iterate, $g^{\{k-1\}}$, and solve the resulting quadratic problem in order to find an optimal $g^{\{k\}}$. Thus, at each iteration, m and n will be vectors of constant terms and the objective function will be represented by $g^T \Sigma_N g$, where Σ_N is calculated from $\gamma(s(g^{\{k-1\}}))$ and the covariance matrix of gross returns, $C(r_i, r_j)$, such that:

$$f_C = g^T \Sigma_N g = g^T D C(r_i, r_j) D g \quad [5.14]$$

Where D is a diagonal matrix whose diagonal elements are given by:

$$D_{ii} = \frac{1}{1 + \alpha_{0,i} g_i^{\{k-1\}}} - 2 \frac{\alpha_{1,i} g_i^{\{k-1\}}}{\left(1 + \alpha_{0,i} g_i^{\{k-1\}}\right)^2} \quad [5.15]$$

When the difference between two consecutive iterates is smaller than a given threshold the algorithm stops. This simple approach can be improved by using the results of the Space Mapping Methodology illustrated above, in order to choose the sequence of $g^{\{k\}}$ so that the series of solutions of the approximate model converge to the solution of the original non-convex model.

The approximation keeps the covariance matrix fixed at the point of the last iteration. Note that for each given vector of weights, all the elements in D are non-negative and Σ_N is positive semidefinite. As a result, the objective function of the subproblem will be a quadratic function.

In the original formulation of the Space Mapping Algorithm the optimal coarse-model solution does not change from one iteration to the other. Furthermore, only the fine-model parameters are updated.

The formulation of the coarse-model [5.14] causes the optimal coarse-model solution, $g_C^{*\{k\}}$, to change at each iteration. The above algorithm needs therefore to be adapted accordingly. $g_C^{*\{k\}}$ is obtained by solving the subproblem given by the minimisation of $f_C^{\{k\}}(g_C^{\{k\}})$ subject to the constraints:

$$\sum_i g_i^{\{k\}} m_i(g_i^{\{k-1\}}) = 1 \quad [5.16]$$

$$\sum_i g_i^{\{k\}} E[n_i(g_i^{\{k-1\}})] = R_{PN} \quad [5.17]$$

³ The method used to create a coarse model provides an approximation which is worse than a Taylor expansion around $g^{\{k-1\}}$, and it will therefore be a good test for the convergence property of the Space Mapping algorithm.

$$g_i^{\{k\}} \geq 0 \quad [5.18]$$

The algorithm is illustrated in Figure 5.2:

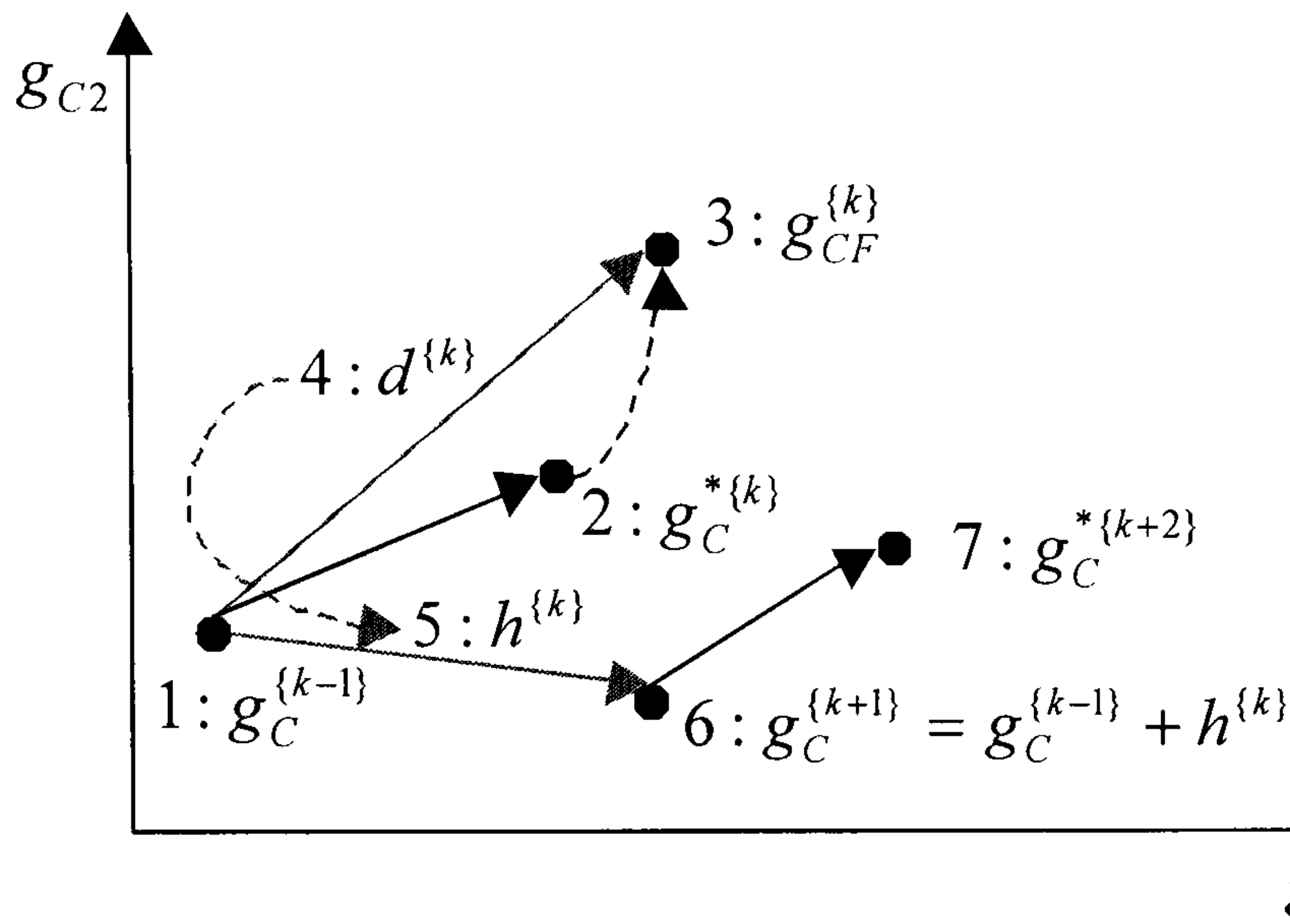


Figure 5.2: Modified Space Mapping Algorithm.

1. Select an initial point $g_C^{\{0\}}$. As a starting point the solution of the problem with fixed unit liquidity costs might be considered.
2. Calculate $g_C^{*\{k\}}$ solution of $\min_{g_C^{\{k\}}} f_C^{\{1\}}(g_C^{\{k\}}) = \min_{g_C^{\{k\}}} g_C^{\{k\}T} \Sigma_N(g_C^{\{k-1\}}) g_C^{\{k\}}$, subject to the constraints [5.16] - [5.18]. Note that the covariance matrix is kept fixed.
3. Find a new point, $g_{CF}^{\{k\}}$, such that the distance $\|f_F(g_C^{*\{k\}}) - f_C(g_{CF}^{\{k\}})\|$ is minimised.
4. Calculate the error function $d^{\{k\}} = g_{CF}^{\{k\}} - g_C^{\{k-1\}}$.
5. Use a quasi-Newtonian update to calculate $h^{\{k\}}$ solving the equations represented by either $B^{\{k\}} h^{\{k\}} = -d^{\{k\}}$ or $(B^{\{k\}T} B^{\{k\}} + \lambda) h^{\{k\}} = -B^{\{k\}T} d^{\{k\}}$.
6. Select the next point $g_C^{\{k+1\}} = g_C^{\{k\}} + h^{\{k\}}$.
7. Go to step 2.

Note that in our case the constraints are also approximated and since the new iterate, $g_C^{\{k+1\}}$, might not be feasible, the approximation around $g_C^{\{k+1\}}$ might not have a solution. This does not occur often in our tests, when it happens we use a projection of h orthogonal to the Jacobian of the approximated constraints as calculated at the previous iterate. If there is still no feasible solution, we successively halve the length of the vector $h^{\{k\}}$ until a new solution is found.

A few differences from the original model need to be highlighted. First, the optimal coarse-model solution is updated at each iteration. Second, the quasi-Newtonian iteration acts upon the coarse model variables, g_C^k , and not upon the fine-model ones. Third, the error function is composed of two parts. The error due to the update of the objective function and the constraints, $(g_C^{\{k-1\}} - g_C^{*\{k\}})$, and the error due to misalignment between the coarse and fine model at the current iterate, $(g_C^{*\{k\}} - g_{CF}^{\{k\}})$. At each iteration, we seek to reduce the sum of both errors. Note that the SM algorithm minimises the distance of the mapping, if the algorithm converges, a solution that is feasible for the coarse-model, will be feasible also for the fine-model.

5.3. Computational Results

We use the same data sets illustrated in the previous chapter. As in the previous chapter we compare the SM algorithm to the non-linear solver of Matlab. Since the algorithm requires a feasible starting point, we use the solution obtained from the previous optimisation as a new initial solution in order to find the next point on the efficient frontier. The results are illustrated the following table.

		FMINCON	SM
4 Assets LC0=0.03 LC1=0.05	Mean percentage error	0.0049%	0.0043%
	Median percentage error	0.0000%	0.0001%
	Time (s)	8.9	10.2
4 Assets LC0=0.1 LC1=0.25	Mean percentage error	0.0070%	0.0177%
	Median percentage error	0.0000%	0.0026%
	Time (s)	7.2	12.2
10 Assets LC0=0.03 LC1=0.05	Mean percentage error	0.3697%	0.0623%
	Median percentage error	0.0000%	0.0052%
	Time (s)	30.2	35.9
10 Assets LC0=0.1 LC1=0.25	Mean percentage error	1.7445%	0.0223%
	Median percentage error	0.0000%	0.0006%
	Time (s)	23.1	36.5
30 Assets LC0=0.03 LC1=0.05	Mean percentage error	0.0024%	2.0488%
	Median percentage error	0.0000%	0.2480%
	Time (s)	515.2	201.9
30 Assets LC0=0.1 LC1=0.25	Mean percentage error	2.1165%	0.0698%
	Median percentage error	0.0000%	0.0089%
	Time (s)	351.8	527.6

Table 5.1: Computational results for SM.

In our tests, the SM algorithm has an accuracy comparable to *fmincon*, even though it is marginally slower. With multiple local minima it sometimes finds a better optimum than *fmincon*. Figure 5.3 and Figure 5.4 show the solutions found by the two optimisers when high liquidity coefficients are used.

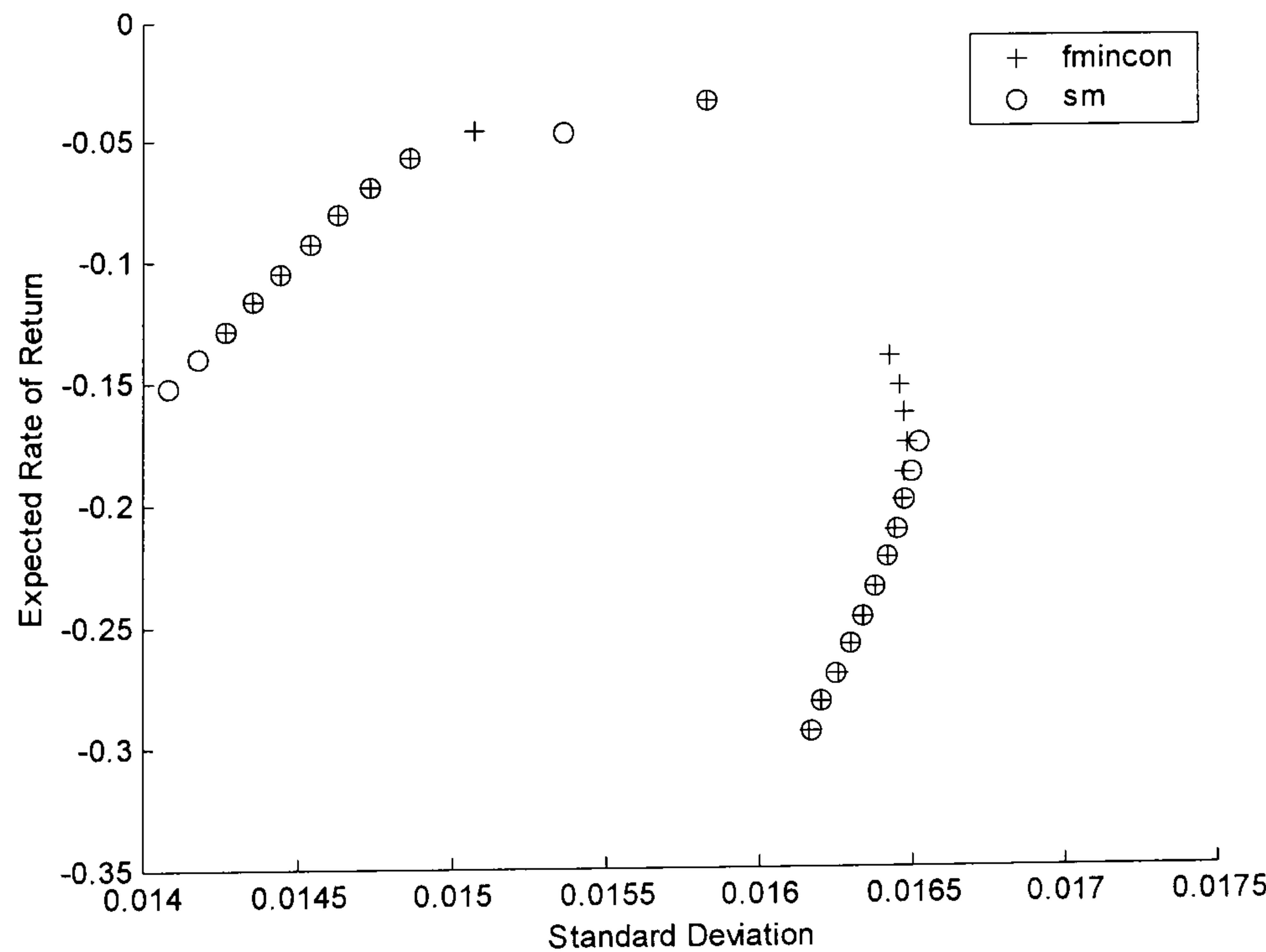


Figure 5.3: 10 Assets (LC 0.1 and 0.25).

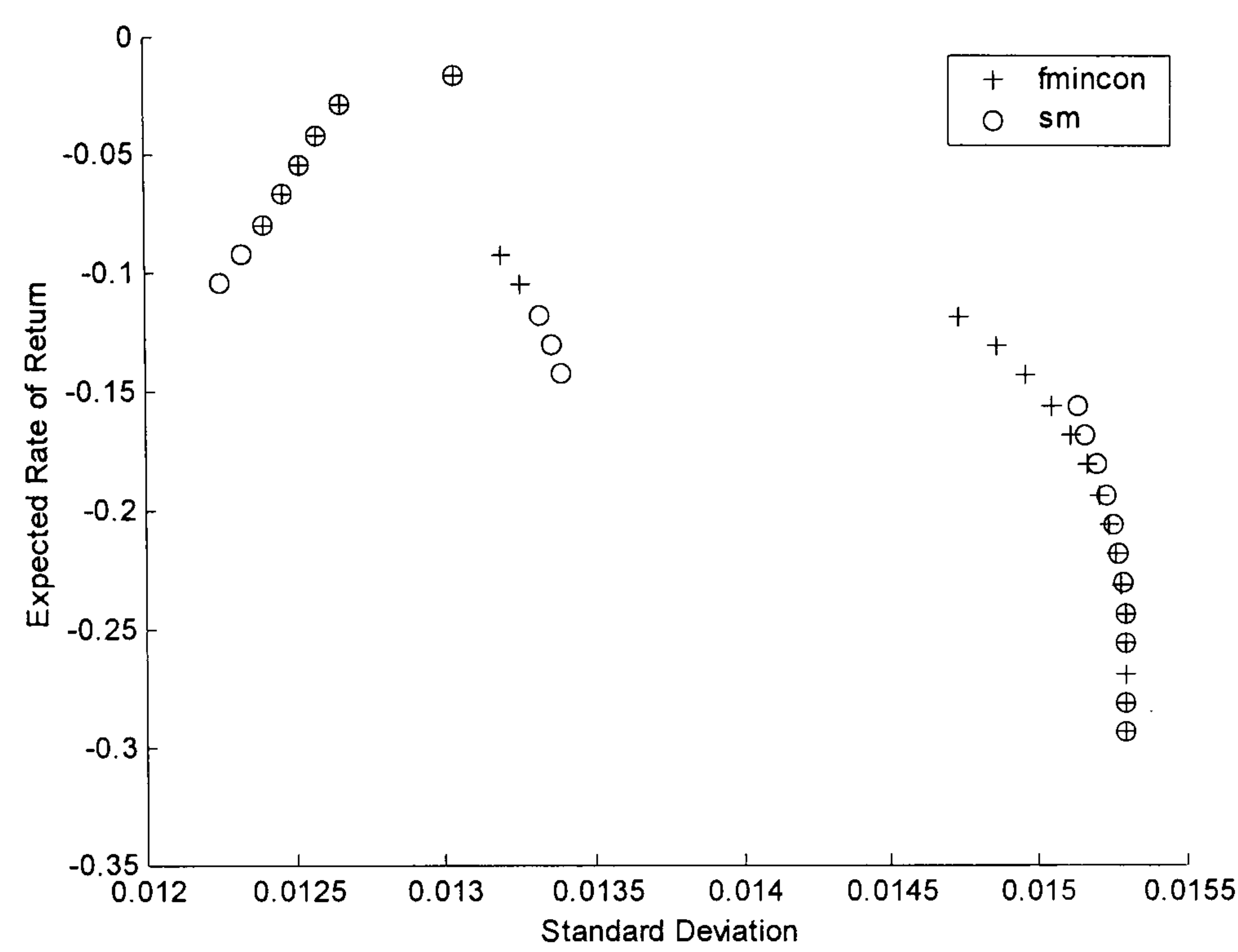


Figure 5.4: 30 Assets (LC 0.1 and 0.25).

6. Stochastic Search Heuristics for Portfolio Optimisation

While the optimisation routines illustrated previously can be characterised as gradient-based methods and involve local searches, in this chapter we present four stochastic heuristic algorithms, two based upon genetic algorithms and the others on simulated annealing, for finding the global optimal solution of the problem presented in Chapter 3. The advantage of these approaches is that they are effectively independent of the objective function adopted. The techniques above perform a global optimisation although convergence to a global optimum can be guaranteed only in a weak probabilistic sense. The algorithms were originally developed for unconstrained problems, but a variety of methods have been proposed for handling constraints, we use a special mapping from the hypercube to the tetrahedron in order to handle the budget constraint while the return constraint is incorporated within the objective function itself. In section 6.1, Genetic Algorithms are introduced. In section 6.2, we illustrate the implications of multiple local minima on the efficient frontier. In section 6.3 we propose a technique to perform a uniform mapping from a hypercube to the tetrahedron. This allows us to describe the problem domain as a hypercube avoiding possible biases. In section 6.4, we discuss an implementation of the Simulated Annealing algorithm. A variant of SA is illustrated in section 6.5. In section 6.6, we modify the GA introducing a selection process based on the Boltzmann probability distribution. The computational results are illustrated in section 6.7.

6.1. Genetic Algorithms

A genetic algorithm (GA) can be described as an "intelligent" probabilistic search algorithm. The theoretical foundations of GAs were originally developed by Holland (1975). GAs are based on the evolutionary process of biological organisms in nature. During the course of evolution, natural populations evolve according to the principles of natural selection and "survival of the fittest". Individuals which are more successful in adapting to their environment will have a better chance of surviving and reproducing, whilst individuals which are less fit will be eliminated. This means that the genes from the highly fit individuals will spread to an increasing number of individuals in each successive generation. The combination of good characteristics from highly adapted parents may produce even more fit offspring. In this way, species evolve to become increasingly better adapted to their environment.

A GA simulates these processes by taking an initial population of individuals and applying genetic operators in each reproduction. In optimisation terms, each individual in the population is encoded into a string or chromosome which represents a possible solution to a given problem. The fitness of an individual is evaluated with respect to a given objective function. The following parental selection schemes have been most commonly used in the implementation of GAs¹:

1. Proportionate reproduction: in this scheme, individuals are chosen for birth in proportion to their fitness value. Proportionate reproduction is usually implemented with a Monte Carlo or roulette wheel selection.

¹ There are a variety of other selection methods including stochastic remainder and stochastic universal selection.

2. Ranking selection. In ranking selection, the population is sorted from best to worst. The number of copies that an individual should receive is given by an assignment function, and is proportional to the rank assignment of an individual.

3. Tournament selection. In tournament selection, a random number of individuals are chosen from the population (with or without replacement) and the best individual from this group is chosen as a parent for the next generation. This process is repeated until the mating pool is filled.

Highly fit individuals or solutions are given opportunities to reproduce by exchanging pieces of their genetic information, in a crossover procedure, with other highly fit individuals. This produces new "offspring" solutions (i.e. children), which share some characteristics taken from both parents.

The ability to leave regions of attraction to local minima is, drastically enhanced by crossing over. This is an advantage if, with high probability, the crossing rules produce offspring of similar or even better fitness. In an asset allocation problem, the chromosomes represent portfolios and combinations of portfolios generally reduce the unsystematic risk, therefore improving the value of the objective function.

Mutation is often applied after crossover by altering some genes in the strings. The offspring can either replace the whole population (generational approach) or replace less fit individuals (steady-state approach). This evaluation-selection-reproduction cycle is repeated until a satisfactory solution is found.

One area of concern using GAs is premature convergence of solutions. Premature convergence refers to a situation where most of the population members have similar bit strings without reaching the optimal point in the space. The search becomes dependent on the mutation operator. One way of

preventing the loss of diversity is to not allow new strings that have a smaller Hamming distance than a specified value; however, this limits the searching of the GA. A common strategy is called incest prevention and involves a random mating of parents but only if their Hamming distance is above a certain threshold. Other techniques involve using isolated subpopulations, with occasional migrations of genomes between subpopulations, or infusing new schema into population with random generation of new strings when convergence occurs.

GAs are often described as an unconstrained optimisation method. There are a variety of constraint implementation methods but these must be used with caution. For example, if one does not allow solutions which lead to constraint violations, it may lead to evaluation of many solutions before a feasible one is found, which can add significant computational time depending upon the amount of infeasibility in the space. Another technique is to modify the solution representation itself so as not to allow the creation of infeasible solutions. Repair algorithms or decoders are special operators, which avoid the construction of illegal solutions. They may work reasonably well but are highly problem specific and may be computationally intensive to run. Additionally, they may destroy the inherent search properties of the GA or simulated annealing methods and may be difficult to implement. Penalty functions are more popular, but penalty functions combined with the objective are usually developed in an ad-hoc manner. Penalty functions may be linear, quadratic, logarithmic, etc. functions of the deviation of the constraints and/or the number of violated constraints. The performance of GAs will depend upon implementation of the constraints, as well as the choice of control parameter settings and parental selection scheme used.

The basic steps of a simple GA are shown below.

Algorithm GA

Begin

Generate an initial population

Evaluate fitness of individuals in the population

While convergence criteria is not satisfied

Recombine parents to produce children

Evaluate fitness of the children

Replace some or all of the population by the children

End

End

Arnone et al. (1993) presented a GA for the portfolio optimisation problem, but with the risk associated with the portfolio being measured by downside risk rather than by variance. Computational results were presented for one problem involving 15 assets. Loraschi et al. (1995) presented a distributed GA for the unconstrained portfolio optimisation problem based on an island model where a GA is used with multiple independent subpopulations (each run on a different processor) and highly-fit individuals occasionally migrate between the subpopulations. Computational results were presented for one problem involving 53 assets comparing their distributed GA with the GA presented in Arnone (1993).

In our GA heuristic the chromosome representation of a solution is a vector of portfolio weights, which is consistent with the budget constraints [6.2]-[6.3]. The constraint on the expected return is incorporated into the objective function using a weighting parameter λ , between 0 and 1, which can be interpreted as the risk aversion parameter. By varying λ , the efficient frontier is obtained². The problem then becomes:

$$\underset{g}{\text{Min}} \lambda \text{VAR} \left[\sum_i w_i n_i(w_i) \right] - (1 - \lambda) \sum_i (w_i n_i(w_i)) \quad [6.1]$$

s.t.

² Since the presence of liquidity constraints increases the steepness of the efficient frontier, for λ close to 1, small variations in the parameters involve big changes in the efficient frontier. In choosing λ we therefore use an adaptive algorithm that oversamples those regions where the change in the objective function is greatest.

$$\sum_i g_i = 1 \quad [6.2]$$

$$w_i = \frac{g_i}{(1 + s_{0,i}(g_i))} \quad [6.3]$$

$$g_i \geq 0 \quad [6.4]$$

Setting

$$\frac{R_A}{2} = \frac{\lambda}{1 - \lambda} \quad [6.5]$$

Where R_A is the Arrow-Pratt absolute risk-aversion index³. We use a steady-state strategy with a population size of N . Parents are chosen by binary tournament selection which works by forming two pools of individuals, each consisting of two individuals drawn from the population randomly. The individuals with the best fitness, each taken from one of the two tournament pools, are chosen to be parents. We can weaken selection by choosing the fitter member with a probability $(1+\epsilon)/2$, where ϵ is a random number in the interval $[0,1]$.

Tournament selection is particularly popular when used with a steady-state GA, where one member of the population is changed at a time. The major difference between steady-state and generational GAs is that for each P members of the population generated in the generational GA there are $2P$ selections. Consequently the selection strength and genetic drift⁴ for a steady-state GA is twice that of the generational GA. The steady-state GA, therefore,

³ The Arrow-Pratt absolute risk aversion index characterises the utility function. It is defined as $R_A = -U''/U'$. For quadratic utility functions or normally distributed returns the expression of R_A is given by [6.5], where R_A measures the trade-off between the expected return and the risk measure.

⁴ Genetic drift is the random change in allele frequency from one generation to the next due to chance fluctuations (as opposed to due to selection pressure).

appears twice as fast although it can lose out in the long term because it does not explore the landscape as well as the generational GA.

Binary tournament selection can be regarded as a means of performing ranking selection with roulette wheel sampling: n -tournament selection is equivalent to a ranking selection, as in Baker (1985), where the individuals are selected with a probability proportional to r^{n-1} , where r is the rank.

Children in our GA heuristic are generated by uniform crossover. In uniform crossover two parents have a single child. Each of the assets will have a weight randomly chosen from one of the parents. Uniform crossover causes very rapid shuffling of the variables in a population, although it may cause considerable disruption. Nevertheless in our problem the site position does not have any intrinsic meaning and therefore disruption is not a major shortcoming.⁵

Children are also subject to mutation, with probability P_M , by multiplying by 0.9 or 1.1, chosen with equal probability, the value of a randomly selected asset k . After crossover and mutation have taken place, the weights are finally scaled proportionally in order to satisfy the constraints [6.2]-[6.3]. The procedure is illustrated below, assuming that the weight of asset k is mutated upwards.

$$1. \quad g_i^{\{C\}} = \gamma g_i^{\{F\}} + (1 - \gamma) g_i^{\{M\}} \quad \gamma \in \{0,1\} \quad [6.6]$$

$$2. \quad g_k^{\{C\}} = 1.1 g_k^{\{C\}} \quad [6.7]$$

$$3. \quad g_i^{\{C\}} = \frac{g_i^{\{C\}}}{\sum_j g_j^{\{C\}}} \quad \forall i \quad [6.8]$$

⁵ As an alternative, a simple crossover technique might require that the weights of the child are an average of the weights of the parents. In this case, in [6.6], γ will be either a constant or a random variable in the range $[0,1]$, instead

$$4. \quad w_i^{\{C\}} = \frac{g_i^{\{C\}}}{m_i(g_i^{\{C\}})} \quad \forall i \quad [6.9]$$

Where $g^{\{C\}}$ are the weights of the child and $g^{\{F\}}$ and $g^{\{M\}}$ are the weights of the parents, γ is a binomial random variable that can take value 0 or 1. Equation [6.6] and [6.7] represent crossover and mutation respectively, while equations [6.8] and [6.9] rescale the weights and ensure that the budget constraint is satisfied. The gross weights are used in the chromosomes in order to avoid that assets with lower transaction costs are over-represented in the population⁶. We use a steady-state population replacement strategy. With this strategy each child replaces the member with the worst objective function.

6.2. Invisible Sections of the Efficient Frontier

In the approach proposed the budget constraint is dealt with directly in the formulation of the solution, while the constraint on the expected net return is handled indirectly via the objective function. This method is motivated by the fact that the budget constraint is a hard constraint while the trade-off between risk and return is an intrinsic characteristic of the model. Dividing the objective function in [6.1] by $(1-\lambda)$ and using [6.5], the objective function can be rewritten as the equation of a straight line in the mean-variance space.

$$E[W_1] = \frac{R_A}{2} VAR[W_1] + c \quad [6.10]$$

a binomial random variable. The convergence of the population is accelerated but the hamming distance is systematically reduced at each iteration. This in turn might lock the algorithm into a local minimum.

⁶ In order to illustrate the problem, consider a two-asset portfolio with a constant spread of 10% on the first asset and of 0% on the second one. Assume that two identical parents with equal gross weights are mated using $\gamma=0.5$ and no mutation. The offspring should be identical to their parents. Now assume that the net weights, w , are used in the chromosomes. Following the procedure illustrated in [6.6]-[6.9] and using w instead of g , the resulting offspring will have gross weights of 47.6% and 52.4%, thus introducing a bias in the population towards the asset with lower transaction costs.

Where c , the intercept, is a linear transformation of the original objective function⁷. Maximising the objective function for any given value of R_A , corresponds to maximising the intercept for a specified gradient. Over the feasible region, the intercept is maximised, uniquely, at the point of tangency with the upper-left border of the region. Systematically varying R_A changes the point of tangency and traces the entire efficient frontier.

As we have seen in section 3.3, in some extreme cases, the presence of multiple local minima can produce a discontinuous efficient frontier. In such cases the point of tangency jumps to the lower curve maintaining smoothness in the increasing gradient but missing out some efficient points. The problem is illustrated in Figure 6.1.

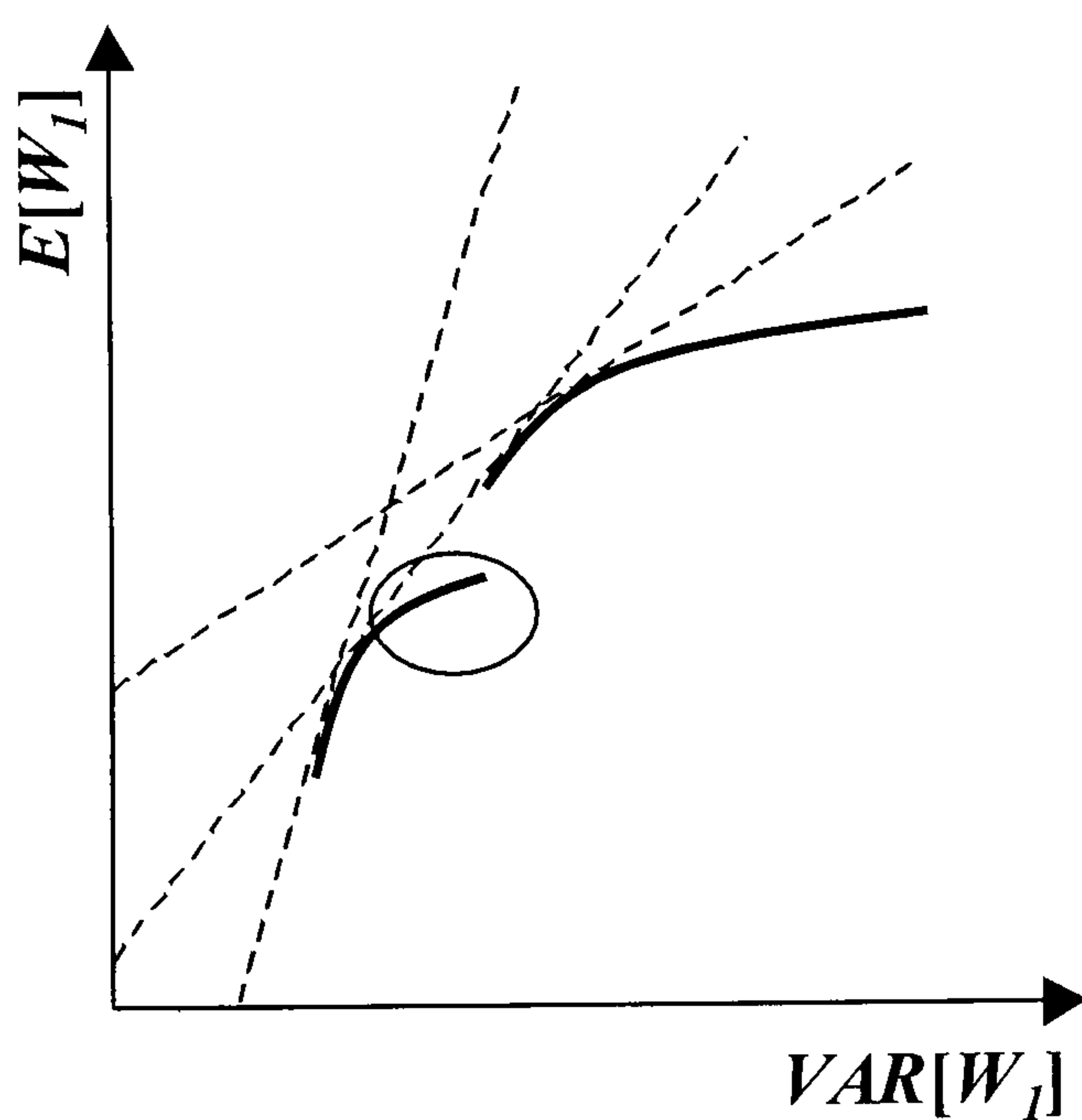


Figure 6.1: Invisible section of the efficient frontier.

Nevertheless the heuristic approach involves investigating many different solutions and it is therefore possible to gain information about such portions. We therefore store all the improved solutions found during the course of each

⁷ We have $c = -f/(1-\lambda)$, where f denotes the value of the original objective function in [6.1]. Minimising f for a given λ is therefore equivalent to maximising c for a given R_A , with R_A calculated using [6.5].

of our heuristics. This set will contain a large number of dominated solutions, however it is a simple matter to extract the subset of efficient solutions.

6.3. Low Discrepancy Sequences and Marginal Probability Distributions

To address the problem illustrated in the previous section the set of efficient solutions will be the union of the sets of undominated improved solutions found by solving the optimisation problem for each value of λ . It is therefore important to initialise the population in each iteration in such a way that the problem space can be examined exhaustively. Low discrepancy series are a natural choice. The deterministic points returned by such sequences are often used in place of random numbers to obtain faster convergence when solving simulation problems. In fact, choosing m points uniformly randomly in an n -dimensional space leads to an error term in Monte Carlo integration that decreases as $1/\sqrt{m}$. Using a Cartesian grid and sampling each point exactly once (in whatever order) the error decreases as m^{-1} . The problem with the grid is that one has to decide in advance how fine it should be. An intermediate scheme is given by quasi-random sequences also known as low discrepancy sequences. Intuitively the points in a quasi-random sequence are uniformly distributed and new points progressively fill the gaps. More formally discrepancy is a measure of non-uniformity of a sequence of points placed in the unit hypercube $[0,1]^n$. The most widely studied distance measure is the star discrepancy. Considering a set of m points in $[0,1]^n$, star discrepancy is calculated using the following function.

$$D_m^*(x_1, \dots, x_m) = \sup_{0 \leq v_j < 1, j=1, \dots, n} \left| \frac{1}{m} \sum_i \left(\prod_{j=1}^n I_{x_{ij} \in [0, v_j)} \right) - \prod_{j=1}^n v_j \right| \quad [6.11]$$

Where I is an indicator function with value of 1 if the j^{th} element of $x_i \in [0, v_j)$, and 0 otherwise. For every subset E of $[0,1]^n$ of the form $[0, v_1) \times \dots \times [0, v_n)$, we divide the number of points in E by m and take the absolute difference between this quotient and the volume of E . The highest such value is taken as the measure of star discrepancy. A sequence of m points in $[0,1]^n$ is a low-discrepancy sequence if for any $m > 1$

$$D_m^*(x_1, \dots, x_m) \leq c(n) \frac{(\log m)^n}{m} \quad \forall m > 1 \quad [6.12]$$

Where the constant $c(n)$ depends only on the problem dimension n . The idea behind the low-discrepancy sequences is to let the fraction of the points within any subset E of $[0,1]^n$ of the form $[0, v_1) \times \dots \times [0, v_n)$ be as close as possible to its volume. That way, the low discrepancy sequences will spread over $[0,1]^n$ as uniformly as possible, reducing gaps and clustering of points. The basic low discrepancy sequences proposed in the literature are those of Halton (1960), Sobol (1967), and Faure (1982). In what follows we will use a modified version of the Sobol sequence as in Press et al. (1997).

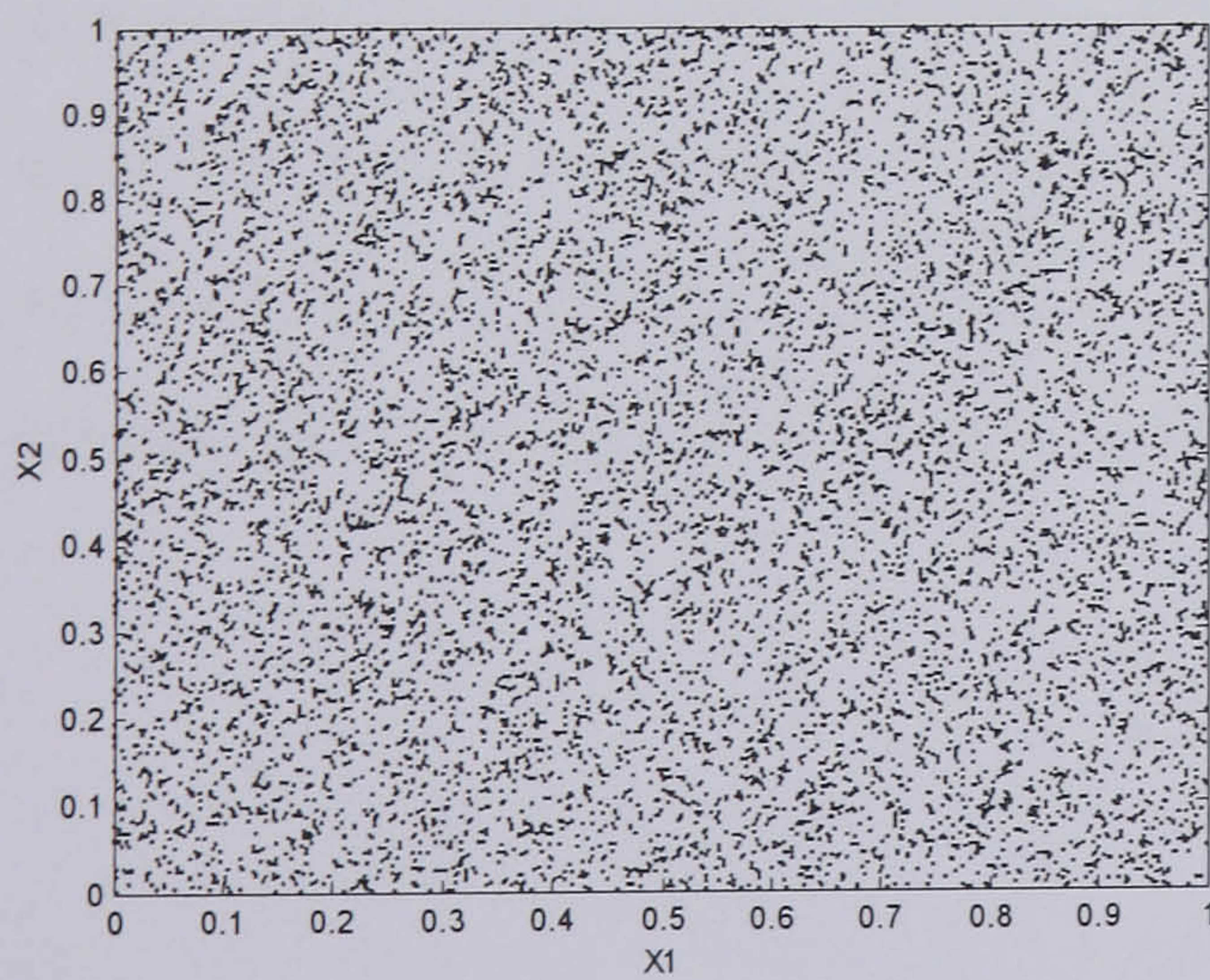


Figure 6.2: Uniform deviates.

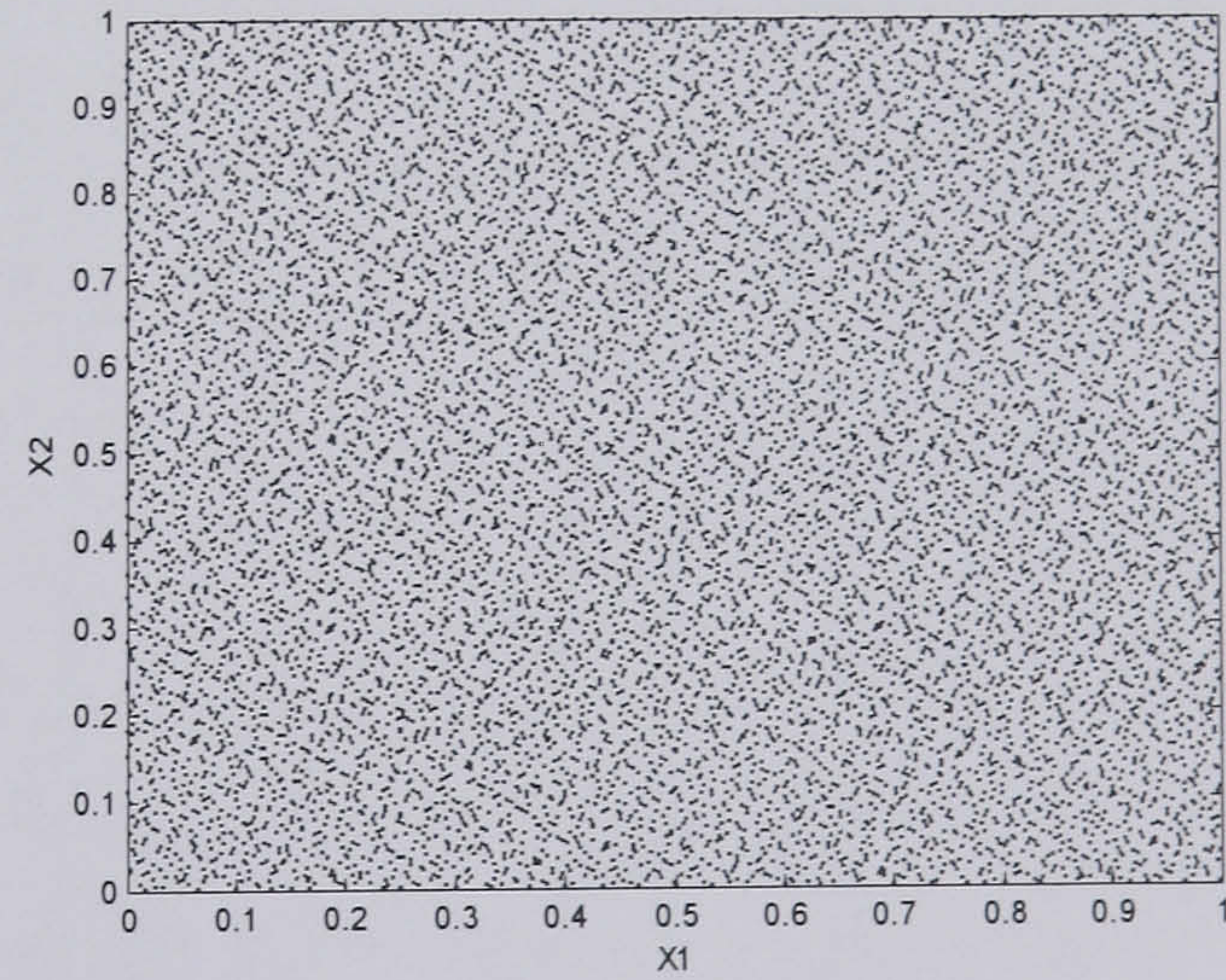


Figure 6.3: Sobol sequence.

As we have seen, low discrepancy sequences return points in the hypercube $[0,1]^n$, but equations [6.2] and [6.4] imply that a solution to the problem is a

point on a tetrahedron in $\mathbb{R}^n$ ⁸. Therefore, when new solutions are generated in the population initialisation phase, and during the course of the algorithm, by crossover and mutation, we need to ensure that the candidate solutions are points of the tetrahedron. Unfortunately, simple non-invertible mapping methods do not preserve the uniform covering property. For instance, the mappings $(i, j) \rightarrow (i/(i+j), j/(i+j))$ or $(i, j) \rightarrow ((i-j)/2+0.5, (j-i)/2+0.5)$ are such that the resultants sum to one but will have more points mapping to the region close to (0.5,0.5). Approaches based on a projection from the hypercube to the tetrahedron suffer from the opposite problem and overweight the boundaries of the domain. A simple solution might involve discarding those points that violate the constraints, but the volume of the tetrahedron drops almost exponentially while the volume of the unit cube is constant. Therefore the number of solutions to be discarded grows exponentially with the dimension of the problem. Also folding the hypercube into the tetrahedron using successive cuts becomes prohibitive as the dimensionality grows.

We follow Stratford (1996) and Bayliffe (1998) and use an iterative procedure to generate the coordinates directly on the tetrahedron creating a mapping from the hypercube of dimension $n-1$. We first calculate the marginal distribution of g_1 , and using an initial variable, u_1 , in the range $[0,1]$, we calculate g_1 . We then use the marginal distribution of g_2 given g_1 and a second variable, u_2 , in order to obtain g_2 , and so on for the other coordinates. The procedure is illustrated in Figure 6.4 for $n=3$.

⁸ $\mathbb{R}^n$ denotes the Euclidean space in which the tetrahedron is embedded, the dimension of the tetrahedron itself is $n-1$.

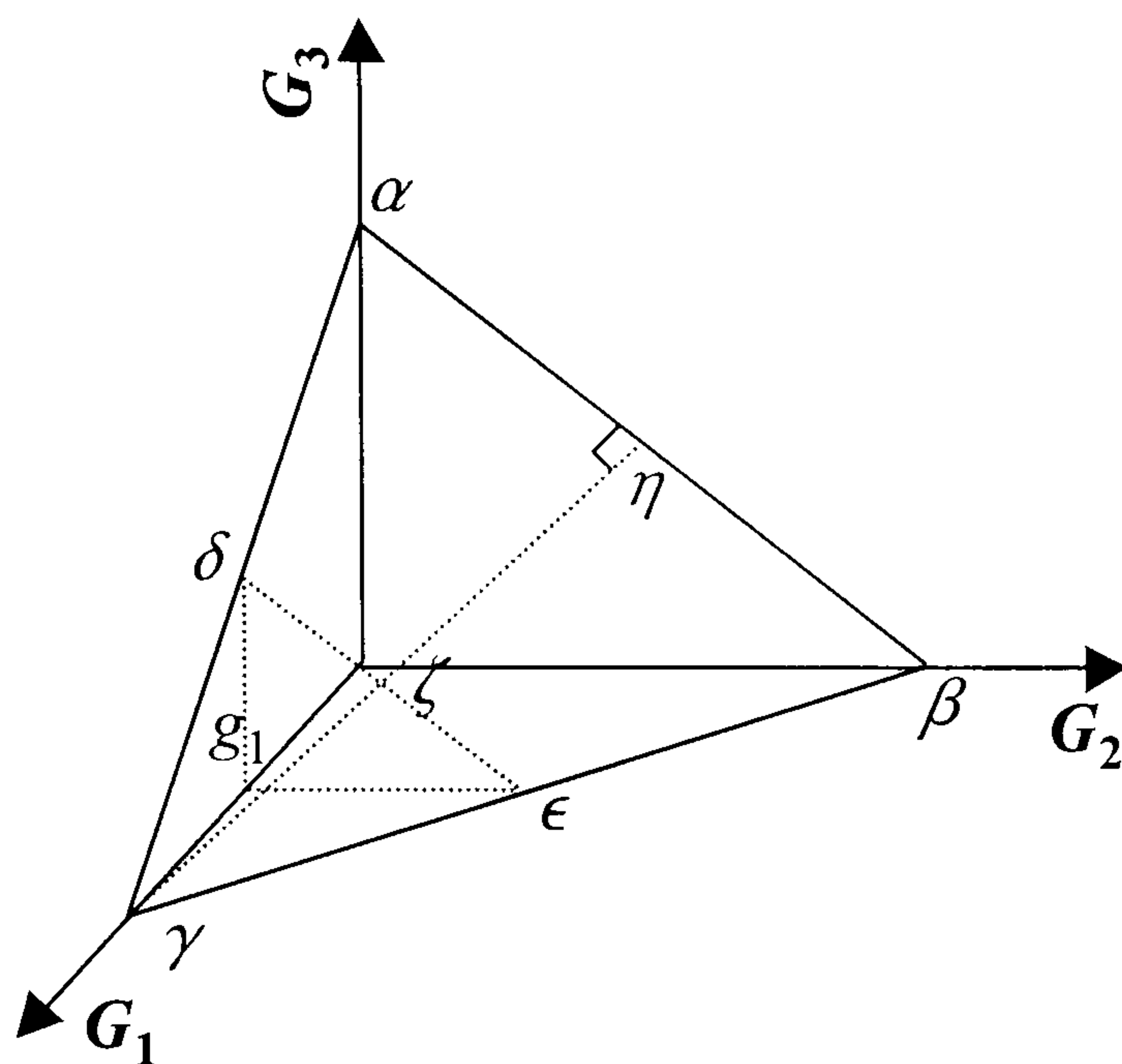


Figure 6.4: Mapping coordinates on the tetrahedron.

The marginal distribution for g_1 , $P(G_1 < g_1)$ is given by the ratio of the area $\alpha\beta\epsilon\delta$ to the area $\alpha\beta\gamma$. Since by construction:

$$\frac{|\delta\epsilon|}{|\alpha\beta|} = \frac{|\gamma\zeta|}{|\gamma\eta|} = \frac{|\gamma\epsilon|}{|\gamma\beta|} = \frac{1 - g_1}{1} \quad [6.13]$$

We have that:

$$|\delta\epsilon\gamma| = (1 - g_1)^2 |\alpha\beta\gamma| \quad [6.14]$$

And therefore,

$$P(G_1 < g_1) = 1 - (1 - g_1)^2 \quad [6.15]$$

$P(G_2 < g_2 | G_1 = g_1)$ must be a point on the segment $\delta\epsilon$, i.e. the uniform distribution on $[0, 1 - g_1]$ so that $P(G_2 < g_2 | G_1 = g_1) = g_2 / (1 - g_1)$, and $g_3 = 1 - g_1 - g_2$. In the n dimensional case, we have $P(G_1 < g_1) = 1 - (1 - g_1)^{n-1}$, generating g_2 is equivalent to generating the other coordinates in $n-1$ dimensions, where the

vertices have a distance of $1-g_1$ from the origin. The required expression is given by:

$$P(G_2 < g_2 | G_1 = g_1) = 1 - \frac{(1 - g_1 - g_2)^{n-2}}{(1 - g_1)^{n-2}} \quad [6.16]$$

For the other coordinates we have:

$$P(G_i < g_i | G_1 = g_1, \dots, G_{i-1} = g_{i-1}) = 1 - \frac{\left(1 - \sum_{j=1}^i g_j\right)^{n-i}}{\left(1 - \sum_{j=1}^{i-1} g_j\right)^{n-i}} \quad i = 1, \dots, n-1 \quad [6.17]$$

$$g_n = 1 - \sum_{j=1}^{n-1} g_j \quad i = n \quad [6.18]$$

By the fundamental transformation law of probabilities, given a transformation of a random deviate u such that $g=f(u)$ the probability distribution of g is determined by:

$$p(g) = p(u) \left| \frac{du}{dg} \right| \quad [6.19]$$

If u is uniformly distributed in $[0,1]$ we have that:

$$F(g) = \int p(g) dg = \int \left| \frac{du}{dg} \right| dg = u \quad [6.20]$$

Inverting [6.20] and using [6.17] and [6.18] we obtain:

$$g_i = M(u) = \begin{cases} \left(1 - \sum_{j=1}^{i-1} g_j\right) \left(1 - (1 - u_i)^{1/(n-i)}\right) & i = 1, \dots, n-1 \\ 1 - \sum_{j=1}^{n-1} g_j & i = n \end{cases} \quad [6.21]$$

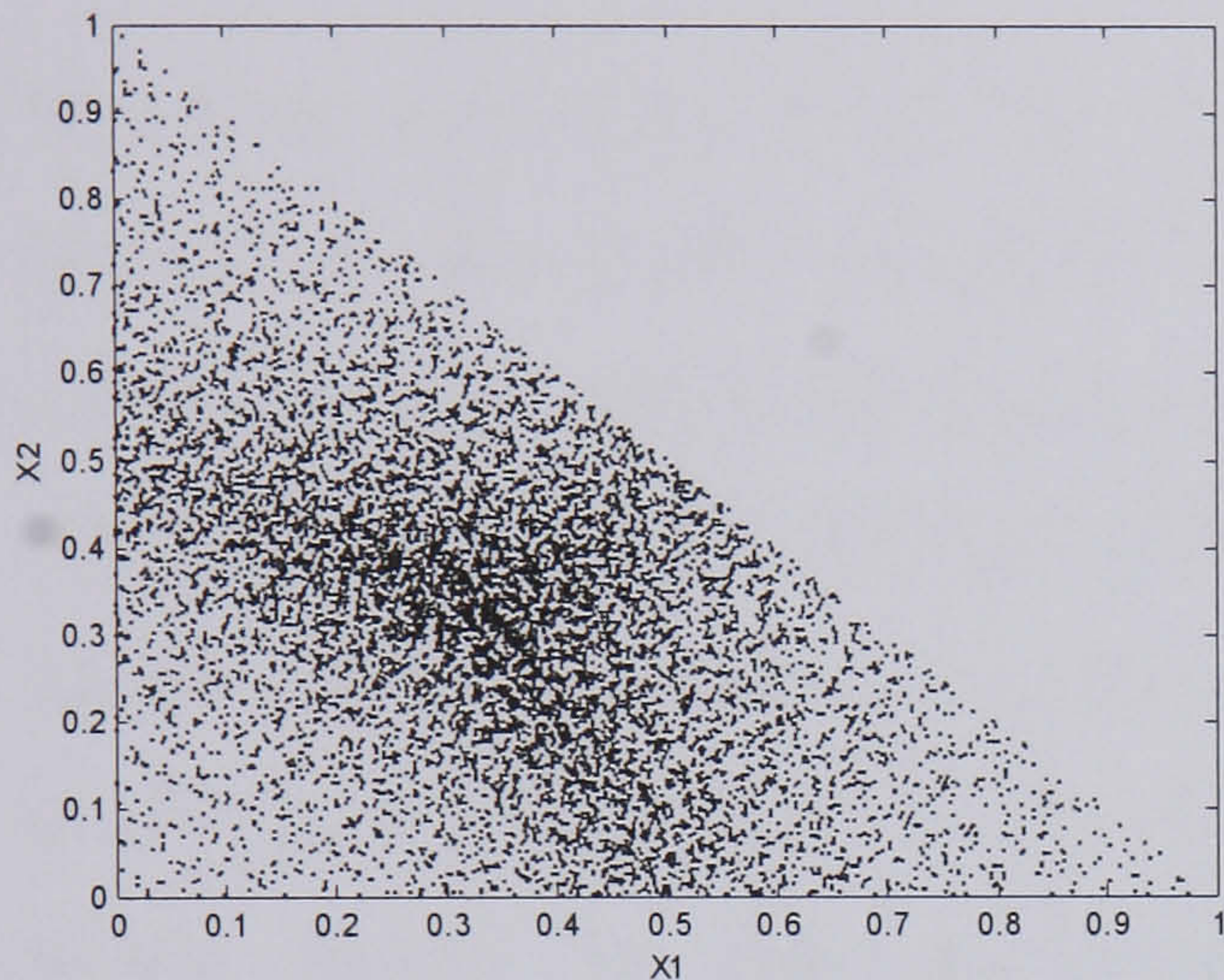


Figure 6.5: Non-invertible mapping.

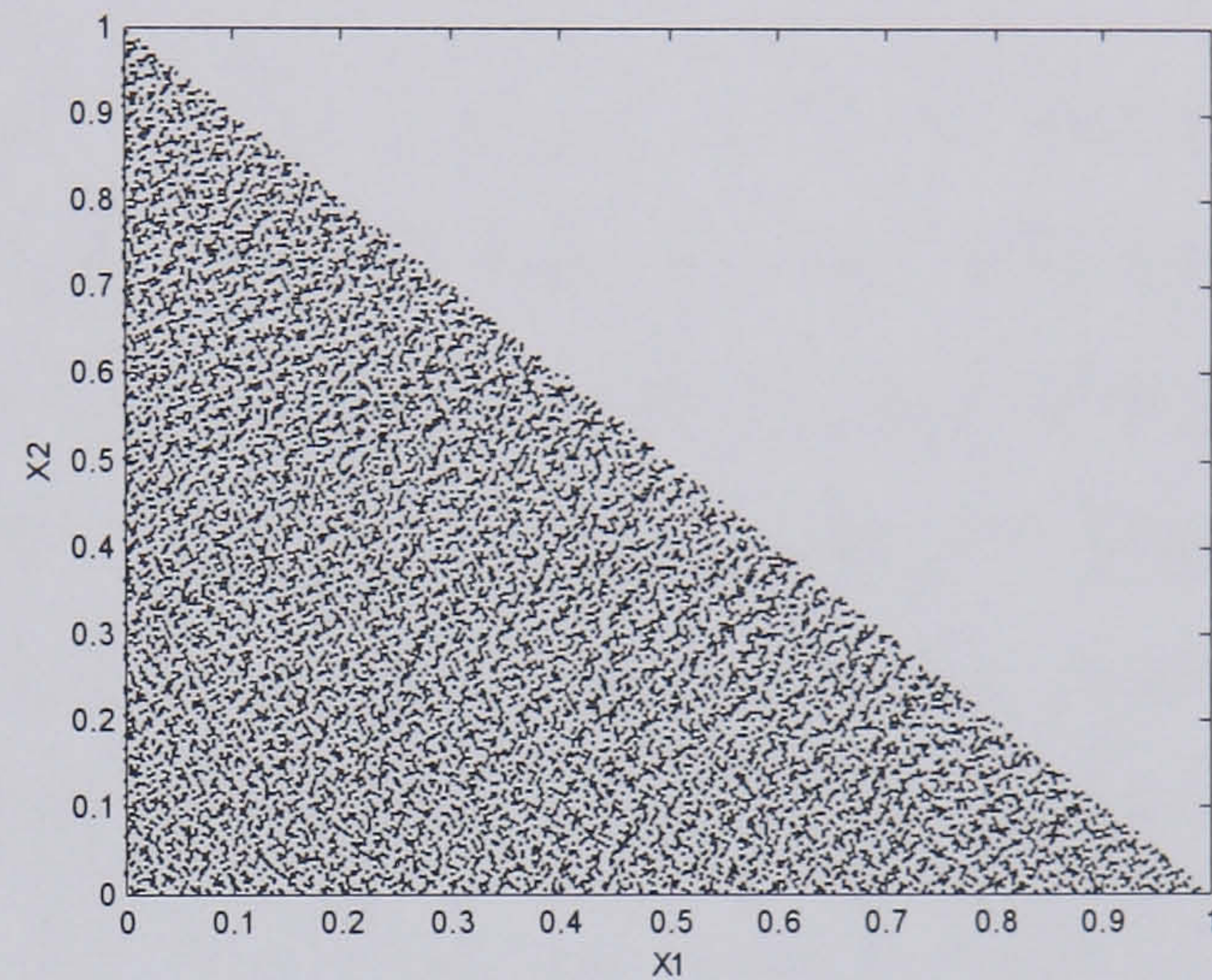


Figure 6.6: Uniform mapping.

Figure 6.5 shows the mapping of the first 10,000 points of a Sobol sequence for a 3-asset portfolio using [6.8]. The mapping of the same sequence⁹ using [6.21] is shown in Figure 6.6. The bias introduced by the non-invertible mapping is evident.

We will therefore need to modify the process illustrated in section 6.2 by substituting [6.8] with [6.21]. Each chromosome will represent a point in the unit hypercube of dimension $n-1$, which, using [6.21], is equivalent to a portfolio of n assets.

This approach transforms the points of the low discrepancy sequence according to a uniform distribution on the tetrahedron, but, since the sum of the coordinates is 1, the average value of a coordinate is $1/n$, and, as n increases the probability of getting a high coordinate falls off exponentially.

When generating our initial population we will therefore add randomly some feasible solutions at the boundary of the domain.

6.4. Simulated Annealing

Simulated Annealing takes its intuition from the fact that the heating (annealing) and slow cooling of a metal brings it into a more uniformly crystalline state, that is believed to be the state where the free energy of bulk matter takes its global minimum. The concept is based on the manner in which liquids freeze or metals recrystallise in the process of annealing. In an annealing process a melt, initially at high temperature, T , and disordered, is slowly cooled so that the system at any time is approximately in thermodynamic equilibrium. As cooling proceeds, the system becomes more ordered and approaches a "frozen" ground state at $T=0$. The role of temperature is to allow the configurations to reach higher energy states with a probability given by Boltzmann's exponential law, so that they can overcome energy barriers that would otherwise force them into local minima. If the initial temperature of the system is too low or cooling is done insufficiently slowly the system may become quenched forming defects or freezing out in metastable states (i.e. trapped in a local minimum energy state). This is quite unlike line search methods and trust region methods on which good local optimisation programs are based. The phenomenon was first described by Metropolis et al. (1953), with an equation that simulates the evolution of a physical system in contact with a heat-bath as it is observed at random times.

Its use for optimisation problems originated with Kirkpatrick et al. (1983) and Cerny (1985). The analogy of the SA to a minimisation problem is

⁹ Note that when using [6.21] only two inputs are required for each point in the three-dimensional tetrahedron.

straightforward. The current state of the thermodynamic system is analogous to the current solution to the problem, the energy equation for the thermodynamic system is analogous to the objective function, and ground state is analogous to the global minimum. Instead of optimising the objective function, $f(w)$ directly, SA samples an associated probability density $p(w, T)$

$$p(w, T) = \exp\left(-\frac{f(w)}{T}\right) \quad [6.22]$$

Which for a low value of T is sharply peaked at the w that minimises $f(w)$. Since the exponential function is monotonic, it does not change the problem, but it can be sampled efficiently using the Metropolis algorithm illustrated below. The resulting sequence of states is guaranteed to asymptotically sample the density function. When T is large SA explores the entire parameter space using a uniformly generated random walk, when T is small SA becomes a descent algorithm. The search is systematically concentrated into regions likely to contain a global minimum, but it is still random enough to escape most local minima¹⁰.

The major difficulty in the implementation of the algorithm is that there is no obvious analogy for the temperature T with respect to a free parameter in the problem. Furthermore, avoidance of entrapment in local minima (quenching) is dependent on the annealing schedule, the choice of initial temperature, how many iterations are performed at each temperature, and how much the temperature is decreased at each step as cooling proceeds.

¹⁰ SA is effective when deep minima have wide rims. SA fails when the global minimum occurs in a "gopher hole", which is too steep and narrow to be discovered by sampling at higher temperatures.

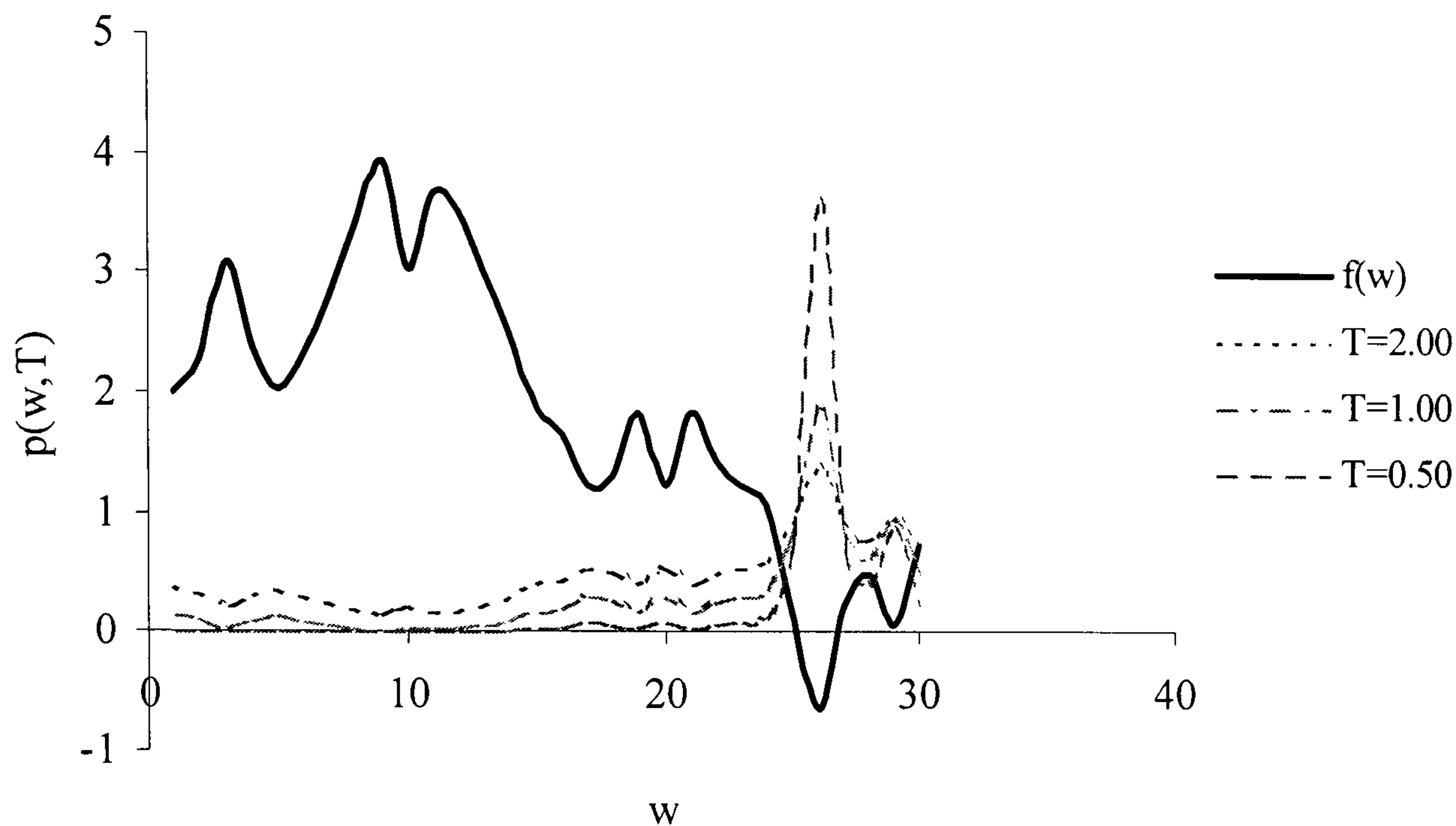


Figure 6.7: Objective function and associated density function $p(w, T)$ for different values of T .

In operation, the Metropolis algorithm assumes an “energy” function $f(w)$ is defined on a state w . We generate a new state y in a neighbourhood of an old state x randomly, and accept the new state if:

$$f(y) < f(x) \quad [6.23]$$

Otherwise we accept y if $P > R$, where R is a random variable uniformly distributed between 0 and 1 and P has the following form:

$$P = \frac{p(y, T^{\{k\}})}{p(x, T^{\{k\}})} = \exp\left(-\frac{f(y) - f(x)}{T^{\{k\}}}\right) \quad [6.24]$$

The expression is based on a heuristic derived from the Boltzmann probability distribution of thermodynamics.

This process is then repeated sufficient times to give good sampling statistics for the current temperature, and then the temperature is decreased, according to an annealing schedule that generates a sequence $T^{\{k\}}$ that converges to zero. The entire process is repeated until a frozen state is achieved at $T=0$. At zero temperature, there are no random solutions generated.

SA therefore incorporates a statistical component, in that, moves to worse solutions are accepted with a specified probability that decreases over the course of the algorithm. The higher the temperature T , the higher the probability of accepting the move. Hence this probability decreases as the temperature decreases. Considering Figure 6.7, the algorithm can be interpreted as a smart randomised descent method. When T is high we are moving on a function $p(w, T)$ which is almost flat and the random component prevails. As T decreases $p(w, T)$ becomes more peaked, near the global minimum, solutions with higher values in terms of $f(w)$ and lower values in terms of $p(w, T)$ are severely penalised and are unlikely to be selected. The algorithm therefore becomes a descent method.

For the portfolio problem, the simulated annealing states correspond to feasible portfolios. The function f is just the objective function in [6.1]. At each iteration, the algorithm randomly selects a set of asset weights to change, and generates a random feasible solution¹². Feasibility is obtained by a process similar to the one described for the GA.

$$1. \quad x_i^{\{k\}} = x_i^{\{k-1\}} (1 + \varepsilon_i) \quad \forall i \quad [6.25]$$

¹¹ As T decreases $p(w, T)$ becomes more peaked in the region of the global minimum. The function $f(w)$ does not represent the actual objective function but is a generic function with multiple local minima chosen for illustrative purposes.

¹² In continuous problems SA becomes less reliable as the number of dimensions increase. A better algorithm in this case is illustrated in the following section.

$$2. \quad g_i^{\{k\}} = M(x_i^{\{k\}}) \quad \forall i \quad [6.26]$$

$$3. \quad w_i^{\{k\}} = \frac{g_i^{\{k\}}}{m_i(g_i^{\{k\}})} \quad \forall i \quad [6.27]$$

Where ε_i is an appropriate random variable¹³, and M represents the mapping from the hypercube to the tetrahedron described in the previous section. As in the GA, the budget constraint [6.3] is embedded within the routine that generates new solutions while the constraint on the expected return is accounted for in the objective function [6.1].

The initial fixed temperatures are updated by the annealing schedule $T^{\{k\}} = \gamma T^{\{k-1\}}$, where γ is a real number in the open interval (0,1). After each iteration, new solutions are evaluated and accepted with respect to the Boltzmann probability distribution. The pseudocode of the algorithm is illustrated below.

Algorithm SA

Begin

Initialise T

Choose a starting point x

While $T >> 0$

Loop N times

 Select y in the neighbourhood of x

If $f(y) < f(x)$ then

$x = y$

Else

 Evaluate $P = \exp(-(f(y) - f(x))/T)$

 Extract R , uniform random variable in $[0,1]$

If $P > R$ then $x = y$

End

End

$T = \gamma T$

End

End

¹³ In our implementation we used a uniform random variable in $[-0.02, +0.02]$.

6.5. Tree Annealing

A variant of simulated annealing, called tree annealing, was developed by Bilbro and Snyder (1991) to globally minimise continuous functions. In tree annealing, candidate subspaces are accepted or rejected by evaluating the function at representative points. The tree annealing algorithm is very similar to simulated annealing: given a point x associated with the current search interval and a point y associated with a candidate search interval, y replaces x depending upon $f(x)$, $f(y)$, and the progress of the algorithm. The algorithm gradually narrows the width of the subspace to close in around the global minimum.

Tree annealing stores information in a binary tree to keep track of which subintervals have been explored. Each node in the tree represents one of two subintervals defined by the parent node. Initially the tree consists of one parent and two child nodes. As better subintervals are found, the path down the tree that leads to these intervals gets deeper and the nodes along this path define smaller and smaller subspaces. The tree annealing algorithm is as follows:

1. Randomly choose an initial point x over the search interval I_0 .
2. Randomly travel down the tree to an arbitrary terminal node i , and generate a candidate point y over the subspace defined by I_i extracting y from a uniform distribution over I_i . The node i is selected in the following way. At any non-terminal node, the probability of going left or right is $n_L/(n_L + n_R)$ or $n_R/(n_L + n_R)$ where n_L is the number of subnodes to the left and n_R is the number of subnodes to the right. To generate a candidate y , start at the root node and descend the tree making left/right decisions based on the probabilities of going left or right at a given node until a terminal node, i , is reached. Then, generate a point uniformly over the subspace, I_i , defined by the terminal node. Descending the tree implies that the size of

the subspaces is decreasing. If a path is continually travelled, that implies that the algorithm is converging on the terminal node in that path.

3. If $f(y) < f(x)$ replace x with y , and go to step 5.
4. Compute P and if $P > R$, where R is a random number uniformly distributed between 0 and 1, replace x with y . P is a modification of the Metropolis criterion.

$$P = \frac{p(y, T)g(x)}{p(x, T)g(y)} = \exp\left(-\frac{f(y) - f(x)}{T}\right) \frac{g(x)}{g(y)} \quad [6.28]$$

Where $g(x)$ and $g(y)$ represent the numerical probabilities of generating x and y . More specifically $g(x)$ is the probability of reaching the node associated with x calculated using n_L and n_R at each non-terminal node along the path from the root to the leaf node. This probability is then divided by the volume of the subspace associated with the node. The ratio $g(y)/g(x)$ is the bias toward y relative to x and intuitively makes y less likely to be accepted if it is more likely to be regenerated.

5. If y is accepted, turn the terminal node containing y into a non-terminal node with two children¹⁴. In our implementation the largest dimension in the relevant interval is divided into two, thus generating two subintervals of equal volume. At this point, the number of right and left nodes at each node will have changed and the tree will need to be updated.
6. Decrease T slightly. If $T \approx 0$ exit otherwise go to step 2.

This algorithm is very similar to simulated annealing, however, although tree annealing compares the function evaluation at two points x and y , these points are samples of subspaces, where the new candidate y is generated directly from the root rather than directly from the current state x . Also, acceptance of

y involves information on the entire tree rather than on y , x and T alone. In fact, while SA performs the optimisation indirectly by using the density function $p(w, T)$ and perturbing the current state, TA generates candidates from a global approximation of $p(w, T)$ given by $g(w)$. Information on $g(w)$ is maintained in a binary tree which is used both to record the density of previous samples and to generate candidates consistent with the recorded density. In the Metropolis procedure candidates are sampled from a uniform distribution and these samples are converted to samples of $p(w, T)$ by accepting each with a probability [6.24]. In TA, $g(w)$ is not uniform but samples from it are converted to samples from the true $p(w, T)$ by accepting each with probability [6.28].

As we have seen previously, in our case the domain is represented by a tetrahedron and each division will generate two new tetrahedrons. As a result the complexity of the computations required increase geometrically with the dimensionality of the problem. On the other hand, non-invertible mappings from subspaces defined on a hypercube might violate the bounds of the intervals. This problem highlights another use for the mapping illustrated in section 6.3, by allowing us to work directly in a hypercube without violating the bounds of the subspaces.

6.6. Genetic Algorithms and Boltzmann Distribution

It is possible to modify the selection process in a GA using the Boltzmann probability distribution, so that, given two candidates, x and y , and their respective objective function values, $f(x)$ and $f(y)$, y is accepted if $P > R$, where R is a random variable uniformly distributed between 0 and 1 and P is

¹⁴ This is a simplification of the actual algorithm, where, before the interval is subdivided, several samples are tested and their distribution is stored.

$$P = \begin{cases} 1 & f(y) < f(x) \\ \exp\left(-\frac{f(y) - f(x)}{T^{(k)}}\right) & f(y) \geq f(x) \end{cases} \quad [6.29]$$

Boltzmann selection has at least two practical benefits. Firstly it has a single control parameter t which extrapolates from low selective pressure (t small) to high selection pressure (t large). Secondly it typically produces a larger ratio of gain in the fitness to loss of variance in the fitness than other selection procedures. Boltzmann tournament selection was first introduced by Goldberg (1990). In our implementation we modify slightly the algorithm illustrated in section 6.1, so that each parent is selected from a pool of three individuals extracted from the population. One individual will compete with another candidate from the same pool, and its probability of being selected is proportional to the Boltzmann probability function. The winner will then compete with the third individual. After a given number of iterations the temperature is lowered and that in turn increases the selection strength.

6.7. Computational Results

Since the points on the efficient frontier are obtained by varying λ we need to use a different error measure. To analyse the experimental results we use the metric suggested in Chang et al. (2000). In order to measure the accuracy of the models proposed, the deviations of the points on the efficient frontier obtained are measured as the minimum absolute distance (vertical or horizontal) from an interpolated benchmark efficient frontier. In our case the horizontal distance is represented by the difference in the standard deviations and the vertical distance by the difference in the expected rate of returns. The benchmark efficient frontier is the set of all undominated improved solutions found by all the algorithms considered. The errors provide us only with a

relative accuracy measure, useful to compare different routines, but it is reasonable to expect that the benchmark will be close to the exact efficient frontier. Each heuristics evaluates $1000 \cdot n$ solutions. Each efficient frontier is calculated using 200 points (by varying λ , and therefore R_A).

It is clear from Table 6.1 that our Genetic Algorithm with Boltzmann tournament selection (GASA) is the most accurate heuristic, with a mean percentage error of 0.023%, the GA heuristic is the next best with a mean error of 0.041%, Simulated Annealing has an average error of 0.468% and the worst heuristic in our tests is Tree Annealing, with a mean percentage error of 4.229%. For the Simulated Annealing algorithm the mean is always significantly greater than the median, which suggests that the distribution of the errors is skewed.

		GA	SA	TA	GASA
4 Assets	Mean percentage error	0.003%	0.002%	0.099%	0.003%
LC0=0.03	Median percentage error	0.000%	0.000%	0.009%	0.000%
LC1=0.05	Time (s)	94.1	24.8	45.8	115.6
4 Assets	Mean percentage error	0.011%	0.051%	0.086%	0.005%
LC0=0.1	Median percentage error	0.000%	0.000%	0.020%	0.000%
LC1=0.25	Time (s)	93.6	24.7	45.4	114.8
10 Assets	Mean percentage error	0.045%	0.127%	5.268%	0.018%
LC0=0.03	Median percentage error	0.011%	0.000%	3.296%	0.000%
LC1=0.05	Time (s)	233.0	59.5	92.9	288.0
10 Assets	Mean percentage error	0.039%	0.539%	3.040%	0.021%
LC0=0.1	Median percentage error	0.023%	0.000%	2.210%	0.000%
LC1=0.25	Time (s)	230.0	58.4	94.6	283.6
30 Assets	Mean percentage error	0.038%	1.170%	8.795%	0.012%
LC0=0.03	Median percentage error	0.000%	0.452%	8.124%	0.000%
LC1=0.05	Time (s)	758.5	202.9	257.0	894.3
30 Assets	Mean percentage error	0.107%	0.916%	8.084%	0.080%
LC0=0.1	Median percentage error	0.118%	0.275%	8.001%	0.089%
LC1=0.25	Time (s)	728.0	201.9	259.5	887.2

Table 6.1: Computational results for search heuristics.

In our tests, GA and GASA are slower than the other heuristics because they use more heavily the Matlab random number generator, which is not very efficient. Calls to the random number generator explain up to 60% of the

difference in execution time. An additional 20% is accounted for by the selection of the worse candidate. At each iteration we loop through all the population members, but the process could be probably made faster by initially ranking the members of the population according to their fitness and inserting new children respecting the sort order, so that the last member is always the one with the worse fitness.

The poor accuracy of the TA heuristic is explained by the fact that the Tree Annealing algorithm tries to find an interval in which a sub-global optimum is located, rather than find the optimum point itself, and a local search would be beneficial. Our findings regarding the relative accuracy of the GA algorithm with respect to the SA heuristic are in line with the results obtained by Chang et al. (2000). It is likely that the SA algorithm requires more fine-tuning when selecting the parameters, particularly regarding the annealing schedule and the generation of new candidates.

As expected the difference between GA and GASA was only marginal, nevertheless the introduction of the Boltzmann selection process was beneficial in our tests. With GASA we decided to select the parents from two sets of three members each (in order to further reduce the selection strength), but the higher number of calls to the random number generator made the algorithm slightly slower.

Using the set of improved solutions from the heuristics it is possible to gain more information on the shape of the exact efficient frontier. As expected it appears that, due to the multiple local minima in the objective function, the efficient frontier is discontinuous for high liquidity coefficients. Figure 6.8 shows the frontier described by the set of undominated improved solutions for ten-asset portfolios with liquidity coefficients of 10% and 25%; Figure 6.9- Figure 6.12 show the results of the four heuristics for the same problem set.

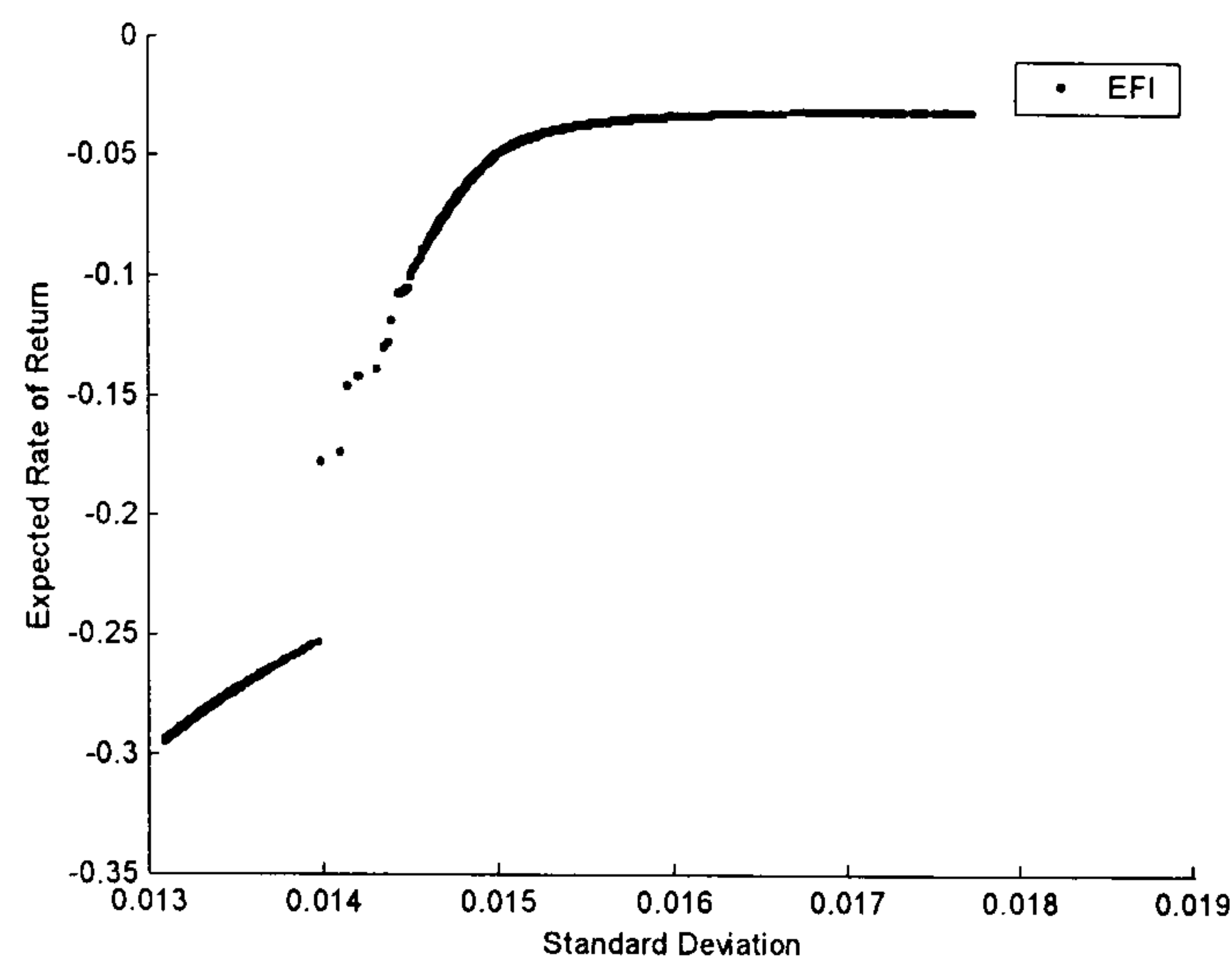


Figure 6.8: Set of undominated improved solutions.

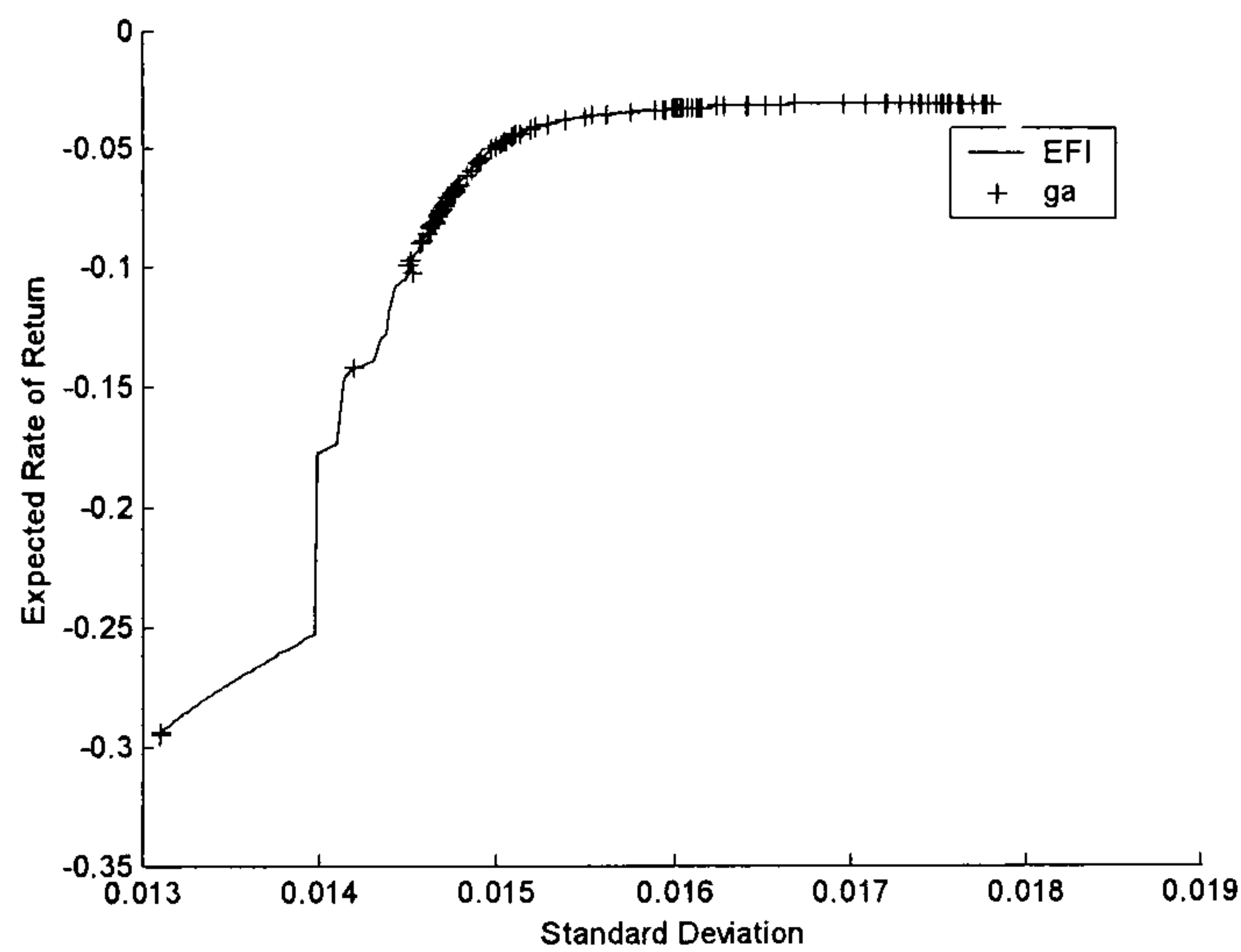


Figure 6.9: Genetic Algorithm.

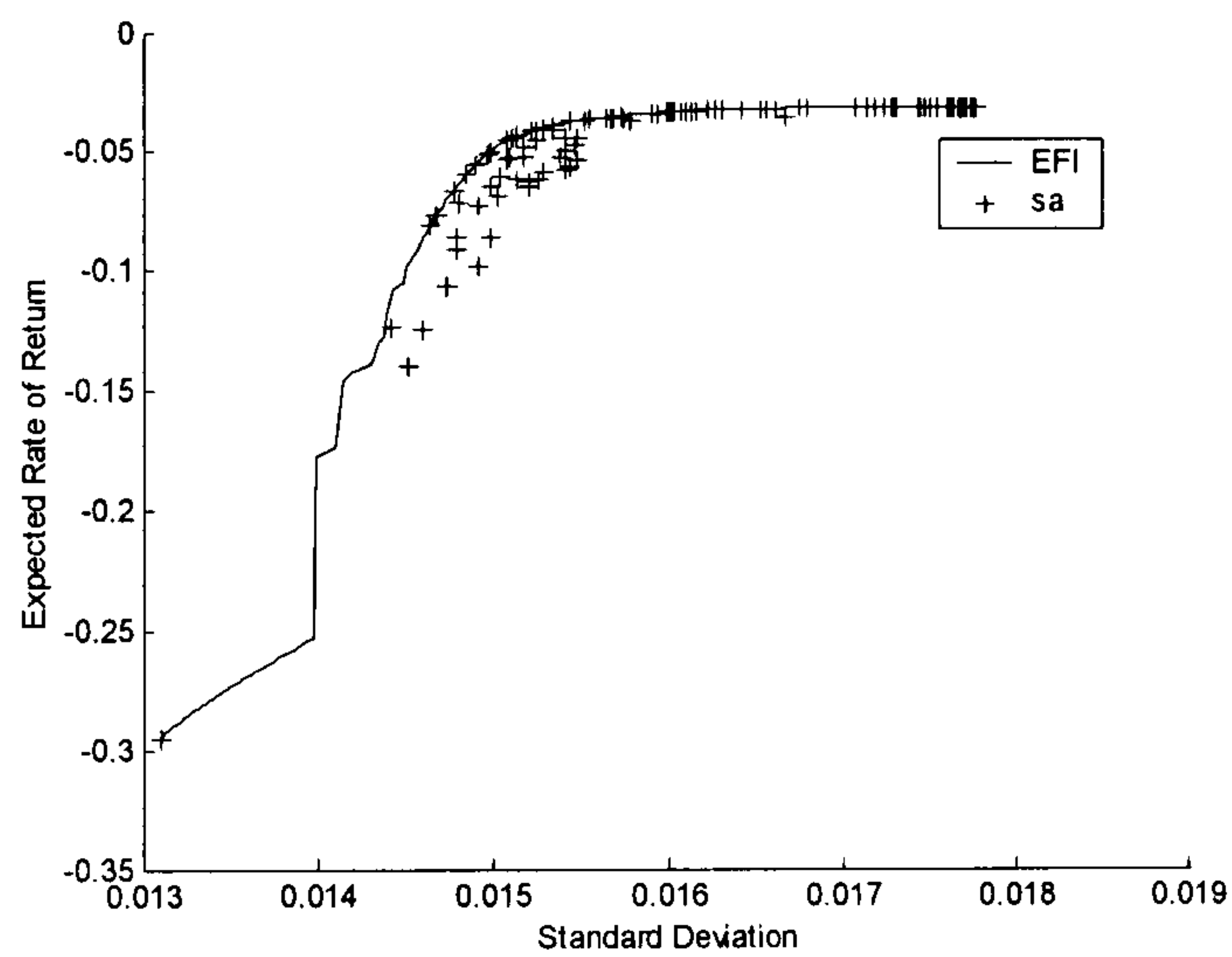


Figure 6.10: Simulated Annealing.

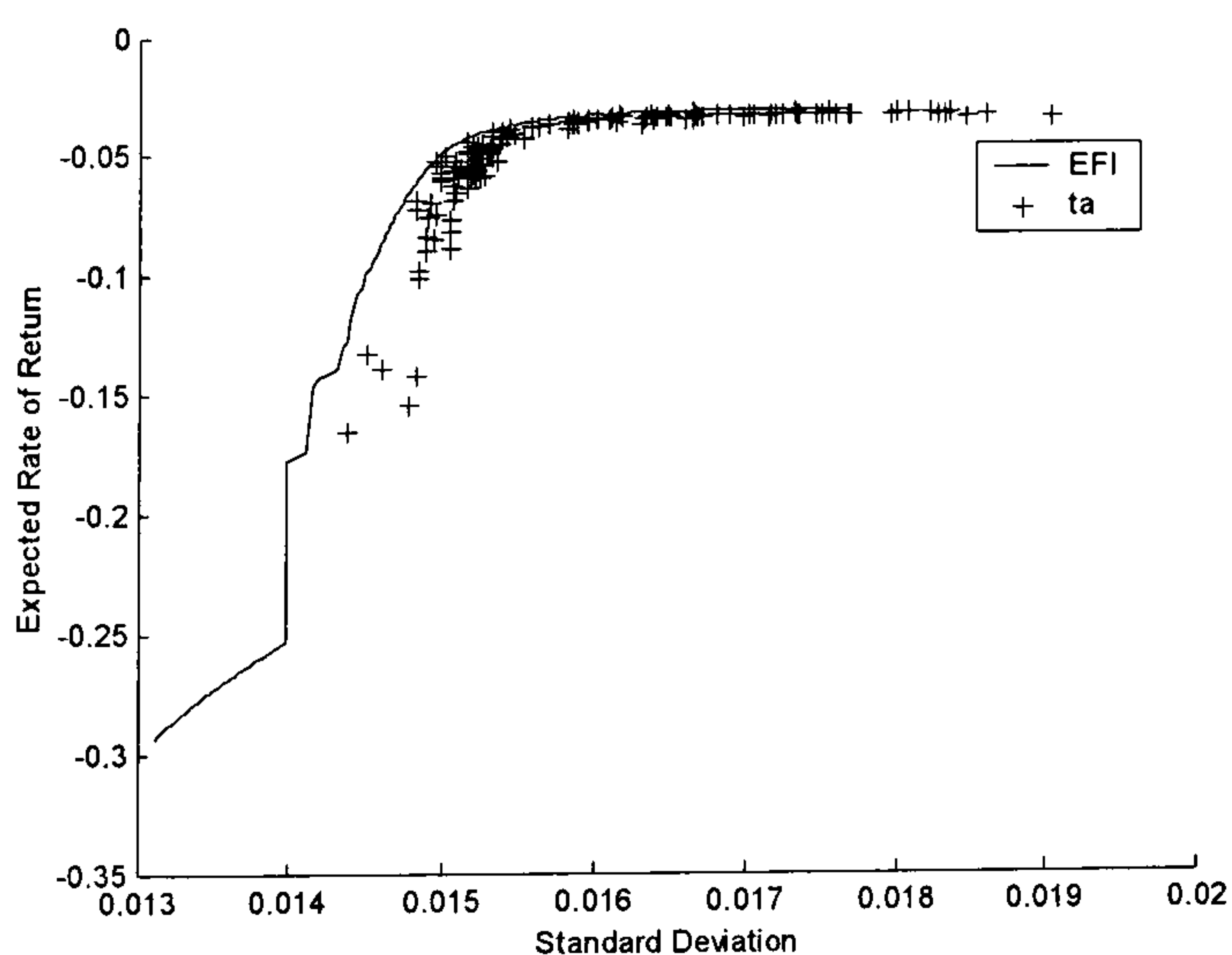


Figure 6.11: Tree Annealing.

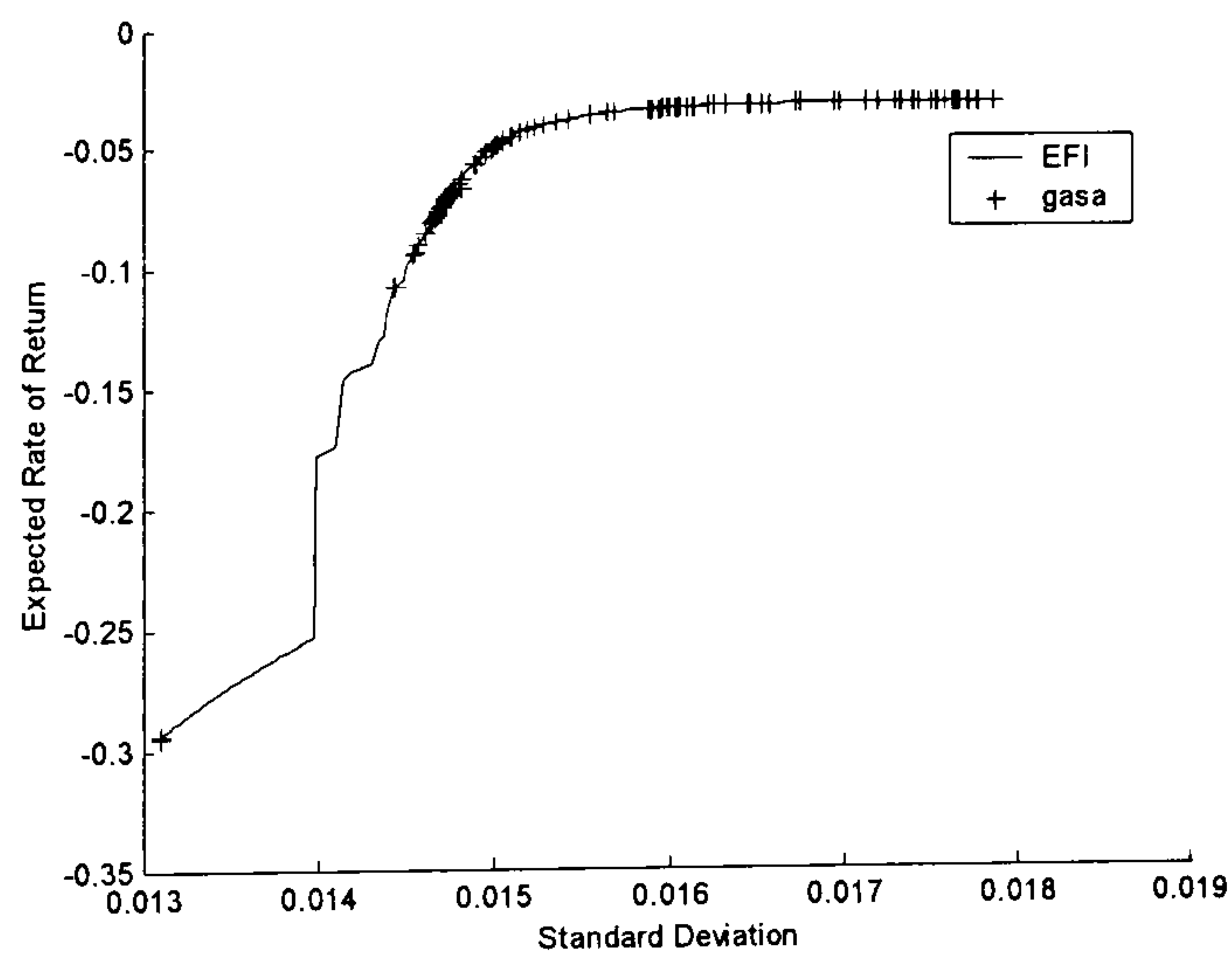


Figure 6.12: GA with Boltzmann tournament selection.

7. Conclusions

In this thesis we have modelled an asset allocation problem under liquidity constraints. The formulation proposed is capable of capturing different dimensions of the liquidity structure. Both the tightness and the depth of the market can in fact be included in the framework.

Differently from existing literature, the effects of uncertainties on the future liquidity spread were considered from a single investor perspective and also within a general equilibrium framework. In the latter case a modification of the traditional CAPM was presented. In the modified CAPM obtained, the adjusted systematic risk measure incorporates the risk associated with the uncertainty about future liquidity market conditions. The modified beta is non-linear in the traditional beta. Furthermore the model implies a convex relation between the expected future transaction spread and the expected return. When the modified CAPM is used within the Black and Litterman model, an estimator of the net expected excess returns is obtained.

We then analysed the effects of more complex liquidity cost functions on asset allocation. We proposed a heuristic algorithm that can find approximate solutions when fixed transaction costs are included. When the spread depends on the transaction size, the model proposed involves the minimisation of a non-convex objective function subject to non-linear constraints. This difficulty was addressed using two different approaches, either generating a sequence of approximations of the original problem, or using stochastic search heuristics.

In the first case, a sequence of subproblems was obtained using either Sequential Convex Programming or the Space Mapping Algorithm. Using SCP, the resulting subproblems are convex and separable. They can therefore

be linearised and modelled as generalised network flow problems with parallel arcs. A specialised version of the generalised network simplex algorithm, which exploits the contiguity of the solution and other structural properties of the netform, was applied to solve the subproblems. We achieved an improvement in the performance of the optimiser, as measured by the average time, of 5 to 10 times. On the other hand the subproblems generated by SMA are quadratic and can be solved using available optimisers. In our tests, the algorithm showed to be comparable in accuracy to the non-linear optimiser from Matlab.

In the second case, we proposed global optimisation techniques based on Genetic Algorithms, Simulated Annealing and Tree Annealing. A uniform mapping from a hypercube to the tetrahedron was used, so that the domain of the problem could be represented by a hypercube. That has the advantage of avoiding the bias implied by other techniques and at the same time allows us to use optimisers that would otherwise be impractical. We found the Genetic Algorithm and its modified version, with Boltzmann tournament selection, to be the more accurate heuristics. Storing improved solutions during the course of each algorithm allowed us to explore sections of the efficient frontier that would otherwise have been invisible, given our formulation of the objective function and the fact that the efficient frontier presents discrete jumps.

It is worth noting that the routines outlined in this thesis do not depend on the particular form of the objective function and other risk measures can easily be incorporated.

7.1. Future Research

There are several possible directions for future research. It would be interesting to examine the problem presented within a multi-period framework. The problem could be formulated as a non-linear stochastic network flow problem (similar to the one proposed by Dembo et al. in 1989), with the relative spread modelled via parallel arcs and using scenarios to describe the uncertainty regarding future transaction costs.

From market microstructure research it appears that the level of illiquidity is correlated to the volatility of the asset, because of the higher inventory costs faced by the market makers. Models should try to capture this feature.

There are also alternative ways of incorporating the effects of liquidity in the portfolio selection process. An illiquid market might be modelled considering the timing impact, i.e. imposing a restriction on the maximum amount that can be traded at a given time. Also the liquidity premium might be expressed as the relative spread between substitutable assets (e.g. on-the-run vs. off-the-run government bonds) as opposed to the spread for individual assets.

Finally, value at risk (VAR) models could be adjusted in order to properly consider the random nature of the liquidity premium; the high correlation of such premia during liquidity crises; and the risk associated with the necessity of raising cash in short periods of time (e.g. because of margin calls) which might force investors to transact large amounts, increasing the average transaction costs, and therefore the VAR.

Bibliography

- [1] Adock C.J., and Meade N., (1994). "A Simple Algorithm to Incorporate Transaction Costs in Quadratic Optimisation". *European Journal of Operational Research*, 79, pp. 85-94.
- [2] Ahlfed D.P., Mulvey J.M., Dembo R.S., and Zenios S.A., (1987). "Nonlinear Programming on Generalized Networks". *ACM Transactions on Mathematical Software*, 13(4), pp. 350-367.
- [3] Ahuja R.A., Magnanti T.L., and Orlin J.J., (1993). "Network Flows, Theory Algorithms, and Applications". Prentice-Hall.
- [4] Amihud Y., and Mendelson H., (1986). "Asset Pricing and the Bid-Ask Spread". *Journal of Financial Economics*, 15, pp. 223-249.
- [5] Amihud Y., and Mendelson H., (1989). "The Effects of Beta, Bid-Ask Spread, Residual Risk and Size on Stock Returns". *Journal of Finance*, 38, pp. 479-486.
- [6] Arnone S., Loraschi A., Tettamanzi A., (1993). "A Genetic Approach to Portfolio Selection". *Neural Network World*, pp. 597-604.
- [7] Avriel M., (1976). "Nonlinear Programming: Analysis and Methods". Prentice-Hall.
- [8] Baker. J.E., (1985). "Adaptive Selection Methods for Genetic Algorithms". *Proceedings of an International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates (Hillsdale, NJ), 24, pp. 101-111.
- [9] Bandler J.W., Bakr M.H., Biernacki R.M., Chen S.H., and Madsen K., (1998). "A Trust Region Aggressive Space Mapping Algorithm for EM Optimization". *IEEE Transactions on Microwave Theory and Techniques*, Vol. 46, pp. 2412-2425.
- [10] Bandler J.W., Bakr M.H., Georgevia N., and Madesen K., (1999). "A Hybrid Aggressive Space Mapping Algorithm for EM Optimization".

Bibliography

- IEEE Transactions on Microwave Theory and Techniques, Vol. 47, pp. 2440-2449.
- [11] Bandler J.W., Biernacki R.M., Chen S.H., Grobelny P.A., and Hemmers R.H., (1994). "Space Mapping Technique for Electromagnetic Optimization". IEEE Transactions on Microwave Theory and Techniques, Vol. 42, pp. 2536-2544.
- [12] Bangia A., Diebold F., Shuerman T., Stroughair J., (1999). "Liquidity on the Outside". Risk, Vol. 12, pp. 68-73.
- [13] Banz R.W., (1981). "The Relationship between Return and Market Value of Common Stocks". Journal of Financial Economics, 9, pp. 3-18.
- [14] Barthelemy J.F., and Haftka R., (1993). "Approximation Concepts for Optimum Structural Design: a Review". Structural Optimization, 5, pp. 129-144.
- [15] Bassi F., Embrechts P., and Kafetzaki M., (1998). "Risk Management and Quantile Estimation". In "A Practical Guide to Heavy Tails". Birkhäuser, pp.111-130.
- [16] Bayliffe D., (1998). "Analytical Models in Sovereign Liability Management". PhD Thesis, Centre for Quantitative Finance, Imperial College, London.
- [17] Berstimas D., Lo A.W., (1998). "Optimal Control of Execution Costs". Journal of Financial Markets, 1, pp. 1-50.
- [18] Bertsekas D.P., and Tseng P., (1985). "Relaxation Methods for Minimum Cost Ordinary and Generalized Network Flow Problems". Operations Research, 36, pp. 93-114.
- [19] Bertsekas D.P., and Tseng P., (1994). "Relax-IV: A Faster Version of the Relax Code for Solving Minimum Cost Flow Problems". LIDS-P-2276.

Bibliography

- [20] Bertsekas D.P., and Tseng P., (2000). "An Epsilon-Relaxation Method for Separable Convex Cost Generalized Network Flow Problems". *Mathematical Programming*, 88, pp. 85-104.
- [21] Bertsekas D.P., Polymenakos L.C., and Tseng P., (1997). "An Epsilon-Relaxation Method for Convex Network Optimization Problems". *SIAM J. on Optimization*, Vol. 7, pp. 853-870.
- [22] Bilbro G.L., Snyder W.E., (1991). "Optimization of Functions with Many Minima". *IEEE Transactions on Systems, Man, and Cybernetics*, 21, pp. 840-849.
- [23] Black F.S., (1974). "International Capital Market Equilibrium with Investment Barriers". *Journal of Financial Economics*, 1, pp. 337-352.
- [24] Black F.S., and Litterman R., (1992). "Global Portfolio Optimization". *Financial Analysts Journal* 48(5), pp. 28-43.
- [25] Black F.S., Jensen M.C., Scholes M., (1972). "The Capital Asset Pricing Model: Some Empirical Tests". *Studies in the Theory of Capital Markets*, Praeger, pp. 79-121.
- [26] Brennan M.J., and Subrahmanyam A., (1996). "Market Microstructure and Asset Pricing: on the Compensation for Liquidity in Stock Returns". *Journal of Financial Economics*, 41, pp. 441-464.
- [27] Brennan M.J., Chordia T., and Subrahmanyam A., (1998). "Alternative Factor Specifications, Security Characteristics, and the Cross-Section of Expected Stock Returns". *Journal of Financial Economics*, 49, pp. 79-121.
- [28] Brown G.G., and McBride R.D., (1984). "Solving Generalized Networks". *Management Science*, 30(12), pp. 1497-1523.
- [29] Campbell J.Y., Lo A.W., MacKinlay A.C., (1997). "The Econometrics of Financial Markets". Princeton University Press.
- [30] Cerny V., (1985). "Thermodynamical Approach to the Travelling Salesman Problem: an Efficient Simulation Algorithm". *Journal of Optimization Theory and Applications*, 45, pp. 41-51.

Bibliography

- [31] Chang T.J., Meade N., Beasley J.E., and Sharaiha Y.M.. (2000). "Heuristics for Cardinality Constrained Portfolio Optimisation". *Computer and Operations Research*, 27 (13), pp. 1271-1302.
- [32] Chickermane H., (1996). "A New Local Approximation Method for Structural Optimization Problems". *International Journal for Numerical Methods in Engineering*, Vol.39, pp. 829-846.
- [33] Chopra V., and Ziemba W., (1993). "The Effects of Errors in Means, Variances, and Covariances on Optimal Portfolio Choice". *The Journal of Portfolio Management*. Winter 1993, pp. 6-11.
- [34] Chriss N., Almgren R., (1999). "Optimal Execution of Portfolio Transactions". *The Journal of Risk*, 3(2), pp. 5-39.
- [35] Chunhachinda P., Dandapani K., Hamid S., and Prakas A., (1997). "Portfolio Selection and Skewness: Evidence from International Stock Markets". *Journal of Banking and Finance*, 21, pp. 143-167.
- [36] Committee on the Global Financial System, (1999). "Market Liquidity: Research Findings and Selected Policy Implications". Bank of International Settlements, CGFS Publications No. 11, ISBN 92-9131-068-9.
- [37] Constantinides G.M., (1986). "Capital Market Equilibrium with Transaction Costs". *Journal of Political Economy*, 94(4), pp. 842-863.
- [38] Crum R.L., Klingman D.D., and Tavis L.A., (1979). "Implementation of Large Scale Financial Planning Models: Solution Efficient Transformations". *Journal of Financial and Quantitative Methods*, 19(1), pp. 137-152.
- [39] Dantzig G.B., (1963). "Linear Programming and Extensions". Princeton University Press.
- [40] Datar V.T., Naik N.Y., Radcliffe R., (1998). "Liquidity and Stock Returns: an Alternative Test". *Journal of Financial Markets*, Vol 1(2), p 203-219.

Bibliography

- [41] Davis M.H.A., and Norman A.R., (1990). "Portfolio Selection with Transaction Costs". *Mathematics of Operations Research*, 15(4), pp. 676-713.
- [42] Dembo R.S., Mulvey J.M., and Zenios S.A., (1989). "Large-Scale Nonlinear Network Models and their Application". *Operations Research*, Vol. 37, pp. 353-372.
- [43] Dunbar N., (1998). "Meriwether's Meltdown". *Risk*, October 1998, pp. 32-36.
- [44] Easley D., and O'Hara M., (1987). "Price, Trade Size, and Information in Securities Markets". *Journal of Financial Economics*, 19, pp. 69-90.
- [45] Eichhorn D., Gupta A., and Stubbs E., (1998). "Using Constraints to Improve the Robustness of Asset Allocation". *The Journal of Portfolio Management*, pp. 41-48.
- [46] Engle R.F., and Lange J., (1997). "Measuring, Forecasting and Explaining Time Varying Liquidity in the Stock Market". NBER Working Papers w6129.
- [47] Engle R.F., and Russell J., (1998). "Autoregressive Conditional Duration: a New Model for Irregularly Spaced Transaction Data". *Econometrica*, 66, pp. 1127-1162.
- [48] Fadel G., Riley R., and Barthelemy J.F., (1990). "Two Point Exponential Approximation Method for Structural Optimization". *Structural Optimization*, 2, pp. 117-124.
- [49] Fama E.F., and French K., (1992). "The Cross-Section of Expected Stock Returns". *The Journal of Finance*, 47(2), pp. 427-465.
- [50] Fama E.F., and French K., (1996). "Multifactor Explanations of Asset Pricing Anomalies". *The Journal of Finance*, 51(1), pp. 55-84.
- [51] Fama E.F., MacBeth J., (1973). "Risk Return and Equilibrium: Empirical Tests". *Journal of Political Economy*, 81, pp. 607-636.

Bibliography

- [52] Fleury C., (1993). "Sequential Convex Programming for Structural Optimization Problems". Optimization of Large Structural Systems. volume I of NATO ASI, Kluwer Academic Publishers, pp. 531-553.
- [53] Fleury C., and Braibant V., (1986). "Structural Optimization - A New Dual Method Using Mixed Variables Information". International Journal for Numerical Methods in Engineering, 23, pp. 409-428.
- [54] Garfinkel R., and Nemhauser G.L., (1972). "Integer Programming". John Wiley and Sons.
- [55] Gill P.E., Murray W., Wright M.H., (1991). "Numerical Linear Algebra and Optimization". Vol. 1, Addison Wesley.
- [56] Glosten R., Harris L.E., (1988). "Estimating the Components of the Bid/Ask Spread". Journal of Financial Economics, 21, pp. 123-142.
- [57] Glover F., and Klingman D., (1981). "The simplex SON algorithm for LP/embedded network problems". Mathematical Programming Studies 15, pp. 148-176.
- [58] Glover F., Klingman D., Phillips N.V., (1992). "Network Models in Optimization and Their Applications in Practice". John Wiley and Sons.
- [59] Goldberg D., (1990). "A Note on Boltzmann Tournament Selection for Genetic Algorithms and Population-Oriented Simulated Annealing". Technical Report Nb. 90003, Department of Engineering Mechanics, University of Alabama.
- [60] Green R., and Hollifield B., (1992). "When Will Mean-Variance Efficient Portfolios be Well Diversified". Journal of Finance, 47(5), pp. 1785-1802.
- [61] Grossman S.J., and Miller M.H., (1988). "Liquidity and Market Structure". Journal of Finance, 43, pp. 617-633.
- [62] Hamon J., and Jacquillat B., (1997). "Expected Returns and Liquidity Premium on the Paris Bourse: an Empirical Investigation". Universite Paris Dauphine, Cahier de Recherche, Number 9706.

Bibliography

- [63] Handa P., and Schwartz R.A., (1996). "Limit Order Trading". *The Journal of Finance*, 51(5), pp. 1835-1861.
- [64] Hankansson N., (1971). "Multi-Period Mean-Variance Analysis: toward a General Theory of Portfolio Selection". *The Journal of Finance*, 26(4), pp. 857-884.
- [65] Hanoch G., Levy H., (1969). "The Efficiency Analysis of Choices Involving Risk". *Review of Economic Studies*, 36, pp. 335-346.
- [66] Harris L., (1998). "Optimal Dynamic Order Submission Strategies in Some Stylized Trading Problems". *Financial Markets, Institutions and Instruments*, 7(2), pp. 1-75.
- [67] Hartley R., (1985). "Linear and Nonlinear Programming". Ellis Horwood.
- [68] Hasbrouck J., (1988). "Trades, Quotes, Inventories, and Information". *Journal of Financial Economics*, 22, pp. 229-252.
- [69] Hasbrouk J., (1991). "Measuring the Information Content of Stock Trades". *Journal of Finance*, 46, pp. 571-595.
- [70] Held M., Wolfe P., and Crowder H.P., (1974). "Validation of Subgradient Optimization". *Mathematical Programming*, 6, pp. 62-88.
- [71] Holland J.H., (1975). "Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence". University of Michigan Press.
- [72] Jacoby G., Fowler D., and Gottesman A., (2000). "The Capital Asset Pricing Model and the Liquidity Effect: a Theoretical Approach". *Journal of Financial Markets*, 3, pp. 69-81.
- [73] Jarrow R., Subramanian A., (1997). "Mopping up Liquidity". *Risk*, 10, pp. 170-173.
- [74] Jobson J., and Korkie B., (1981). "Putting Markowitz Theory to Work". *Journal of Portfolio Management*, 7(4), pp. 70-74.
- [75] Johnson R., and Wichern D., (1992). "Applied Multivariate Statistical Analysis". Saddle River, NJ: Prentice-Hall.

Bibliography

- [76] Jones H. and Mitra G., (1997). "Solution to Convex Quadratic Programming Problem Using Interior Point Method". Technical Report TR/03/97, Department of Mathematics and Statistics. Brunel University.
- [77] Jorion P., (1992). "Portfolio Optimisation in Practice". Financial Analyst Journal, Jan./Feb., pp. 68-74.
- [78] Kahn R.N., Roulet J., and Tajbakhsh S., (1996). "Three Steps to Global Asset Allocation". The Journal of Portfolio Management, pp. 23-31.
- [79] Kallberg J.G., Ziemba W.T., (1983). "Comparison of Alternative Utility Functions in Portfolio Selection Problems". Management Sciences, 29, pp. 1257-1276.
- [80] Kellerer H., Masini R., Speranza M.G., (2000). "Selecting Portfolios with Fixed Costs and Minimum Transaction Lots". Annals of Operations Research, 99, pp. 287-304.
- [81] Kirpatrick S., Gelatt C.D., Vecchi M.P., (1983). "Optimization by Simulated Annealing". Science, 220, pp. 671-680.
- [82] Konno H. and Yamazaki H., (1991). "Mean-Absolute Deviation Portfolio Optimization Model and its Applications to Tokyo Stock Market". Management Science, 37, pp. 519-531.
- [83] Konno H., and Kobayashi K., (1997). "An Integrated Stock-Bond Portfolio Optimization Model". Journal of Economic Dynamics and Control, 21, pp. 1427-1444.
- [84] Koskosidis Y., and Duarte A., (1997). "A Scenario-Based Approach to Active Asset Allocation". The Journal of Portfolio Management, pp. 75-85.
- [85] Kothari S.P., Shanken J., Sloan R.G., (1995). "Another Look at the Cross-Section of Expected Stock Returns". Journal of Finance, 50, pp. 185-224.
- [86] Kyle A.S., (1985). "Continuous Auctions and Insider Trading". Econometrica, 53, pp. 1315-1335.

Bibliography

- [87] Levy H., and Markowitz H.M., (1984). "Approximating Expected Utility by a Function of Mean and Variance". *American Economic Review*, 69, pp. 308-317.
- [88] Levy H., Kroll Y., and Markowitz H.M., (1984). "Mean-Variance Versus Direct Utility Maximisation". *The Journal of Finance*, Vol. 39(1), pp. 47-61.
- [89] Lintner J., (1965). "The Valuation of Risky Assets and the Selection of Risky Investments in Stock Portfolios and Capital Budgets". *The Review of Economics and Statistics*, 47, pp. 13-37.
- [90] Litzenberger R.H., Ramaswamy K., (1980). "Dividends, Short Selling Restrictions, Tax-Induced Investor Clienteles and Market Equilibrium". *Journal of Finance*, 35, pp. 469-485.
- [91] Lobo M.S., Fazel M., Boyd S., (2002). "Portfolio Optimization with Linear and Fixed Transaction Costs". Submitted to *Operations Research*. Available at <http://www.stanford.edu/~boyd/portfolio.html>.
- [92] Loraschi A., Tettamanzi A., Tomassini M., Verda P., (1995). "Distributed Genetic Algorithms with an Application to Portfolio Selection Problems". *Artificial Neural Nets and Genetic Algorithms*, pp. 384-387.
- [93] Lucas A., and Klaassen P., (1998). "Extreme Returns, Downside Risk, and Optimal Asset Allocation". *The Journal of Portfolio Management*, Fall 1998, pp. 71-79.
- [94] Magill M.J.P., and Constantinides G.M., (1976). "Portfolio Selection with Transaction Costs". *Journal of Economic Theory*, 13, pp. 245-263.
- [95] Marins F., Senne E., Darby-Dowman K., Machado A., and Perin C., (1997). "Algorithms for Networks Piecewise-Linear Programs: A Comparative Study". *European Journal of Operational Research*, pp. 183-199.

Bibliography

- [96] Markowitz H.M., (1952). "Portfolio Selection". *Journal of Finance*, pp. 77-91.
- [97] McInish T.H., and Wood R.A., (1992). "An Analysis of Intraday Patterns in Bid/Ask Spreads for NYSE Stocks". *The Journal of Finance*, 47, pp. 753-764.
- [98] Merton R.C., (1987). "A Simple Model of Capital Market Equilibrium with Incomplete Information". *Journal of Finance*, 42, pp. 483-510.
- [99] Metropolis N., Rosenbluth A.W., Rosenbluth M.N., Teller A.H., Teller E., (1953). "Equation of State Calculations by Fast Computing Machines". *Journal of Chemical Physics*, 21, pp. 1087-1092.
- [100] Michaud R., (1998). "Efficient Asset Management". Harvard Business School Press, Boston.
- [101] Mulvey J.M. and Ziemba W.T., (1995). "Asset and Liability Allocation in a Global Environment". *Finance Elsevier*, 9, pp. 435-463.
- [102] Mulvey J.M., and Vladimirou H., (1988). "Solving Multistage Stochastic Networks: an Application of Scenario Aggregation". Report SOR-88-1, Dep. Of Civil Engineering And Operations Research, Princeton University.
- [103] Mulvey J.M., and Vladimirou H., (1992). "Stochastic Network Programming for Financial Planning Problems". *Management Science*, 38, pp. 1642-1665.
- [104] Nemhauser G.L., and Wolsey L.A., (1999). "Integer and Combinatorial Optimization". John Wiley and Sons.
- [105] O'Hara M., (1987). "Market Microstructure Theory". Blackwell Publications, pp. 215-251.
- [106] Perold A.F., (1984). "Large-Scale Portfolio Optimization". *Management Science*, 30, pp. 1143-1160.
- [107] Powell M.J.D., (1978). "The Convergence of Variable Metric Methods For Nonlinearly Constrained Optimization Calculations". *Nonlinear Programming 3*, Academic Press.

Bibliography

- [108] Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P.. (1997). "Numerical Recipes in C". Cambridge University Press.
- [109] Roy A.D., (1952). "Safety First and the Holding of Assets". *Econometrica*, 20, pp. 413-449.
- [110] Sharpe W., (1964). "Capital Asset Prices: a Theory of Market Equilibrium under Conditions of Risk". *Journal of Finance*, 19(3), pp. 425-442.
- [111] Sharpe W., (1991). "Capital Asset Prices with and without Negative Holdings". *Journal of Finance*, pp. 489-509.
- [112] Stratford R., (1996). "Neural Networks in Liability Management". PhD Thesis, Centre for Quantitative Finance, Imperial College, London.
- [113] Stulz R.M., (1981). "On the Effects of Barriers to International Investment". *The Journal of Finance*, 36, pp. 923-934.
- [114] Svanberg K., (1993). "The Method of Moving Asymptotes (MMA) with Some Extensions". In Rozvany, G., ed., *Optimization of Large Structural Systems*, volume I of NATO ASI, Kluwer Academic Publishers, pp. 555-566.
- [115] Svanberg K., (1995). "A Globally Convergent Version of MMA without Line-Search". *Proceedings of the First World Congress of Structural and Multidisciplinary Optimization*, Elsevier Science.
- [116] Theobald D., King M., and Bordewich M., (1999). "Valuing Market Liquidity". JP Morgan.
- [117] Uppal R., (1993). "A General Equilibrium Model of International Portfolio Choice". *Journal of Finance*, June, pp. 529-553.
- [118] Vanderbei R.J. and Carpenter T. J., (1993). "Symmetric Indefinite Systems for Interior Point Methods". *Mathematical Programming*, 58, pp. 1-32.
- [119] Vanderbei R.J., (1994). "An Interior Point Code for Quadratic Programming". Technical Report SQR-94-15, Statistics and Operations Research, Princeton University.

Bibliography

- [120] Vangelisti M., (1998). "Should You Worry about Transaction Costs?". Barra, Newsletter 166.
- [121] Wann-Ming W., (2000). "Model Formulation and Solution Algorithm of Traffic Signal Control in an Urban Network". Computers, Environment and Urban Systems, 24(4), pp. 355-377.
- [122] Wann-Ming W., and Jayakrishnan R., (1999). "Network Traffic Signal Optimization with Embedded Platoon Dispersion Simulation". Transportation Research Record, 1683, pp. 150-159.
- [123] Wolsey L.A., (1998). "Integer Programming". John Wiley and Sons.
- [124] Zhang W.H., Fleury C., (2002). "Composite Structures Optimization Using Sequential Convex Programming". Advances in Engineering Software November 2002, 33(7-10), pp. 697-711.