ARBEITSBERICHT
PROZESS- UND PRODUKT-ENGINEERING: METHODEN

2004/1, ISSN 1814-1307 (Print), ISSN 1814-1315 (Online)

IDENTIFYING WORST CASE SCENARIOS OF SECURITY PORTFOLIOS WITH QUASI-RANDOM SEARCH ALGORITHMS

Thomas Breuer thomas.breuer@fhv.at
Filip Pistovcak

13. Juli 2004

Fachhochschule Vorarlberg
Forschungszentrum
Prozess- und Produkt-Engineering

Identifying Worst Case Scenarios of Security Portfolios with Quasi-Random Search Algorithms

Filip Pistovčák Thomas Breuer

Research Centre for Process- and Product-Engineering
Vorarlberg University of Applied Sciences
Achstrasse 1, A-6850 Dornbirn

A Research Project
carried out from March 2001 to November 2003

funded by
Jubiläumsfonds der Oesterreichischen Nationalbank (Project No. 8364)
Zukunftsfonds des Landes Vorarlberg

13 July, 2004

Executive Summary

We report on the use of quasi-random numbers in searching for worst-case scenarios of security portfolios. A systematic search for the worst-case scenario above some minimal plausibility requires to find the global minimum of the portfolio-value function within a search domain of all sufficiently plausible scenarios, which usually is an ellipsoid in the high dimensional space of risk factors.

We compare the performance of Monte Carlo and Quasi Monte Carlo search algorithm, which use sequences of scenarios transformed from the unit cube. As a benchmark we use the Multilevel Coordinate Search algorithm of W. Huyer and A. Neumaier applied to the transformed problem on the cube. It turns out that QMC does not perform significantly better than MC for most parameter settings. This might be due to the destruction of low-discrepancy properties by the transformation from the cube to the ellipsoid, or to the fact that for generic portfolios the worst case scenario is on the surface of the ellipsoid.

Contents

Chapter 1

Worst Case Scenarios in Risk Management

Identifying plausible worst case scenarios once was nothing but an art in strategic management. Now it is becoming more of a science. The management of security portfolios more than other fields of management relies on mathematical models. It is perhaps for this reason that the *systematic* identification of plausible worst case scenarios developed in this field. In this section we briefly review the use of worst case scenarios in market risk management.

Risk is uncertainty about the future. For investors, risk is uncertainty about the future value of a portfolio. Usually there are several sources of risk in portfolios: the future value of a portfolio might be affected by market prices (market risk), by default or downgrading of counterparties (credit risk), by the liquidity of markets or by numerous other sources. Here we concentrate on market risk, caused by uncertainty about future market prices of traded securities.

Risk measures quantify the risk of a portfolio by a single number. The identification of plausible worst case scenarios automatically leads to a natural risk measure, Maximum Loss. In order to evaluate the advantages and disadvantages of Maximum Loss as a risk measure it is useful to compare it to a standard risk measure, Value at Risk.

1.1 Value at Risk, Some Shortcomings, and Alternatives

Value at Risk (VaR) is a statistical measure of the potential losses of a portfolio over a certain holding period. Roughly speaking, it is defined as the loss which will not be exceeded with a specified probability a assuming the portfolio remains constant throughout the holding period T. Therefore, losses in excess of VaR only should occur with a low probability $1-a$. (Thus, VaR is the current portfolio value minus the a-quantile of the profit-loss distribution at the end of the holding period.[1])

VaR became the standard risk measure for security portfolios when the Basle Committee on Banking Supervision admitted VaR for the calculation of regulatory capital requirements for market risk [4]. For this purpose the holding period and the confidence level equal ten days and 99%, respectively. Within the risk management process of a financial institution however, the parameters T and a could be set in a different way, according to the needs of the firm. Nevertheless, the length of the holding period should be chosen such that all positions in the portfolio can be liquidated during $[0, T]$ - even under adverse conditions. Thus, T depends on the assessment of how illiquid markets potentially could become. The confidence level should reflect the degree of severity of events in which management is interested.

Apart from the disagreement about how to choose T and a appropriately, there is increasing skepticism towards the assumptions on which many VaR implementations are based. There are first and foremost two assumptions the validity of which is debatable. For one, the market characteristics are assumed not to change in the future. There are folkore statements to the effect that VaR is the risk measure for "normal situations". Only if future market movements resemble those of the past models can produce reliable results. Yet, there have always been breaks in market movements. Of course, any statistical risk measure intended as a prediction has to cope with that kind of problem.

Furthermore numerous VaR models assume that changes in risk factors follow a multivariate normal distribution. However, changes in financial time series are, as a rule, not normally distributed. This holds true for the unconditional as well as for the conditional distribution (see for example [8, 21]). The non-normality is much more pronounced for the unconditional case than for the conditional one. For example, residuals of GARCH models which are assumed normal still do show kurtosis excess although a smaller one than the original returns. Thus, such time series are marked by fat tails. It follows that extreme changes in the risk factors are considerably more likely than accounted for under the assumption of a normal distribution. The slump in stock prices triggered by the equity crash of 1987, for instance, amounted to

---

[1] The a-quantile of the profit-loss distribution is unique only if the distribution function does not have constant value a in some range. Otherwise the set of a-quantiles is a closed interval. In such situations one conventionally defines VaR with the left end-point of the interval of a-quantiles.

something between 10 and 20 standard deviations. Considering that under normality a 7 standard deviations change should on average occur at one day in three billion years, the assumption of normality seems inadequate.

Apart from possible shortcomings in VaR implementations one might ask what can VaR really tell us about risk and what are the limitations of VaR as a risk measure. By its sheer definition as a quantile of the profit-loss distribution, a VaR-figure does not shed light on the dimension of extreme losses exceeding that figure. This is the first reason why a complementary measure such as stress testing might be necessary: stress tests should capture potential extreme losses.

Another important problem with VaR is that it is not in general a coherent risk measure. The notion of coherence is due to Artzner *et al.* [2, 3]. More specifically, VaR fails to be sub-additive in general. It is so only under rather restricted conditions. For guaranteeing sub-additivity, essentially the value of the portfolio has to be a linear function of risk factors the changes of which are elliptically distributed. If either the portfolio value is a non-linear function, or if the risk factors are non-elliptically distributed, then sub-additivity fails.

Let us consider an example.

Example 1 *Assume that a bank sets itself a VaR-limit of EUR 70 mill. Management give Trading Desk A a VaR-limit of EUR 50 mill and Trading Desk B a limit of EUR 20 mill. Some might consider this limit system to be conservative, since diversification effects are expected to bring down the joint VaR of the two trading desks well below the sum of the separate VaR of the two trading desks. But assume now that both trading desks hold an option position in the same underlying, which has normally distributed returns. The current value of the underlying is 10.000, Trading Desk A holds a million short European puts with a strike of 9.200 and Trading Desk B holds a million short European calls with a strike of 11.300. Assume both options mature in three months, the volatility of logarithmic returns is 5% and the risk free interest rate is also 5%. If neither desk has any other positions in its portfolio, the VaR at the 95%-level will be EUR 42.92 mill for Trading Desk A and EUR 18.47 mill for Trading Desk B. So both desks are comfortably below their VaR-limits. And yet, the joint VaR of both portfolios is EUR 80.91 mill—well above the VaR-limit the bank set itself.*

Note that in this example VaR-figures were calculated by 10'000 Monte Carlo simulations. A VaR-calculation with the popular square root formula would not have been admissible for this non-linear portfolio.

What is so awkward about the lack of sub-additivity is the fact that this can give rise to regulatory arbitrage or to the break-down of global risk management within one single firm. This is also a serious concern for regulators.

If regulation allows the capital requirement of a firm to be calculated as the sum of the requirements of its subsidiaries and if the requirements are based on VaR, the firm could create artificial subsidiaries in order to save regulatory capital.

As an alternative to Value at Risk Maximum Loss was introduced in [22, 6]. Another coherent risk measure is Expected Shortfall [2, 3, 23, 1]. Maximum Loss can be considered from two points of view: either as a risk measure, or as a by-product of a search for worst case scenarios. The basic idea is the following: Maximum Loss is the worst loss incurred in any of the scenarios above some minimal plausibility. A more formal definition of this will be given in Section 2.

In contrast to VaR, Maximum Loss is essentially a coherent risk measure. Especially the sub-additivity property holds in general. Therefore, if capital requirements are defined via MaxLoss, regulatory arbitrage is not possible by splitting a firm's portfolio into sub-portfolios and defining the overall requirement as sum of the requirements for the sub-portfolios.

1.2 A Practical Guide to Managing Risk with Worst Case Scenarios

We conclude by discussing some practical aspects of managing risk with stress tests, following [7]. First we present a way of reporting MaxLoss and the results of a systematic search for worst case scenarios. Then we show how knowledge of the worst case scenario suggests specific measures to reduce risk if desired.

How can the results of a systematic search of a worst-case scenario be presented in a concise and readily understandable manner? It is certainly not enough to simply report the values of the risk factors in the worst-case scenario. Listing the values of, for example, 500 risk factors in the worst-case scenario would hopelessly overtax the capacity of any recipient of the report. Consequently, reports should include only the most important risk factors of the worst-case scenario.

What are the "most important" risk factors of a worst-case scenario? Sensitivities are certainly not an appropriate indicator of the importance of a risk factor: sensitivities in the present market state are completely unrelated to the worst-case scenario to be characterized; and all sensitivities will be zero in the worst-case scenario if it is a local minimum.

The following approach appears more useful: The search for the key risk factors is a search for a subset of risk factors which have a certain ex-

	Risk Factors	**Relative Changes**	**Loss of P-Value**	**Percentage of MaxLoss**
Report 1	FTSE100	-13%	206%	74%
Report 2	FTSE100	-13%	264%	94%
	DJI	-8%		
Report 3	FTSE100	-13%	271%	97%
	DJI	-8%		
	NIK225	-5%		

Table 1.1: Reports about the worst case scenario of the Sample Portfolio, at three levels of detail.

planatory power, i.e. which explain the loss under the worst-case scenario up to a previously defined degree. For example, an explanatory power of 80% means that we are looking for a subset of the risk factors which will be able to explain at least 80% of the loss under the worst-case scenario. This means: Let us assume that, instead of the complete worst-case scenario $\boldsymbol{r}_{\mathrm{WC}} = (r_{\mathrm{WC},1}, \dots, r_{\mathrm{WC},n})$, only the values of a subset of w risk factors are reported. This corresponds to a simplified report scenario

$$\boldsymbol{r}_{\mathrm{report}} = (r_{\mathrm{CM},1}, \dots, r_{\mathrm{WC},i_1}, \dots, r_{\mathrm{WC},i_2}, \dots, r_{\mathrm{WC},i_w}, \dots, r_{\mathrm{CM},n}),$$

where the risk factors $r_{i_1}, r_{i_2}, \dots, r_{i_w}$ have their worst-case values r_{WC,i_1},r_{WC,i_2}, $\dots, r_{\mathrm{WC},i_w}$, and all other risk factors have their actual values. The subset of risk factors will explain 80% of the loss suffered from the worst-case scenario if

$$P_0 - P_1(\boldsymbol{r}_{\mathrm{report}}) \geq 0.8(P_0 - P_1(\boldsymbol{r}_{\mathrm{WC}})),$$

where P_0 is the current portfolio value and $P_1(\boldsymbol{r})$ is the portfolio value in scenario $\boldsymbol{r}$ at the end of the holding period. The task is to find a fairly small w and a set of w risk factors which has a high explanatory power. This can be solved by optimization algorithms in a discrete w-dimensional space.

As an example of how this reporting method works, Table 1.1 shows results of identifying the one, two, and three most important risk factors in the worst case scenario of some Sample Portfolio. The Worst Case Loss of the Sample Portfolio is 274% of its present value. In this worst case the portfolio has a negative value, which can be due to the presence of written options or other short positions.

This kind of information can be used to report in an understandable way about a stress test program. For example Report 1 would be:

"Leaving all other risk factors unchanged, a move of -13% in the FTSE100 would lead to a relative loss of 206%.

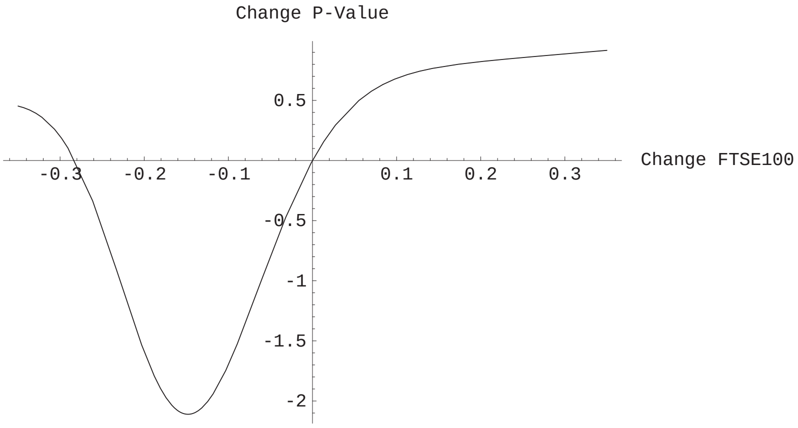

Figure 1.1: *The value of the Sample Portfolio as a function of relative changes in the FTSE 100, the risk factor which accounts for most of the losses in the worst case scenario.*

Figure 1.1 shows a plot of the portfolio value as a function of relative changes in the FTSE100, the risk factor which is responsible for 74% of losses in the worst case scenario:

Report 2 would be:

"Leaving all other risk factors unchanged, a move of -13% in the FTSE100 and of -8% in the DJI would lead to a relative loss of 264%.

Since with two risk factors we already can explain 94% of the loss in the worst case scenario, there is no point in reporting more risk factors. Figure 1.2 shows a plot of the portfolio value as a function of relative changes in the two risk factors which together are responsible for the bulk of losses in the worst case scenario. It is evident from this plot (and also from the first two rows of Table 1.1) that the portfolio value depends much more sensitively on the FTSE than on the DJI. We also see that a move of roughly -15% in the FTSE and of roughly -10% in the DJI would be very harmful to the portfolio.

If a bank decides that it does not want to take the risk of this portfolio it can buy insurance in the form of risk reducing positions. Taking up a so-called condor in the FTSE with a peak at -10% of the present value of the FTSE reduces MaxLoss from 276% to 178%. The cost of such an insurance is close to zero: it is 3 times a millionth part of the uninsured portfolio value. The effect of the insurance is nicely displayed when we compare Figure 1.3 to Figure 1.1 The more precisely we know the dangerous scenarios for a

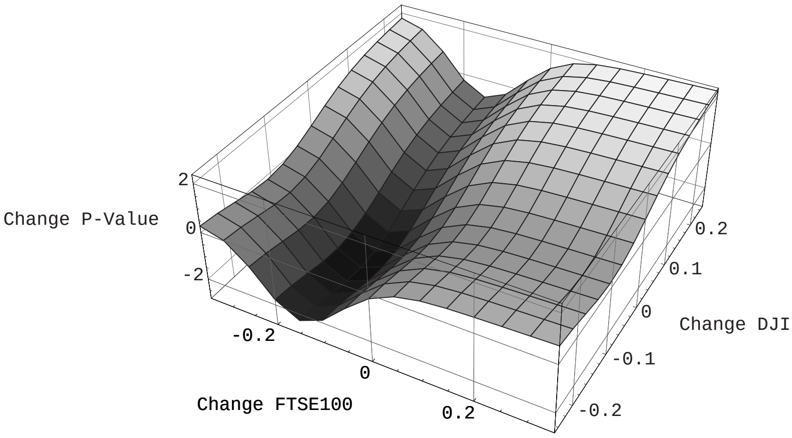

Figure 1.2: *The value of the portfolio as a function of relative changes in the FTSE 100 and the DJI, the two risk factors which together account for 96% of the losses in the worst case scenario.*

portfolio, the easier it is to take up risk reducing positions which at the same time are affordable and do not affect the gain potential in other market states.

Let us sum up this section. There are two main benefits of working with MaxLoss as a risk measure instead of VaR:

(1) MaxLoss is a coherent risk measure whereas VaR is coherent only under rather restricted assumptions. Especially the violation of the sub-additivity property can cause problems from the point of view of global risk management and regulation.

(2) MaxLoss not only provides a risk assessment measured in monetary units but also a worst-case scenario $\boldsymbol{r}_{\mathrm{WC}}$ which has created that figure. The knowledge of the worst-case scenario can be the basis of informed risk decisions and suggest possible risk reducing transactions.

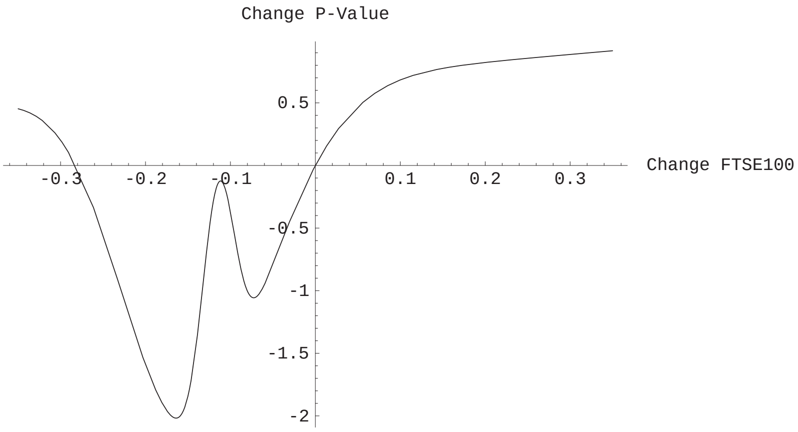

Figure 1.3: *The value of the insured portfolio as a function of relative changes in the FTSE 100, the risk factors which accounts for most of the losses in the worst case scenario. Compare this plot to the corresponding plot of the uninsured portfolio, Figure 1.1. Observe the effect of risk reducing positions, which produces profits if the FTSE100 drops by roughly 10%.*

Chapter 2

Optimisation Problem

Identifying a plausible worst case scenario is an optimisation problem. In defining this optimisation problem more formally, we use the following notation. We have a portfolio of securities with value depending on risk factors $x_1, \ldots, x_n$, where $x_i \in \mathbf{R}$ $(i = 1, \ldots, n)$. The relevant risk factors could be values of equity indices, interest rates, foreign exchange rates, etc. The values of these risk factors characterise the market situation as it is of relevance to the portfolio. We assume we are at the date $t = 0$, and the portfolio has current value P_0. The vector of values of the risk factors today is $\mathbf{x}_0 \in \mathbf{R}^n$.

Our aim is to stress test the portfolio, i.e., to estimate how big losses we can expect if the market situation changes adversely to our portfolio. We also assume the portfolio does not changes during the time horizon. In a sense this gives us an upper bound for the potential loss because, in reality, some counteractions during the day can mitigate losses if market moves adversely. The intial values of risk factors, $\mathbf{x}_0$, are known, however the values at the end of the time horizon, $\mathbf{x}_1 \in \mathbf{R}^n$, are uncertain. We define componentwise $\mathbf{r} = \log(\mathbf{x}_1) - \log(\mathbf{x}_0)$, the logarithmic return of the vector of risk factors during the time horizon, as the *scenario.* Since $\mathbf{x}_1$ is uncertain, $\mathbf{r}$ is a random vector with some distribution. In finance one often assumes that this log-return is normally distributed with zero mean and non-zero variance, called *volatility.* However, it is also possible to model the log-return assuming some fat-tailed distribution like Student's t-distribution. We denote the value of the portfolio at the end of the time horizon given some random scenario $\mathbf{r}$ as $P_1 = P_1(\mathbf{r})$.

2.1 The Objective Function

In our setup stress testing consists of selecting the scenarios $\mathbf{r}$ according to specific criteria and calculating the values of our portfolio in these scenarios. By comparing the portfolio value in these scenarios, $P_1(\mathbf{r})$, with the current portfolio value P_0 one can assess the loss that would be incurred if the market moved from the current market state to the state represented by the scenario $\mathbf{r}$ without rebalancing the portfolio:

$$\mathrm{Loss}(\mathbf{r}) = P_0 - P_1(\mathbf{r}).$$

The function P_1 is a real valued function of n real valued variables. The function P_1 depends on the portfolio. At the moment we do not make any further assumptions about P_1. Often it is not given in an explicit form but just as a result of valuation routines. It need not be continuous or differentiable. It will in general not be linear. The only thing we can assume is that we can evaluate P_1 at any point. But depending on the complexity of the portfolio, one evaluation of P_1 might take up to several minutes.

In this setup the portfolio is restricted to financial instruments which do not depend on the whole evolution of the risk factors during the time horizon, which excludes for example path-dependent options. In a more general framework it would be possible to consider multi-step scenarios. But this was beyond the scope of this project.

Now the question arises which criteria one should use to select the scenarios. The traditional approach is to take some standard set of scenarios as in [9], for example, parallel yield curve shifts, changes in the steepness or shape of yield curves, changes in values of equity indices, changes in values of key currencies, etc. A more sophisticated approach involves a systematic search for the *worst case scenario*:

$$\mathbf{r}_{wc} \in \mathrm{argmin}_{\mathbf{r}\in\mathcal{E}} P_1(\mathbf{r}), \tag{2.1}$$

where "argmin" denotes the set of arguments $\mathbf{r} \in \mathcal{E}$ that minimise P_1, and $\mathcal{E}$ is a search domain (or "trust region" as it is called in [22]). For simplicity we assume that the minimum exists in $\mathcal{E}$. This is usually the case if $\mathcal{E}$ is closed and the portfolio does not contain some exotic financial instruments with non-continuous value at the end of the time horizon. The Maximum Loss is then defined as

$$\mathrm{MaxLoss}_{\mathcal{E}} = P_0 - P_1(\mathbf{r}_{wc}) = \max_{\mathbf{r}\in\mathcal{E}}(P_0 - P_1(\mathbf{r})). \tag{2.2}$$

So the optimisation problem is to minimise the portfolio value $P_1(\mathbf{r})$ subject to $\mathbf{r} \in \mathcal{E}$.

2.2 Choosing the Search Domain

Now the question is how to choose the search domain $\mathcal{E}$. We are looking for scenarios which lead to serious losses, however we should also require them to be plausible in some sense. But how should we measure the plausibility? Certainly, it should be a probabilistic concept: The higher the probability of a move from the current market state to the scenario $\mathbf{r}$, the higher the plausibility of the scenario $\mathbf{r}$ should be.

We take the *plausibility* of a scenario $\mathbf{r} \in \mathbf{R}^n$ as the probability of all scenarios with density lower than or equal to the density of the scenario $\mathbf{r}$:

$$\mathrm{Plaus}(\mathbf{r}) = \Pr\left\{\mathbf{s} \in \mathbf{R}^n \mid f(\mathbf{s}) \leq f(\mathbf{r})\right\}. \tag{2.3}$$

Alternatively, it is also possible to take the plausibility of the scenario to be the Mahalanobis distance of that scenario from the current market state. Both of these concepts have some advantages and some disadvantages.

We assume the logarithmic risk factor changes to be elliptically distributed. An n-dimensional distribution with the density function f is *elliptic* if f is of the form

$$f(\mathbf{r}) = (\det \Sigma)^{-1/2} g(\mathbf{r}^T \cdot \Sigma^{-1} \cdot \mathbf{r}), \quad \mathbf{r} \in \mathbf{R}^n, \tag{2.4}$$

where Σ is a symmetric, positive definite $(n \times n)$-matrix, g is a strictly decreasing function from the non-negative numbers to the non-negative numbers satisfying

$$\int_0^\infty t^{n-1} g(t^2)dt = \Gamma(n/2)/(2\pi^{n/2})$$

in order to ensure normalisation. If g is continuous and strictly decreasing then the distribution is unimodal, i.e. it has only one local maximum.

If risk factor log-changes follow an unimodal elliptic distribution, the search domain will be the interior (together with the boundary) of an n-dimensional ellipsoid:

$$\begin{aligned} \mathcal{E} &= \{\mathbf{r} \in \mathbf{R}^n \,|\, \mathrm{Plaus}(\mathbf{r}) \geq \alpha,\ \alpha \in [0,1]\} & (2.5)\\ &= \{\mathbf{r} \in \mathbf{R}^n \,|\, \mathbf{r}^T \cdot \Sigma^{-1} \cdot \mathbf{r} \leq k_{max}^2,\ k_{max} \in \mathbf{R}_+\}. & (2.6) \end{aligned}$$

The shape of this ellipsoid is determined by the matrix Σ (which is in our setting the variance-covariance matrix of the log-returns of the risk factors), and the size of the ellipsoid is given either by a required plausibility level α or, equivalently, by a parameter k_{max}. $\mathcal{E}$ contains only scenarios with plausibility higher or equal to α or scenarios in which the log-return of no risk factor is more than k standard deviations from the center (current market state).

To sum up, the optimization problem is to find the minimum of the portfolio value function P_1 within the n-dimensional ellipsoid $\mathcal{E}$. The point where this minimum is achieved is the worst case scenario which leads to the maximum loss among the scenarios above the plausibility threshold. If there is at least one risk factor of which the portfolio value is a monotone function, the worst case scenario will be on the surface of the ellipsoid. This will usually be the case.

Chapter 3

The Search Algorithms

In the project, before starting to design and test optimisation algorithms we had to build the framework necessary to evaluate the objective function. First we had to choose which market data to use as underlying risk factors. The requirement was to have some regularly published market data set covering the most important risk factors. We have chosen the data published daily by RiskMetrics (see [20] or [14]). They are publicly available (though with some lag), and, additionally, they do not contain only values of risk factors on some date, but also precalculated volatilities and correlations of these risk factors on that day. As the programming language we have chosen Matlab, with Microsoft Excel as an interface.

A first cornerstone was to implement the portfolio valuation as a function of RiskMetrics's risk factors, which are sometimes different from the parameters used in valuations in finance textbooks. Matlab offers some financial toolboxes, so we could partially use already available routines. However, not all financial instruments were covered by these toolboxes, and sometimes, even if they were included, they were too slow for our purposes. Altogether we implemented 21 financial instrument types.

The second step was to implement various optimisation routines searching for worst case scenarios. Though the aim was to develop a quasi-Monte Carlo routine, we needed also its Monte Carlo counterpart and, if possible, also some direct (deterministic) algorithm to compare the results with. We implemented 2 non-stochastic searches, 5 Monte Carlo (MC) algorithms, and for the best of the MC algorithms we created also its quasi-Monte Carlo (QMC) counterpart.

Fmincon Algorithm by Matlab

The first natural candidate for a search routine was the deterministic constrained optimization algorithm implemented in Matlab—`fmincon`. However this algorithm had relatively big troubles with elliptic boundaries, and its results were so poor that we did not consider it as a usable algorithm in our problem.

Multilevel Coordinate Search

The second deterministic algorithm we tested was the *Multilevel Coordinate Search* (MCS) introduced in [13]. It is a bound constrained optimization algorithm working in a cube:

$$\min f(\mathbf{x}) \tag{3.1}$$

$$\text{s.t. } \mathbf{x} \in [\mathbf{u}, \mathbf{v}] \tag{3.2}$$

with finite or infinite bounds, where

$$[\mathbf{u}, \mathbf{v}] = \{\mathbf{x} \in \mathbf{R}^n \mid u_i \leq x_i \leq v_i, i = 1, \ldots, n\}, \tag{3.3}$$

with $\mathbf{u}$ and $\mathbf{v}$ being $\mathbf{R}^n$-vectors with components in $\bar{\mathbf{R}} = \mathbf{R} \cup \{-\infty, \infty\}$, and $u_i < v_i$ for $i = 1, \ldots, n$.

In our setting, we defined the bounds $[\mathbf{u}, \mathbf{v}]$ as the $[-1, 1]^n$-cube and the function f as $f(\mathbf{x}) = P_1(h(\mathbf{x}))$, where $h : [-1, 1]^n \to \mathbf{R}^n$ maps points from the cube to our elliptic search domain,

$$h(\mathbf{x}) = k_{max} \cdot \text{fac}\left(\sqrt{\mathbf{x}^T \mathbf{x}}\right) \cdot L \cdot \mathbf{x}, \tag{3.4}$$

where k_{max} characterises the size of the search domain, fac is the function

$$\text{fac}(s) = \begin{cases} 0 & \text{if } s = 0, \\ s(s-3) + 3 & \text{if } s \in (0, 1), \\ \frac{1}{s} & \text{if } s \geq 1, \end{cases} \tag{3.5}$$

and L is the Cholesky-decomposition of Σ:

$$\Sigma = LL^T. \tag{3.6}$$

3.1 Zoom-in Procedure

Most of our algorithms use a zoom-in procedure (called “localisation of search in [15]) in order to increase efficiency. The basic idea is to gradually restrict search to the parts of the search domain in which there is a higher chance to find the optimum. This gradual restriction increases speed, but it also increases the danger of missing the global optimum.

As stated above, our search domain $\mathcal{E}$ is an n-dimensional ellipsoid (where n is the number of underlying risk factors). The basic idea of this algorithm is to find the worst case scenario through a sequence of M focusation steps within $\mathcal{E}$. During the search we are zooming to the most interesting areas in $\mathcal{E}$ until we find the best approximation of the worst case scenario. The speed of zooming is given by a shrink factor $\delta \in (0, 1)$.

In the first focusation step we generate N pseudo-random scenarios in $\mathcal{E}$. Call these points $\mathbf{r}_1^1, \mathbf{r}_2^1, \ldots, \mathbf{r}_N^1$. Then we calculate the portfolio value in all these N points, and take the one which leads to the maximum loss:

$$\mathbf{r}_1^* \in \mathrm{argmin}_{\mathbf{r}_1^1, \mathbf{r}_2^1, \ldots, \mathbf{r}_N^1} P_1(\mathbf{r}_i^1). \tag{3.7}$$

This scenario is the optimum in the first focusation step, and will be the starting point in the second focusation step.

In the second focusation step we shift the ellipsoid such that it has its center in $\mathbf{r}_1^*$. We also shrink it by a factor of δ, and start to sample points in this new shifted and shrinked ellipsoid:

$$\mathcal{E}_2 = \{\mathbf{r} \in \mathbf{R}^n \mid (\mathbf{r} - \mathbf{r}_1^*)^T \Sigma^{-1} (\mathbf{r} - \mathbf{r}_1^*) \leq (\delta \cdot k_{max})^2\}. \tag{3.8}$$

The sampling procedure is the same as in the first focusation step—the points are generated in $\mathcal{E}_2$ using the algorithm OSPHERE. However, since we do not want to leave the original search domain, we take only points which are in the intersection of our shrinked ellipsoid $\mathcal{E}_2$ and the original search domain $\mathcal{E}$. We sample until we have again N points, $\mathbf{r}_1^2, \mathbf{r}_2^2, \ldots, \mathbf{r}_N^2$, in $\mathcal{E} \cap \mathcal{E}_2$, and take as $\mathbf{r}_2^*$ the one which leads to the greatest loss:

$$\mathbf{r}_2^* \in \mathrm{argmin}_{\mathbf{r}_1^*, \mathbf{r}_1^2, \mathbf{r}_2^2, \ldots, \mathbf{r}_N^2} P_1(\mathbf{r}_i^2). \tag{3.9}$$

Similarly, $\mathbf{r}_2^*$ is the center of the shrinked ellipsoid

$$\mathcal{E}_3 = \{\mathbf{r} \in \mathbf{R}^n \mid (\mathbf{r} - \mathbf{r}_2^*)^T \Sigma^{-1} (\mathbf{r} - \mathbf{r}_2^*) \leq (\delta^2 \cdot k_{max})^2\}, \tag{3.10}$$

in the third focusation step.

We repeat this procedure through M focusation steps. The optimal scenario after all focusation steps is taken as an approximation of the worst case scenario:

$$\mathbf{r}_{wc} \in \mathrm{argmin}_{\mathbf{r}_{M-1}^*, \mathbf{r}_1^M, \mathbf{r}_2^M, \ldots, \mathbf{r}_N^M} P_1(\mathbf{r}_\mathbf{i}^\mathbf{M}). \tag{3.11}$$

Here is a little pseudo-code of the zoom-in procedure:

```
$\mathbf{r}_{wc} = \mathbf{0}$                    (set initial worst case scenario)
$\mathbf{c} = \mathbf{0}$                         (set initial center of ellipsoid)
$k = k_{max}$                                     (set initial size of ellipsoid)
for $i = 1$ to $M$                                (make $M$ focusation steps)
    for $j = 1$ to $N$                            (make $N$ evaluations in foc. step)
        $\mathbf{r} = \mathrm{scn}(\mathcal{E}, \mathbf{c}, k)$   (sample a random scenario in the
                                                   intersection of the ellipsoid
                                                   with center $\mathbf{c}$ and size $k$
                                                   and the original domain $\mathcal{E}$)
        if $P_1(\mathbf{r}) < P_1(\mathbf{r}_{wc})$
            $\mathbf{r}_{wc} = \mathbf{r}$        (new worst case scenario)
        end if
    end for
    $\mathbf{c} = \mathbf{r}_{wc}$                (center in the next foc. step)
    $k = k \cdot \delta$                          (shrink the ellipsoid)
end for
```

This focusation procedure yields $\mathbf{r}_{wc}$ as approximation to the worst case scenario and $P_0 - P_1(\mathbf{r}_{wc})$ as approximation to MaxLoss. We use the expression `scn`, which can be any algorithm producing a random scenario in the intersection of the original domain $\mathcal{E}$ with the ellipsoid with center $\mathbf{c}$ and radius k.

In the terminology of evolutionary optimisation strategies, this zoom-in procedure would be called a $(1 + \lambda)$-selection with $\lambda = N$.

3.2 Cube and Ell

For `scn` we need to specify algorithms producing a random scenario in the intersection of the original domain $\mathcal{E}$ with the ellipsoid with center $\mathbf{c}$ and radius k. We considered two large classes.

The one class (which we call Cube) generates scenarios by transforming uniform sequences from a cube to the ellipsoid with the map h of eq. (3.4).

```
Function scn-Cube-MC($\mathcal{E}, \mathbf{c}, k$)
                                                  (input: center $\mathbf{c}$ and max. distance $k$)
repeat
    $\mathbf{x} = \mathrm{rand(n)}$               (sample a point in a $(0,1)^n$-cube)
    $\mathbf{x} = 2 \cdot \mathbf{x} - 1$         (transform it to the $(-1,1)^n$-cube)
    $\mathbf{s} = \sqrt{\mathbf{x}^T \mathbf{x}}$
    $\mathbf{r} = \mathbf{c} + k \cdot fac(\mathbf{s}) \cdot L \cdot \mathbf{x}$   (map it to our shifted ellipsoid
```

```
                                like in equation (3.4))
until $\mathbf{r} \in \mathcal{E}$                 (take only the point in $\mathcal{E}$)
return r
```

`scn-Cube` first generates pseudo-random scenarios in the $(-1,1)^n$-cube, and uses the function h of eq. (3.4) to map them to our shrinked ellipsoid $\mathcal{E}_i$.

The other class (which we call Ell) produces scenarios directly in the ellipsoid using the algorithm OSPHERE of Fishman [11], pp. 234–235.[1] This algorithm generates points in an n-dimensional ellipsoid using n numbers sampled from the $N(0,1)$ distribution and one $Beta(n,1)$ number for each generated point.

```
Function scn-Ell-MC($\mathcal{E}, \mathbf{c}, k$)
                          (input: center c and max. distance $k$)
repeat
    $\mathbf{a} = \mathtt{normrnd(n)}$         (sample vector of $n$ indep. $N(0,1)$
                           numbers)
    $b = \mathtt{betarnd(n,1)}$      (sample number from $Beta(n,1)$
                           distribution)
    $\mathbf{y} = b \cdot \frac{\mathbf{a}}{\sqrt{\mathbf{a}^T\mathbf{a}}}$          (random point in a unit sphere)
    $\mathbf{x} = \mathbf{c} + k \cdot L \cdot \mathbf{y}$       (map it to our shifted ellipsoid; here
                           $L$ is Cholesky decomposition of $\Sigma$
                           s.t. $\Sigma = LL^T$)
until $\mathbf{x} \in \mathcal{E}$              (take only the point in $\mathcal{E}$)
return x
```

So the function `scn-Ell` samples a point in the ellipsoid with center $\mathbf{c}$ and maximum Mahalanobis distance k. If this point is also inside $\mathcal{E}$ then the function returns it, if not then the procedure is repeated until it finds one in $\mathcal{E}$.

3.3 MC and QMC

In numerical integration Quasi-Monte Carlo (QMC) shows faster convergence than Monte Carlo. One might hope for a similar speed up also in optimisation. When trying to apply QMC for the search of worst case scenarios in ellipsoids we face a problem. So far, there is no algorithm known which generates low-discrepancy sequences in an n-dimensional ellipsoid. Though

---

[1] In [11] on p. 235, equation (141) should read $z^T B^{-1} z = r^2$ instead of $z^T B z = r^2$.

it is possible to generate points in a cuboid which encloses the ellipsoid, this is practical only if n is low. The reason can be illustrated by an example of a unit cube and a sphere with radius 1/2 inside this cube. As the number of dimensions grows, the unit cube has still the volume equal to one, while the volume of the sphere goes quickly to zero by a factor of the order of $1/2^n$. If $n = 9$ the volume of the sphere is less than 1% of the volume of the cube, and already for $n = 20$ we will hardly find any point out of a million in the sphere. This is an enormous amount of wasted sample points, and also the computational time used to find enough points is considerable.

Therefore we tried an alternative way. The procedures `scn-Ell` and `scn-Cube` both use some random numbers. `scn-Ell` calls `normrnd(n)` to produce a sample vector of n independent $N(0, 1)$ numbers and calls `betarnd(n, 1)` to produce sample number from a $Beta(n, 1)$ distribution. `scn-Cube` calls `rand(n)` to produce a sample a point in a $(0, 1)^n$-cube.

To construct a Quasi-Monte Carlo version `scn-Ell-QMC` of `scn-Ell-MC` we replaced the pseudo-random normally distributed numbers by a special prepared sequence. This was made by passing a Niederreiter-Xing low-discrepancy sequence (see [16] and [17]) through the inverse of the normal cumulative distribution function (`norminv` function of the Matlab software). The Niederreiter-Xing sequences were produced by a software implementation of Pirsic (see [19]).

```
Function scn-Ell-QMC($\mathcal{E}, \mathbf{c}, k$)
                                (input: center $\mathbf{c}$ and max. distance $k$)
repeat
    $\mathbf{q} = \mathtt{nx(n)}$               ($n$-dimensional Niederreiter-Xing
                                 point)
    $\mathbf{a} = \mathtt{norminv}(\mathbf{q})$          (produces $n$ indep.  $N(0,1)$ numbers)
    $b = \mathtt{betarnd}(\mathtt{n}, 1)$         (sample number from $Beta(n,1)$
                                 distribution)
    $\mathbf{y} = b \cdot \frac{\mathbf{a}}{\sqrt{\mathbf{a}^T\mathbf{a}}}$            (random point in a unit sphere)
    $\mathbf{x} = \mathbf{c} + k \cdot L \cdot \mathbf{y}$        (map it to our shifted ellipsoid;
                                 here $L$ is Cholesky decomp. of $\Sigma$
                                 s.t. $\Sigma = LL^T$)
until $\mathbf{x} \in \mathcal{E}$                  (take only the point in $\mathcal{E}$)
return x
```

The only difference between the scn-Ell-MC and the scn-Ell-QMC algorithms is in how $N(0, 1)$ numbers are sampled. MC uses the Matlab function `randn`, QMC uses points from an n-dimensional Niederreiter-Xing sequence and then applies `norminv`, the Matlab's inverse of the cumulative distribution function.

To avoid ±∞, all points containing zero or one in one coordinate are removed from the Niederreiter-Xing sequence before `norminv` is applied. Note that we did not change the pseudo-randomly generated $Beta(n, 1)$ numbers, so `scn` is in fact something like a pseudo-quasi-sampling function. Our objective was to test, how this approach affects the results, and we hoped that replacing at least some pseudo-random parts of our algorithm by the quasi-random parts would improve the convergence.

To construct a Quasi-Monte Carlo version `scn-Cube-QMC` of `scn-Cube-MC` we replaced the MC-sequences in the unit cube by an n-dimensional Niederreiter-Xing low-discrepancy sequence. So the only difference between the MC and the QMC algorithms is in how they sample points in the unit cube. MC uses the Matlab function `rand`, QMC uses the n-dimensional Niederreiter-Xing sequence. Our objective was to test, how this approach affects the results.

```
Function scn-Cube-QMC($\mathcal{E}, \mathbf{c}, k$)
                                  (input: center $\mathbf{c}$ and max. distance $k$)
repeat
    $\mathbf{x} = \mathrm{nx(n)}$                 ($n$-dimensional Niederreiter-Xing
                                   point)
    $\mathbf{x} = 2 \cdot \mathbf{x} - 1$             (transform it to the $(-1, 1)^n$-cube)
    $\mathrm{s} = \sqrt{\mathbf{x}^T \mathbf{x}}$
    $\mathbf{r} = \mathbf{c} + k \cdot fac(\mathrm{s}) \cdot L \cdot \mathbf{x}$   (map it to our shifted ellipsoid
                                   like in equation (3.4))
until $\mathbf{r} \in \mathcal{E}$                  (take only the point in $\mathcal{E}$)
return $\mathbf{r}$
```

3.4 Staying on the Surface

For most security portfolios there will be at least one risk factor of which the portfolio value is a monotone (increasing or decreasing) function. For such portfolios the minimum will be on the surface of the ellipsoid, not in the interior. In order to take this into account one might consider algorithms which only produce scenarios on the surface. The simplest one could look like this:

```
for $j = 1$ to $N$          (make $N$ evaluations)
    $\mathbf{r} =$ scn-surf($\mathcal{E}$)    (sample a random scenario
                        on the surface of $\mathcal{E}$)
    if $P_1(\mathbf{r}) < P_1(\mathbf{r}_{wc})$
        $\mathbf{r}_{wc} = \mathbf{r}$           (new worst case scenario)
    end if
```

```
    end for
```

`scn-surf` represents some function producing a scenario on the surface of $\mathcal{E}$. A version of Ell-MC with $b = 1$ would do:

```
Function scn-Surf-MC($\mathcal{E}$)
                            (input: $\mathcal{E}$)
    $\mathbf{a} = \mathrm{normrnd(n)}$         (sample vector of $n$ indep. $N(0,1)$
                             numbers)
    $\mathbf{y} = \frac{\mathbf{a}}{\sqrt{\mathbf{a}^T\mathbf{a}}}$          (random point on the
                            surface of unit sphere)
    $\mathbf{x} = L \cdot \mathbf{y}$           (map it to our shifted ellipsoid; here
                             $L$ is Cholesky decomposition of $\Sigma$
                             s.t. $\Sigma = LL^T$)
return x
```

And a QMC version of `scn-surf` could be the following algorithm Surf-QMC which is a version of Ell-QMC with $b = 1$:

```
Function scn-Surf-QMC($\mathcal{E}$)
                               (input: $\mathcal{E}$)
    $\mathbf{q} = \mathrm{nx(n)}$               ($n$-dimensional Niederreiter-Xing
                                point)
    $\mathbf{a} = \mathrm{norminv}(\mathbf{q})$          (produces $n$ indep.  $N(0,1)$ numbers)
    $\mathbf{y} = \frac{\mathbf{a}}{\sqrt{\mathbf{a}^T\mathbf{a}}}$             (random point on the
                               surface of unit sphere)
    $\mathbf{x} = L \cdot \mathbf{y}$              (map it to our shifted ellipsoid;
                                here $L$ is Cholesky decomp. of $\Sigma$
                                s.t. $\Sigma = LL^T$)
return x
```

Chapter 4

Tests of the Algorithms

We tested the algorithms on two portfolios. In all experiments we assumed log-normally distributed changes of risk factors, used a time horizon of one day, and required the plausibility of a worst case scenario to be at least 1 %.

4.1 The Portfolios

Our first portfolio was a portfolio of 6 put options forming two so-called condors on Nikkei 225 and S&P 100 indices, and a cash position. This portfolio had 7 risk factors, so we optimised over a 7-dimensional ellipsoid. (The two equity indices, the USD/JPY exchange rate and two yield vertices for each currency were the risk factors. Time to maturity of the options was 47 calendar days.) Figure 4.1 shows the shape of the option portfolio value function around the worst case scenario, as a function of the two most important risk factors.

This figure gives the somewhat misleading impression that for the option portfolio the global minimum within the search domain is a local minimum. But this impression is an artifact of the reduced, two-dimensional, representation. If we take into account the remaining five dimensions, the worst case scenario is not an unconstrained local minimum. It is on the surface of the elliptic search domain. As noted above, this is a generic feature of security portfolios. If the portfolio value depends monotonically on at least one risk factor, the worst case scenario will be on the surface of the ellipsoid.

The second portfolio consisted of equity indices of 22 OECD countries weighted by the fraction of their Gross Domestic Product (GDP) to the total GDP of all these 22 countries. Table 4.1 illustrates the portfolio construction.

The first column lists all OECD countries for which we had relevant market data. Their GDP is in the second column. The total GDP of all

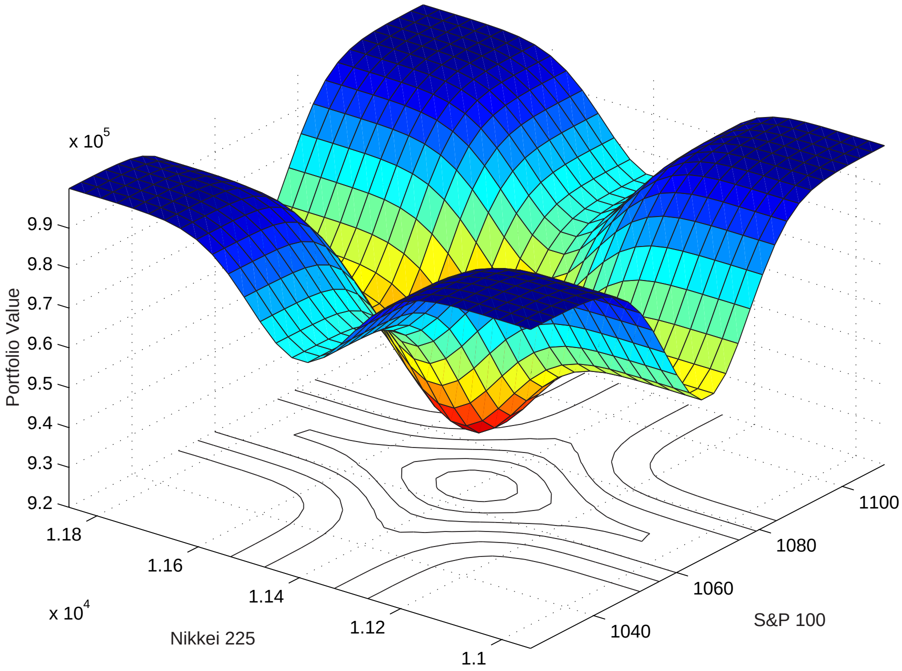

Figure 4.1: Value of the option portfolio as a function of the Nikkei 225 and S&P 100 indices—two most important risk factors—around the worst case scenario

these 22 countries is \$24 600 bil. The third column contains fraction of each country's GDP to this total. We constructed our equity portfolio with the total value of \$10 mil. from the equity indices of these 22 countries. Each country participates in this portfolio with its corresponding GDP weight. The fourth column lists portfolio value in each equity index.

This equity portfolio has 34 risk factors (22 equity indices and 12 foreign exchange rates), so we optimised over a 34-dimensional ellipsoid. Figure 4.2 shows the shape of the equity portfolio function around the worst case scenario, as a function of the Nikkei 225 and the S&P 100, the two most important risk factors. In these two risk factors the equity portfolio value function is linear, but as a whole the portfolio function is not linear, since the value of foreign equity is the product of the equity price in local currency multiplied by the exchange rate.

Country	GDP (bil. USD)	Portfolio weight (%)	Portfolio value on 15-May-2002
Canada	694.5	2.82	282 317.07
Mexico	617.4	2.51	250 975.61
United States	10 143.2	41.23	4 123 252.03
Australia	366.2	1.49	148 861.79
Japan	4 141.4	16.83	1 683 495.93
Korea	422.2	1.72	171 626.02
New Zealand	50.2	0.20	20 406.50
Austria	188.5	0.77	76 626.02
Belgium	229.6	0.93	93 333.33
Denmark	161.5	0.66	65 650.41
Finland	120.9	0.49	49 146.34
France	1 309.8	5.32	532 439.02
Germany	1 846.1	7.50	750 447.15
Ireland	103.3	0.42	41 991.87
Italy	1 088.8	4.43	442 601.63
Netherlands	380.1	1.55	154 512.20
Norway	163.7	0.67	66 544.72
Portugal	109.8	0.45	44 634.15
Spain	581.8	2.37	236 504.07
Sweden	209.8	0.85	85 284.55
Switzerland	247.1	1.00	100 447.15
United Kingdom	1 424.1	5.79	578 902.44
Total	24 600.0	100.00	10,000 000.00

Table 4.1: OECD equity index construction based on GDP in 2001. Source of GDP data: [18].

4.2 Test Results for the Option Portfolio

Multilevel Coordinate Search

As a benchmark in our tests we used the MCS algorithm by Huyer and Neumaier. Table 4.2 lists the results of the MCS algorithm on the option portfolio. Here `local` is the maximum number of portfolio evaluations in a local search and `gamma` is a parameter for the precision of local search part of the algorithm. We took `gamma` as constant and equal to the machine accuracy (variable `eps` in Matlab). We tried various values for `local` as listed in the first column. The second column shows the realised losses in worst case scenarios identified by MCS with respective values for its parameters. The third column contains the maximum losses as percentages of the current portfolio value, the fourth column shows the times used to find the optima at our test computer. In the fifth column we see the number of portfolio

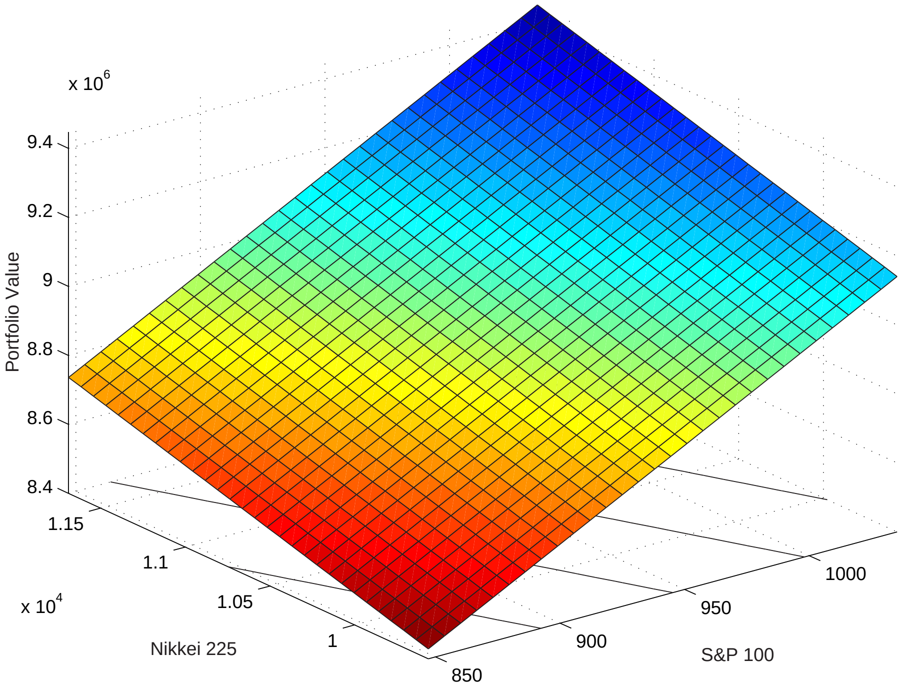

Figure 4.2: Value of the equity portfolio as a function of the Nikkei 225 and S&P 100 indices—two most important risk factors—around the worst case scenario

evaluations used, and in the last column the number of portfolio evaluations in local searches.

The optimal result for the option portfolio was obtained with the parameter `local` set to 30. With this setting, the MCS algorithm required the total of 2 476 portfolio evaluations to find its optimum.

Ell-MC

For the MC and QMC algorithms, we set the number of evaluations in one focusation step, N, such that the total number of evaluations in a search was always approximately equal to the number of portfolio evaluations used by the MCS algorithm, N_{total}:

$$N = \text{round}\left(\frac{N_{total}}{M}\right), \tag{4.1}$$

where M is the number of focusation steps, and the fraction is rounded to the nearest integer. In particular, for the option portfolio $N_{total} = 2\ 476$.

MCS(local, gamma = eps)					
	Losses		Run-times	No. of func. evals	
local	Abs. loss	Rel. loss	Time (sec.)	ncall	ncloc
0	24 038.52	2.4131%	13.41	540	-
10	62 536.23	6.2776%	12.63	635	357
20	62 598.82	6.2839%	15.91	880	602
30	**62 662.56**	**6.2903%**	**38,94**	**2 476**	**2 232**
40	62 625.29	6.2865%	38.86	2 450	2 171
50	62 625.00	6.2865%	39.22	2 474	2 231
60	62 625.67	6.2866%	39.56	2 463	2 220
70	62 625.72	6.2866%	39.47	2 472	2 230

Table 4.2: *Maximum Loss numbers of the option portfolio generated by MCS and run-times of MCS*

With this setting we could compare the algorithms fairly.

Table 4.3 shows the average results of 10 independent runs for various settings of number of focusation steps M and the shrink-factor δ of the Ell-MC algorithm. The values 20, 24 and 28 for M and 0.7, 0.8 and 0.9 for δ are based on previous tests and calibration of MiniMaxLoss with other portfolios (see [5]). The complete results of all 10 independent runs with each parameter combination are included in the appendix in Table A.1.

	δ	0.7		0.8		0.9	
M		Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
20	Avg	62 111.85	6.2350%	**62 403.83**	**6.2643%**	62 229.87	6.2468%
	StdDev	348.08	0.0349%	**428.61**	**0.0430%**	268.75	0.0270%
24	Avg	62 272.04	6.2511%	62 149.68	6.2388%	62 234.97	6.2474%
	StdDev	252.52	0.0253%	327.22	0.0328%	380.07	0.0382%
28	Avg	62 244.84	6.2483%	62 243.41	6.2482%	62 242.88	6.2482%
	StdDev	310.50	0.0312%	306.36	0.0308%	254.30	0.0255%

Table 4.3: *Averages and standard deviations of Maximum Loss numbers of the option portfolio generated by Ell-MC*

We see in Table 4.3 that we got the best result (on average) with the parameter combination $(M, \delta) = (20, 0.8)$. However, all other results were relatively near the best one, so we cannot make clear conclusions. Comparing with MCS we see that Ell-MC is a little bit worse.

Ell-QMC

For Ell-QMC we used the same parameter combinations as for the Ell-MC. Also N, the number of portfolio evaluations in one focusation step, was always the same as in the MC case for the respective value of M. Table

4.4 list the results. The complete list of all test runs can be again found in Appendix in Table A.2.

	δ	0.7		0.8		0.9	
M		Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
20	Avg	62 268.53	6.2507%	62 291.26	6.2530%	62 109.83	6.2348%
	StdDev	165.18	0.0166%	184.19	0.0185%	200.31	0.0201%
24	Avg	**62 321.66**	**6.2561%**	62 202.87	6.2441%	62 032.99	6.2271%
	StdDev	**145.92**	**0.0146%**	293.39	0.0295%	301.91	0.0303%
28	Avg	62 252.18	6.2491%	62 260.98	6.2500%	62 011.96	6.2250%
	StdDev	283.19	0.0284%	342.37	0.0344%	383.82	0.0385%

Table 4.4: *Averages and standard deviations of Maximum Loss numbers of the option portfolio generated by Ell-QMC*

Here we got the best results with the combination $(M, \delta) = (24, 0.7)$, though there are also other combinations with relatively similar values. Particularly, all combinations with $\delta = 0.7$ and $\delta = 0.8$ are within one standard deviation away from the best result. Interestingly, all combinations with $\delta = 0.9$ are more than one standard deviation away. Comparing the optimal results of Ell-MC and Ell-QMC we see that MC is a little bit better but the difference is not significant. (This might be due to the fact that in QMC only one random part of MC was replaced by a QMC method. Usually in higher dimensions the advantage of QMC is more significant.) MCS is slightly better than both.

Cube-MC

The next step is to try the fully deterministic QMC, but first we test its stochastic counterpart. The Setup is the same as for the previous two algorithms. Table 4.5 shows the results, while the full list of all test runs is in Appendix in Table A.3.

The winner was the setting $(M, \delta) = (20, 0.7)$, however, here again, all other combinations were within one standard deviation away. We also see that Cube-MC is a little bit worse than both Ell-MC and Ell-QMC (and thus also than MCS), though not significantly.

Cube-QMC

Finally, we have the fully deterministic QMC algorithm. Table 4.6 shows the results.

	δ	0.7		0.8		0.9	
M		Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
20	Avg	**62 305.29**	**6.2544%**	62 233.42	6.2472%	62 140.33	6.2379%
	StdDev	**306.22**	**0.0307%**	348.68	0.0350%	269.77	0.0271%
24	Avg	62 053.97	6.2292%	62 219.36	6.2458%	62 130.38	6.2369%
	StdDev	307.48	0.0309%	299.79	0.0301%	259.96	0.0261%
28	Avg	62 130.84	6.2369%	62 128.11	6.2366%	62 214.49	6.2453%
	StdDev	431.71	0.0433%	327.17	0.0328%	187.04	0.0188%

Table 4.5: *Averages and standard deviations of Maximum Loss numbers of the option portfolio generated by Cube-MC.*

δ	0.7		0.8		0.9	
M	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
20	61 771.25	6.2008%	**62 066.74**	**6.2305%**	61 849.86	6.2087%
24	61 772.61	6.2009%	61 762.35	6.1999%	61 582.53	6.1819%
28	61 772.92	6.2010%	61 769.31	6.2006%	61 582.53	6.1819%

Table 4.6: *Maximum Loss numbers of the option portfolio generated by Cube-QMC.*

Surprisingly, the results were relatively poor compared to other the algorithms. Only one combination, $(M, \delta) = (20, 0.7)$, gave us a comparable result to the MC counterpart.

MC Searching on the Surface of the Ellipsoid

Finally we tested the MC and QMC algorithms with searching on a surface of the ellipsoid. Table 4.7 shows the results of both algorithms, while the full table is in Appendix in Table A.9.

	$\text{MC}_{\text{surface}}$		$\text{QMC}_{\text{surface}}$	
	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
Avg	50 480.48	5.0674%	50 689.20	5.0884%
StdDev	847.35	0.0851%	-	-

Table 4.7: Maximum Loss numbers of the option portfolio generated by Surf-MC and Surf-QMC

We see that this algorithms cannot compete with the algorithms using the zoom-in procedure. But comparing the QMC and the MC version we see that QMC outperformed MC, though only slightly.

4.3 Test Results on Equity Portfolio

Multilevel Coordinate Search

Table 4.8 shows the results of the MCS algorithm on the equity portfolio. The meaning of columns is the same as in the option portfolio case. Also for the equity portfolio we chose 30 as the best setting for the maximum number of evaluations in a local search. Although this setting was not the overall winner, we did a trade-off between the result's improvement and the evaluation time. With `local` = 30, the MCS needed 25 156 portfolio evaluations to find its optimum. We note that also settings with `local` equal to 20 or even 10 are relatively good taking into account low number of portfolio evaluations and thus accordingly shorter search-time.

MCS(local, gamma = eps)					
	Losses		Run-times	No. of func. evals	
local	Abs. loss	Rel. loss	Time (sec.)	ncall	ncloc
0	1 027 166.77	10.2717%	386.20	7 076	-
10	1 058 642.81	10.5864%	452.27	13 702	6 750
20	1 058 712.29	10.5871%	514.83	19 692	12 740
30	**1 058 737.50**	**10.5874%**	**574.25**	**25 156**	**18 204**
40	1 058 744.52	10.5874%	630.33	30 621	23 669
50	1 058 749.30	10.5875%	696.23	36 083	29 131
60	1 058 753.48	10.5875%	756.39	42 074	35 122
70	1 058 757.35	10.5876%	821.98	48 065	41 113

Table 4.8: *Maximum Loss numbers of the equity portfolio generated by MCS and run-times of MCS*

Ell-MC

For the equity portfolio we used the same procedure of setting N (the number of portfolio evaluations in one focusation step) as in the option portfolio-case (see equation (4.1)). The total number of portfolio evaluations was thus approximately (due to rounding errors) 25 156. Table 4.9 shows the results. The full list of results of all test runs can be again seen in Appendix in Table A.4.

Here the situation is quite different as in the option portfolio-case. We got the optimal result with $(M, \delta) = (28, 0.9)$, and only one other combination, $(24, 0.8)$, was within one standard deviation away. It is also significantly better than the MCS result.

	δ	0.7		0.8		0.9	
M		Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
20	Avg	1 034 219.43	10.3422%	1 061 187.61	10.6119%	1 051 968.98	10.5197%
	StdDev	5 356.13	0.0536%	2 369.73	0.0237%	1 942.56	0.0194%
24	Avg	1 025 918.05	10.2592%	1 062 188.25	10.6219%	1 059 161.05	10.5916%
	StdDev	12 734.98	0.1273%	2 037.42	0.0204%	1 459.03	0.0146%
28	Avg	1 026 778.44	10.2678%	1 061 256.85	10.6126%	**1 062 764.77**	**10.6276%**
	StdDev	12 559.01	0.1256%	1 760.16	0.0176%	**873.43**	**0.0087%**

Table 4.9: *Averages and standard deviations of Maximum Loss numbers of the equity portfolio generated by Ell-MC*

Ell-QMC

The results of the QMC version of the previous algorithm are in Table 4.10. The full table is again in Appendix in Table A.5.

	δ	0.7		0.8		0.9	
M		Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
20	Avg	1 027 389.89	10.2739%	1 057 640.63	10.5764%	1 048 110.28	10.4811%
	StdDev	8 244.35	0.0824%	3 470.15	0.0347%	4 190.69	0.0419%
24	Avg	1 022 521.93	10.2252%	1 055 237.20	10.5524%	1 056 071.96	10.5607%
	StdDev	9 323.02	0.0932%	5 751.58	0.0575%	2 723.71	0.0272%
28	Avg	1 020 212.62	10.2021%	1 052 821.11	10.5282%	**1 061 327.69**	**10.6133%**
	StdDev	7 706.72	0.0771%	7 675.02	0.0768%	**1 100.46**	**0.0110%**

Table 4.10: *Averages and standard deviations of Maximum Loss numbers of the equity portfolio generated by Ell-QMC*

Here the difference from the option portfolio-case is even more stressed. There is no result within one standard deviation from the optimal one, which we obtained with the $(28, 0.9)$ setting. Interesting is also the fact that we got our best result in the opposite side of the table than with the option portfolio. For the option portfolio, $\delta = 0.7$ was a relatively good choice, while for the equity portfolio this value for the shrink factor gave us quite poor results while $\delta = 0.9$ with $M = 28$ was the winner.

Cube-MC

The results of Cube-MC are in Table 4.11, or in Appendix in A.6 (the full version).

The winner is again the setting $(28, 0.9)$ with two other settings, $(24, 0.8)$ and $(28, 0.8)$, within one standard deviation away. Here again the choice of $\delta = 0.7$ gave us rather poor results.

	δ	0.7		0.8		0.9	
M		Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
20	Avg	1 031 922.22	10.3192%	1 060 842.23	10.6084%	1 048 770.61	10.4877%
	StdDev	8 417.10	0.0842%	1 447.74	0.0145%	2 748.38	0.0275%
24	Avg	1 025 593.29	10.2559%	1 062 398.99	10.6240%	1 058 008.66	10.5801%
	StdDev	6 274.32	0.0627%	1 333.53	0.0133%	1 594.11	0.0159%
28	Avg	1 024 203.42	10.2420%	1 061 640.33	10.6164%	**1 062 558.43**	**10.6256%**
	StdDev	9 601.57	0.0960%	1 928.16	0.0193%	**1 235.91**	**0.0124%**

Table 4.11: *Averages and standard deviations of Maximum Loss numbers of the equity portfolio generated by Cube-MC*

Cube-QMC

Table 4.12 shows the results of QMC version of the previous algorithm.

δ	0.7		0.8		0.9	
M	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
20	1 028 004.41	10.2800%	1 044 633.28	10.4463%	1 046 639.71	10.4664%
24	1 026 114.52	10.2611%	1 058 713.09	10.5871%	1 058 257.10	10.5826%
28	1 025 315.02	10.2532%	1 057 969.07	10.5797%	**1 061 572.47**	**10.6157%**

Table 4.12: *Results on the equity portfolio generated by Cube-QMC*

Here once more the choice of $(28, 0.9)$ was the winning setting, though the result of MC were on average better. The choice of $\delta = 0.7$ was again not very good.

Surf-MC

We tested the MC and QMC with searching on the surface of the ellipsoid also on the equity portfolio. The results are in Table 4.13, while Table A.10 in Appendix is the full version.

	$\text{MC}_{\text{surface}}$		$\text{QMC}_{\text{surface}}$	
	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
Avg	96 009.00	0.9601%	107 943.01	1.0794%
StdDev	6 324.75	0.0632%	-	-

Table 4.13: *Maximum Loss numbers of the equity portfolio generated by Surf-MC and Surf-QMC.*

Here the output is even worse than with the option portfolio, showing that searching on a surface without focusation is not a good idea. Again, the good news is that the QMC outperformed the MC.

Searching in the Interior without Focusation

Observing the fact that with the equity portfolio the optimal shrink factor was always 0.9, we tried to turn the focusation off while still searching in the interior of the elliptic domain. We chose $\delta = 1$ and $M = 1$. The aggregated results of the MC and QMC algorithms with OSPHERE and mapping from a cube to an ellipsoid on the option portfolio are in Table 4.14, while the full table is again in Appendix in Table A.7.

	$\text{MC}_{\text{osphere}}$		$\text{QMC}_{\text{osphere}}$	
	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
Avg	61 806.27	6.2043%	61 849.38	6.2087%
StdDev	410.78	0.0412%	402.28	0.0404%

	MC_{cube}		QMC_{cube}	
	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
Avg	61 857.90	6.2095%	62 085.72	6.2324%
StdDev	395.21	0.0397%	-	-

Table 4.14: Maximum Loss numbers of the option portfolio generated by Ell-MC, Ell-QMC, Cube-MC, and Cube-QMC without zooming-in

In this tables we also see that the QMC algorithms outperformed the MC ones though not significantly. Running these tests for the equity portfolio, the results were as in the Table 4.15 the full version of which we see in Table A.8 in Appendix. We see that these results again cannot compete with the focusing algorithms. The QMC algorithms here again outperformed the MC ones, especially Ell-QMC was much better than Ell-MC.

	$\text{MC}_{\text{osphere}}$		$\text{QMC}_{\text{osphere}}$	
	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
Avg	677 353.44	6.7735%	759 179.90	7.5918%
StdDev	34 512.09	0.3451%	17 077.32	0.1708%

	MC_{cube}		QMC_{cube}	
	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
Avg	634 577.74	6.3458%	697 073.09	6.9707%
StdDev	12 770.34	0.1277%	-	-

Table 4.15: Maximum Loss numbers of the equity portfolio generated by MC and QMC with OSPHERE and with mapping from a cube to an ellipsoid without focusation

4.4 Is Quasi-Monte Carlo better than Monte Carlo?

The question of primary interest is whether QMC outperforms MC. To investigate this question we considerend Cube-MC and Cube-QMC because here QMC is fully deterministic. Since Cube-MC delivers random results the question has to be decided by statistical test procedures. The null hypothesis was that QMC is better than MC (on average). The test statistics is

$$Z = (\overline{\mathrm{ML}}_{\mathrm{MC}} - \mathrm{ML}_{\mathrm{QMC}})\sqrt{m}/s,$$

where $\overline{\mathrm{ML}}_{\mathrm{MC}}$ is the average of Maximum Loss results produced in the sample of MC runs, $\mathrm{ML}_{\mathrm{QMC}}$ is the Maximum Loss produced by QMC, m is the size of the sample, and s is the standard deviation of the sample. We tested the null hypothesis at a confidence level of 1%. The resulting rejection level for this one-sided null hypothesis is 2.82. For a value of Z above 2.82 MC is significantly better (with 99% probability) than QMC and the null hypothesis has to be rejected.

Results of the statistical tests of QMC and the MC on the option portfolio for various parameter combinations are summarised in Table 4.16. Results for the equity portfolio are in Table 4.17.

With the option portfolio, the null hypothesis can be rejected for the 7 parameter settings out of 10. In all these 7 settings MC performs significantly better than QMC. With the equity portfolio, MC performs significantly better only in 3 out of 10 cases. For the 7-dimensional option portfolio MC and QMC are not quite as good as MCS, but for the 34-dimensional equity portfolio, for both QMC and MC there are some parameter settings at which they outperform MCS.

Note also that the zooming-in procedure improves results of MC and QMC for higher dimensional problems enormously. For the 7-dimensional option portfolio, QMC performs best without focusation, MC does not perform significantly better with focusation than without focusation (see Table 4.16). In contrast, for the 34-dimensional equity portfolio, both MC and QMC perform significantly better with focusation than without focusation: Maximum Losses are only $6-7\%$ without focusation, but $10-11\%$ with focusation.

	δ	0.7	0.8	0.9	1.0
M					
1	MC-Avg				61 857.90
	MC-StdDev				395.21
	QMC				62 085.72
	Z				-1.82
	QMC better				not rejected
20	MC-Avg	62 305.29	62 233.42	62 140.33	
	MC-StdDev	306.22	348.68	269.77	
	QMC	61 771.25	62 066.74	61 849.86	
	Z	5.51	1.51	3.40	
	QMC better	rejected	not rejected	rejected	
24	MC-Avg	62 053.97	62 219.36	62 130.38	
	MC-StdDev	307.48	299.79	259.96	
	QMC	61 772.61	61 762.35	61 582.53	
	Z	2.89	4.82	6.66	
	QMC better	rejected	rejected	rejected	
28	MC-Avg	62 130.84	62 128.11	62 214.49	
	MC-StdDev	431.71	327.17	187.04	
	QMC	61 772.92	61 769.31	61 582.53	
	Z	2.62	3.47	10.68	
	QMC better	not rejected	rejected	rejected	

Table 4.16: *Maximum loss of the option portfolio produced by Cube-MC and Cube-QMC. Statistics of MC results from 10 independent runs with 2,476 evaluations each. Confidence level for rejection of null hypothesis 1%. Rejection level for test statistics 2.82.*

	δ	0.7	0.8	0.9	1.0
M					
1	MC-Avg				634 577.74
	MC-StdDev				12 770.34
	QMC				697 073.09
	Z				-15.48
	QMC better				not rejected
20	MC-Avg	1 031 922.22	1 060 842.23	1 048 770.61	
	MC-StdDev	8 417.10	1 447.74	2 748.38	
	QMC	1 028 004.41	1 044 633.28	1 046 639.71	
	Z	1.47	35.40	2.45	
	QMC better	not rejected	rejected	not rejected	
24	MC-Avg	1 025 593.29	1 062 398.99	1 058 008.66	
	MC-StdDev	6 274.32	1 333.53	1 594.11	
	QMC	1 026 114.52	1 058 713.09	1 058 257.10	
	Z	-0.26	8.74	-0.49	
	QMC better	not rejected	rejected	not rejected	
28	MC-Avg	1 024 203.42	1 061 640.33	1 062 558.43	
	MC-StdDev	9 601.57	1 928.16	1 235.91	
	QMC	1 025 315.02	1 057 969.07	1 061 572.47	
	Z	-0.37	6.02	2.52	
	QMC better	not rejected	rejected	not rejected	

Table 4.17: *Maximum relative loss of the equity portfolio produced by Cube-MC and Cube-QMC. Statistics of MC results from 10 independent runs with 25,156 evaluations each. Confidence level for rejection of null hypothesis 1%. Rejection level for test statistics 2.82.*

Chapter 5

Outlook

In view of the successful application of QMC methods in numerical integration it might be surprising that for the focusing optimisation algorithms of this paper QMC is not significantly better than MC. One reason might be that the algorithms produce sequences of scenarios in the ellipsoid by transforming sequences from the cube. This transformation map h of equation (3.4) partially destroys the low-discrepancy properties of a sequence. This is illustrated in Figure 5.1: The original two-dimensional Niederreiter-Xing sequence (represented by dots) is evenly distributed through the cube, the transformed sequence (crosses) clusters near the surface of the ellipsoid.

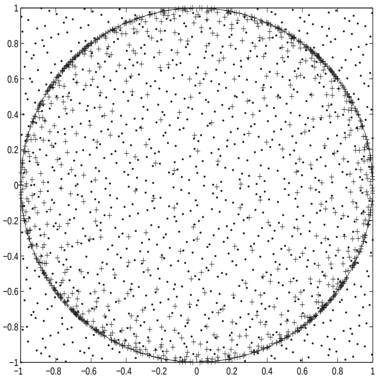

Figure 5.1: The transformation h of eq. (3.4) of points in a cube to points in an ellipsoid can destroy the low-discrepancy property of sequences in the cube. The original two-dimensional Niederreiter-Xing sequence (represented by dots) is evenly distributed through the cube, the transformed sequence (crosses) clusters near the surface of the ellipsoid.

This phenomenon has particularly bad effects for focusing search procedures. In later focusation steps the scenarios cluster on the part of the

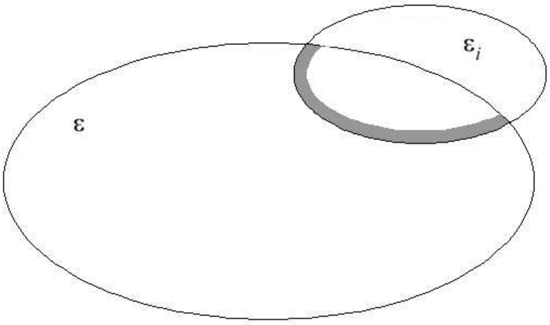

Figure 5.2: The algorithms MC and QMC generated scenarios clustered on the part of the shrinked surface $\mathcal{E}_i$ which is inside the original ellipsoid $\mathcal{E}$. It would be desirable to generate many scenarios in the intersection of the surface of the original ellipsoid $\mathcal{E}$ with the shrinked ellipsoid $\mathcal{E}_i$ instead.

shrinked surface $\mathcal{E}_i$ $(i = 2, \ldots, M)$ which is inside the original ellipsoid $\mathcal{E}$. This region is shaded in Figure 5.2. Instead it would be desirable to generate scenarios, which are evenly distributed over the volume of the intersection of the original ellipsoid $\mathcal{E}$ with the shrinked ellipsoid $\mathcal{E}_i$. Taking into account that for security portfolios the worst case scenarios are generically on the surface of the ellipsoid $\mathcal{E}$, even better results could be expected of QMC algorithms which produce sequences evenly distributed on shrinking regions of the surface of $\mathcal{E}$.

Appendix A

Detailed Tables of Test Results

	δ	0.7		0.8		0.9	
M	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
20	1	61 696.72	6.1933%	62 130.08	6.2368%	61 782.92	6.2020%
	2	62 420.28	6.2660%	62 794.98	6.3036%	62 398.81	6.2638%
	3	61 982.91	6.2221%	62 705.45	6.2946%	62 132.78	6.2371%
	4	62 312.90	6.2552%	62 363.52	6.2603%	62 405.60	6.2645%
	5	62 028.99	6.2267%	62 666.17	6.2906%	62 210.42	6.2449%
	6	61 861.73	6.2099%	61 395.93	6.1631%	62 257.50	6.2496%
	7	61 592.54	6.1829%	62 745.10	6.2986%	62 571.49	6.2811%
	8	62 247.00	6.2486%	62 692.45	6.2933%	62 366.03	6.2605%
	9	62 738.73	6.2979%	62 355.26	6.2594%	62 405.89	6.2645%
	10	62 236.66	6.2475%	62 189.35	6.2428%	61 767.21	6.2004%
	Avg	62 111.85	6.2350%	62 403.83	6.2643%	62 229.87	6.2468%
	StdDev	348.08	0.0349%	428.61	0.0430%	268.75	0.0270%
	Min	61 592.54	6.1829%	61 395.93	6.1631%	61 767.21	6.2004%
	Max	62 738.73	6.2979%	62 794.98	6.3036%	62 571.49	6.2811%
	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
24	1	62 139.85	6.2378%	62 623.60	6.2864%	61 826.58	6.2064%
	2	61 810.15	6.2047%	62 218.20	6.2457%	62 399.12	6.2638%
	3	62 367.47	6.2607%	62 347.54	6.2587%	62 582.74	6.2823%
	4	61 920.22	6.2158%	61 952.06	6.2190%	62 717.90	6.2958%
	5	62 581.40	6.2821%	62 509.37	6.2749%	61 618.01	6.1854%
	6	62 537.77	6.2778%	61 838.93	6.2076%	61 894.55	6.2132%
	7	62 268.30	6.2507%	61 808.22	6.2045%	62 358.95	6.2598%
	8	62 392.20	6.2631%	61 626.17	6.1862%	62 600.32	6.2840%
	9	62 443.18	6.2683%	62 253.30	6.2492%	61 952.41	6.2190%
	10	62 259.85	6.2499%	62 319.41	6.2558%	62 399.11	6.2638%
	Avg	62 272.04	6.2511%	62 149.68	6.2388%	62 234.97	6.2474%
	StdDev	252.52	0.0253%	327.22	0.0328%	380.07	0.0382%
	Min	61 810.15	6.2047%	61 626.17	6.1862%	61 618.01	6.1854%
	Max	62 581.40	6.2821%	62 623.60	6.2864%	62 717.90	6.2958%
	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
28	1	62 622.13	6.2862%	62 208.82	6.2447%	62 750.84	6.2991%
	2	62 047.31	6.2285%	61 957.37	6.2195%	62 072.78	6.2311%
	3	61 820.79	6.2058%	62 044.79	6.2283%	62 164.96	6.2403%
	4	62 503.23	6.2743%	61 916.87	6.2154%	62 296.11	6.2535%
	5	62 234.12	6.2473%	62 401.43	6.2641%	62 067.83	6.2306%
	6	62 066.17	6.2304%	62 649.45	6.2890%	61 920.88	6.2158%
	7	62 779.59	6.3020%	62 293.42	6.2532%	62 312.00	6.2551%
	8	62 228.96	6.2468%	61 879.21	6.2116%	62 565.37	6.2805%
	9	61 896.36	6.2134%	62 303.58	6.2542%	62 035.39	6.2273%
	10	62 249.75	6.2488%	62 779.16	6.3020%	62 242.62	6.2481%
	Avg	62 244.84	6.2483%	62 243.41	6.2482%	62 242.88	6.2482%
	StdDev	310.50	0.0312%	306.36	0.0308%	254.30	0.0255%
	Min	61 820.79	6.2058%	61 879.21	6.2116%	61 920.88	6.2158%
	Max	62 779.59	6.3020%	62 779.16	6.3020%	62 750.84	6.2991%

Table A.1: Losses of the option portfolio generated by Ell-MC

	δ	0.7		0.8		0.9	
M	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
20	1	62 420.14	6.2659%	62 316.34	6.2555%	62 055.04	6.2293%
	2	62 408.96	6.2648%	62 344.37	6.2583%	62 001.37	6.2239%
	3	62 452.46	6.2692%	62 385.53	6.2625%	62 136.23	6.2374%
	4	62 133.41	6.2372%	61 913.49	6.2151%	62 189.14	6.2428%
	5	62 077.72	6.2316%	62 385.49	6.2625%	62 277.27	6.2516%
	6	62 332.87	6.2572%	62 383.54	6.2623%	61 701.70	6.1938%
	7	62 006.68	6.2244%	62 058.65	6.2297%	62 318.84	6.2558%
	8	62 366.51	6.2606%	62 186.85	6.2425%	62 041.54	6.2279%
	9	62 119.29	6.2357%	62 517.08	6.2757%	61 983.65	6.2221%
	10	62 367.31	6.2606%	62 421.26	6.2661%	62 393.53	6.2633%
	Avg	62 268.53	6.2507%	62 291.26	6.2530%	62 109.83	6.2348%
	StdDev	165.18	0.0166%	184.19	0.0185%	200.31	0.0201%
	Min	62 006.68	6.2244%	61 913.49	6.2151%	61 701.70	6.1938%
	Max	62 452.46	6.2692%	62 517.08	6.2757%	62 393.53	6.2633%
	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
24	1	62 206.61	6.2445%	62 435.23	6.2675%	62 060.87	6.2299%
	2	62 372.26	6.2611%	62 289.34	6.2528%	62 417.55	6.2657%
	3	62 568.30	6.2808%	61 491.51	6.1727%	62 092.43	6.2330%
	4	62 146.94	6.2385%	62 284.87	6.2524%	62 321.66	6.2561%
	5	62 308.93	6.2548%	62 140.29	6.2379%	61 854.56	6.2092%
	6	62 120.55	6.2359%	62 016.72	6.2254%	62 247.48	6.2486%
	7	62 308.53	6.2547%	62 200.77	6.2439%	61 661.69	6.1898%
	8	62 512.23	6.2752%	62 223.55	6.2462%	62 149.32	6.2388%
	9	62 277.15	6.2516%	62 385.59	6.2625%	62 085.39	6.2323%
	10	62 395.07	6.2634%	62 560.80	6.2801%	61 439.00	6.1675%
	Avg	62 321.66	6.2561%	62 202.87	6.2441%	62 032.99	6.2271%
	StdDev	145.92	0.0146%	293.39	0.0295%	301.91	0.0303%
	Min	62 120.55	6.2359%	61 491.51	6.1727%	61 439.00	6.1675%
	Max	62 568.30	6.2808%	62 560.80	6.2801%	62 417.55	6.2657%
	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
28	1	61 898.88	6.2136%	62 437.65	6.2677%	62 568.98	6.2809%
	2	62 518.31	6.2758%	62 543.31	6.2783%	61 897.17	6.2134%
	3	62 204.51	6.2443%	62 219.05	6.2458%	62 437.75	6.2677%
	4	62 335.18	6.2574%	62 654.04	6.2894%	61 715.59	6.1952%
	5	61 722.96	6.1960%	61 941.84	6.2179%	61 992.48	6.2230%
	6	62 339.15	6.2578%	61 692.25	6.1929%	61 505.31	6.1741%
	7	62 441.24	6.2681%	61 779.09	6.2016%	62 456.51	6.2696%
	8	62 579.76	6.2820%	62 330.72	6.2570%	62 096.23	6.2334%
	9	62 448.27	6.2688%	62 557.98	6.2798%	61 955.58	6.2193%
	10	62 033.50	6.2271%	62 453.84	6.2693%	61 493.97	6.1730%
	Avg	62 252.18	6.2491%	62 260.98	6.2500%	62 011.96	6.2250%
	StdDev	283.19	0.0284%	342.37	0.0344%	383.82	0.0385%
	Min	61 722.96	6.1960%	61 692.25	6.1929%	61 493.97	6.1730%
	Max	62 579.76	6.2820%	62 654.04	6.2894%	62 568.98	6.2809%

Table A.2: Losses of the option portfolio generated by Ell-QMC

	δ	0.7		0.8		0.9	
M	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
20	1	62 411.08	6.2650%	62 651.27	6.2891%	62 180.35	6.2419%
	2	62 759.65	6.3000%	62 328.71	6.2568%	62 188.11	6.2427%
	3	62 257.57	6.2496%	62 286.08	6.2525%	62 353.21	6.2592%
	4	62 212.88	6.2451%	62 383.88	6.2623%	61 958.31	6.2196%
	5	62 697.73	6.2938%	61 745.52	6.1982%	61 905.69	6.2143%
	6	61 740.10	6.1977%	62 442.24	6.2682%	62 442.26	6.2682%
	7	62 175.21	6.2414%	61 521.39	6.1757%	62 262.19	6.2501%
	8	62 547.24	6.2787%	62 409.36	6.2649%	61 543.21	6.1779%
	9	62 157.70	6.2396%	62 457.01	6.2696%	62 203.77	6.2442%
	10	62 093.71	6.2332%	62 108.77	6.2347%	62 366.19	6.2605%
	Avg	62 305.29	6.2544%	62 233.42	6.2472%	62 140.33	6.2379%
	StdDev	306.22	0.0307%	348.68	0.0350%	269.77	0.0271%
	Min	61 740.10	6.1977%	61 521.39	6.1757%	61 543.21	6.1779%
	Max	62 759.65	6.3000%	62 651.27	6.2891%	62 442.26	6.2682%
	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
24	1	62 378.92	6.2618%	62 265.15	6.2504%	62 622.99	6.2863%
	2	62 168.95	6.2407%	61 845.05	6.2082%	61 952.34	6.2190%
	3	61 937.21	6.2175%	62 141.66	6.2380%	62 160.99	6.2399%
	4	61 609.07	6.1845%	62 473.85	6.2713%	62 287.12	6.2526%
	5	62 411.38	6.2651%	62 237.36	6.2476%	62 108.15	6.2346%
	6	62 205.29	6.2444%	62 252.92	6.2492%	62 166.22	6.2405%
	7	62 165.82	6.2404%	61 969.41	6.2207%	61 865.20	6.2102%
	8	61 701.79	6.1938%	62 757.87	6.2998%	61 985.13	6.2223%
	9	61 647.04	6.1883%	61 790.35	6.2027%	62 401.59	6.2641%
	10	62 314.23	6.2553%	62 459.97	6.2699%	61 754.06	6.1991%
	Avg	62 053.97	6.2292%	62 219.36	6.2458%	62 130.38	6.2369%
	StdDev	307.48	0.0309%	299.79	0.0301%	259.96	0.0261%
	Min	61 609.07	6.1845%	61 790.35	6.2027%	61 754.06	6.1991%
	Max	62 411.38	6.2651%	62 757.87	6.2998%	62 622.99	6.2863%
	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
28	1	62 498.44	6.2738%	62 620.95	6.2861%	62 404.26	6.2644%
	2	61 925.47	6.2163%	62 543.73	6.2784%	62 205.70	6.2444%
	3	61 730.78	6.1967%	61 828.95	6.2066%	62 080.05	6.2318%
	4	62 569.13	6.2809%	61 954.13	6.2192%	62 376.00	6.2615%
	5	62 247.37	6.2486%	62 383.81	6.2623%	62 023.41	6.2261%
	6	61 490.49	6.1726%	61 761.91	6.1999%	61 950.44	6.2188%
	7	62 479.46	6.2719%	62 167.48	6.2406%	62 299.37	6.2538%
	8	62 583.10	6.2823%	62 349.31	6.2588%	62 095.77	6.2334%
	9	62 269.03	6.2508%	61 892.95	6.2130%	62 538.83	6.2779%
	10	61 515.12	6.1751%	61 777.85	6.2015%	62 171.07	6.2409%
	Avg	62 130.84	6.2369%	62 128.11	6.2366%	62 214.49	6.2453%
	StdDev	431.71	0.0433%	327.17	0.0328%	187.04	0.0188%
	Min	61 490.49	6.1726%	61 761.91	6.1999%	61 950.44	6.2188%
	Max	62 583.10	6.2823%	62 620.95	6.2861%	62 538.83	6.2779%

Table A.3: Losses of the option portfolio generated by Cube-MC

	δ	0.7		0.8		0.9	
M	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
20	1	1 039 696.13	10.3970%	1 063 128.74	10.6313%	1 053 017.84	10.5302%
	2	1 031 491.68	10.3149%	1 062 016.95	10.6202%	1 054 941.36	10.5494%
	3	1 035 962.37	10.3596%	1 061 289.76	10.6129%	1 051 675.81	10.5168%
	4	1 024 755.07	10.2476%	1 059 695.03	10.5970%	1 052 059.71	10.5206%
	5	1 037 871.34	10.3787%	1 055 573.24	10.5557%	1 051 591.34	10.5159%
	6	1 034 809.90	10.3481%	1 061 610.62	10.6161%	1 054 099.10	10.5410%
	7	1 031 801.98	10.3180%	1 062 935.14	10.6294%	1 051 202.15	10.5120%
	8	1 036 788.11	10.3679%	1 062 280.79	10.6228%	1 048 564.91	10.4856%
	9	1 027 372.64	10.2737%	1 059 745.88	10.5975%	1 053 014.69	10.5301%
	10	1 041 645.09	10.4165%	1 063 599.92	10.6360%	1 049 522.89	10.4952%
	Avg	1 034 219.43	10.3422%	1 061 187.61	10.6119%	1 051 968.98	10.5197%
	StdDev	5 356.13	0.0536%	2 369.73	0.0237%	1 942.56	0.0194%
	Min	1 024 755.07	10.2476%	1 055 573.24	10.5557%	1 048 564.91	10.4856%
	Max	1 041 645.09	10.4165%	1 063 599.92	10.6360%	1 054 941.36	10.5494%
	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
24	1	1 011 305.57	10.1131%	1 064 917.67	10.6492%	1 062 012.49	10.6201%
	2	1 042 386.59	10.4239%	1 061 003.99	10.6100%	1 058 852.01	10.5885%
	3	1 036 788.47	10.3679%	1 063 382.19	10.6338%	1 060 327.54	10.6033%
	4	1 025 210.90	10.2521%	1 062 327.51	10.6233%	1 058 247.85	10.5825%
	5	1 031 433.80	10.3143%	1 057 252.17	10.5725%	1 058 937.34	10.5894%
	6	1 011 191.42	10.1119%	1 061 975.15	10.6198%	1 058 805.97	10.5881%
	7	1 005 247.38	10.0525%	1 062 560.16	10.6256%	1 058 830.45	10.5883%
	8	1 038 400.57	10.3840%	1 061 928.14	10.6193%	1 059 930.53	10.5993%
	9	1 026 224.97	10.2622%	1 063 508.30	10.6351%	1 059 305.20	10.5931%
	10	1 030 990.77	10.3099%	1 063 027.24	10.6303%	1 056 361.09	10.5636%
	Avg	1 025 918.05	10.2592%	1 062 188.25	10.6219%	1 059 161.05	10.5916%
	StdDev	12 734.98	0.1273%	2 037.42	0.0204%	1 459.03	0.0146%
	Min	1 005 247.38	10.0525%	1 057 252.17	10.5725%	1 056 361.09	10.5636%
	Max	1 042 386.59	10.4239%	1 064 917.67	10.6492%	1 062 012.49	10.6201%
	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
28	1	1 031 590.17	10.3159%	1 063 380.69	10.6338%	1 062 441.74	10.6244%
	2	1 016 183.36	10.1618%	1 059 445.12	10.5945%	1 063 373.29	10.6337%
	3	1 015 783.21	10.1578%	1 063 035.35	10.6304%	1 062 043.48	10.6204%
	4	1 043 233.25	10.4323%	1 060 023.39	10.6002%	1 063 518.63	10.6352%
	5	1 029 240.26	10.2924%	1 060 311.35	10.6031%	1 063 896.50	10.6390%
	6	1 021 948.33	10.2195%	1 059 289.32	10.5929%	1 063 404.32	10.6340%
	7	1 026 160.94	10.2616%	1 059 856.49	10.5986%	1 061 142.27	10.6114%
	8	1 004 444.66	10.0444%	1 061 379.78	10.6138%	1 063 371.21	10.6337%
	9	1 035 408.76	10.3541%	1 061 676.65	10.6168%	1 062 363.54	10.6236%
	10	1 043 791.47	10.4379%	1 064 170.33	10.6417%	1 062 092.70	10.6209%
	Avg	1 026 778.44	10.2678%	1 061 256.85	10.6126%	1 062 764.77	10.6276%
	StdDev	12 559.01	0.1256%	1 760.16	0.0176%	873.43	0.0087%
	Min	1 004 444.66	10.0444%	1 059 289.32	10.5929%	1 061 142.27	10.6114%
	Max	1 043 791.47	10.4379%	1 064 170.33	10.6417%	1 063 896.50	10.6390%

Table A.4: Losses of the equity portfolio generated by Ell-MC

	δ	0.7		0.8		0.9	
M	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
20	1	1 016 443.42	10.1644%	1 056 835.14	10.5684%	1 055 032.84	10.5503%
	2	1 023 309.07	10.2331%	1 050 993.51	10.5099%	1 041 795.32	10.4180%
	3	1 021 898.97	10.2190%	1 061 202.39	10.6120%	1 047 835.28	10.4784%
	4	1 034 668.29	10.3467%	1 058 213.46	10.5821%	1 048 257.43	10.4826%
	5	1 038 810.58	10.3881%	1 062 506.99	10.6251%	1 047 252.63	10.4725%
	6	1 025 835.24	10.2584%	1 056 758.11	10.5676%	1 053 689.05	10.5369%
	7	1 025 807.79	10.2581%	1 053 263.72	10.5326%	1 042 975.33	10.4298%
	8	1 016 762.17	10.1676%	1 058 750.36	10.5875%	1 049 573.42	10.4957%
	9	1 031 835.31	10.3184%	1 057 869.51	10.5787%	1 045 269.27	10.4527%
	10	1 038 528.05	10.3853%	1 060 013.12	10.6001%	1 049 422.20	10.4942%
	Avg	1 027 389.89	10.2739%	1 057 640.63	10.5764%	1 048 110.28	10.4811%
	StdDev	8 244.35	0.0824%	3 470.15	0.0347%	4 190.69	0.0419%
	Min	1 016 443.42	10.1644%	1 050 993.51	10.5099%	1 041 795.32	10.4180%
	Max	1 038 810.58	10.3881%	1 062 506.99	10.6251%	1 055 032.84	10.5503%
	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
24	1	1 021 939.59	10.2194%	1 056 958.27	10.5696%	1 058 935.81	10.5894%
	2	1 037 511.32	10.3751%	1 047 465.09	10.4747%	1 057 173.41	10.5717%
	3	1 021 469.82	10.2147%	1 057 255.01	10.5726%	1 052 963.45	10.5296%
	4	1 014 085.41	10.1409%	1 057 625.46	10.5763%	1 058 578.66	10.5858%
	5	1 022 422.11	10.2242%	1 061 305.27	10.6131%	1 056 464.08	10.5646%
	6	1 027 569.93	10.2757%	1 056 708.90	10.5671%	1 057 267.21	10.5727%
	7	1 013 306.45	10.1331%	1 059 959.20	10.5996%	1 050 133.17	10.5013%
	8	1 030 994.40	10.3099%	1 050 252.99	10.5025%	1 056 710.94	10.5671%
	9	1 029 387.01	10.2939%	1 044 591.69	10.4459%	1 054 817.23	10.5482%
	10	1 006 533.29	10.0653%	1 060 250.15	10.6025%	1 057 675.64	10.5768%
	Avg	1 022 521.93	10.2252%	1 055 237.20	10.5524%	1 056 071.96	10.5607%
	StdDev	9 323.02	0.0932%	5 751.58	0.0575%	2 723.71	0.0272%
	Min	1 006 533.29	10.0653%	1 044 591.69	10.4459%	1 050 133.17	10.5013%
	Max	1 037 511.32	10.3751%	1 061 305.27	10.6131%	1 058 935.81	10.5894%
	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
28	1	1 023 889.90	10.2389%	1 060 966.20	10.6097%	1 061 666.90	10.6167%
	2	1 010 612.48	10.1061%	1 057 139.14	10.5714%	1 058 577.50	10.5858%
	3	1 035 562.85	10.3556%	1 052 628.47	10.5263%	1 062 842.00	10.6284%
	4	1 021 512.90	10.2151%	1 054 057.07	10.5406%	1 061 534.99	10.6153%
	5	1 009 159.82	10.0916%	1 057 518.12	10.5752%	1 061 207.35	10.6121%
	6	1 024 236.42	10.2424%	1 058 396.49	10.5840%	1 061 264.83	10.6126%
	7	1 016 749.46	10.1675%	1 056 435.77	10.5644%	1 061 642.35	10.6164%
	8	1 025 361.94	10.2536%	1 040 211.20	10.4021%	1 061 846.14	10.6185%
	9	1 017 225.43	10.1723%	1 038 043.49	10.3804%	1 061 846.14	10.6185%
	10	1 017 814.94	10.1781%	1 052 815.19	10.5282%	1 060 848.70	10.6085%
	Avg	1 020 212.62	10.2021%	1 052 821.11	10.5282%	1 061 327.69	10.6133%
	StdDev	7 706.72	0.0771%	7 675.02	0.0768%	1 100.46	0.0110%
	Min	1 009 159.82	10.0916%	1 038 043.49	10.3804%	1 058 577.50	10.5858%
	Max	1 035 562.85	10.3556%	1 060 966.20	10.6097%	1 062 842.00	10.6284%

Table A.5: Losses of the equity portfolio generated by Ell-QMC

	δ	0.7		0.8		0.9	
M	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
20	1	1 036 457.71	10.3646%	1 060 208.58	10.6021%	1 047 223.28	10.4722%
	2	1 034 679.11	10.3468%	1 058 707.27	10.5871%	1 048 863.08	10.4886%
	3	1 032 741.24	10.3274%	1 061 380.14	10.6138%	1 046 401.34	10.4640%
	4	1 044 383.50	10.4438%	1 060 794.21	10.6079%	1 050 289.93	10.5029%
	5	1 041 158.48	10.4116%	1 062 840.61	10.6284%	1 045 955.03	10.4596%
	6	1 032 516.60	10.3252%	1 059 228.75	10.5923%	1 052 530.57	10.5253%
	7	1 031 198.64	10.3120%	1 060 742.08	10.6074%	1 053 253.66	10.5325%
	8	1 022 538.00	10.2254%	1 062 042.48	10.6204%	1 046 121.27	10.4612%
	9	1 015 720.51	10.1572%	1 059 618.91	10.5962%	1 046 488.96	10.4649%
	10	1 027 828.47	10.2783%	1 062 859.24	10.6286%	1 050 578.99	10.5058%
	Avg	1 031 922.22	10.3192%	1 060 842.23	10.6084%	1 048 770.61	10.4877%
	StdDev	8 417.10	0.0842%	1 447.74	0.0145%	2 748.38	0.0275%
	Min	1 015 720.51	10.1572%	1 058 707.27	10.5871%	1 045 955.03	10.4596%
	Max	1 044 383.50	10.4438%	1 062 859.24	10.6286%	1 053 253.66	10.5325%
	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
24	1	1 030 112.26	10.3011%	1 063 231.55	10.6323%	1 059 783.10	10.5978%
	2	1 030 071.21	10.3007%	1 063 705.81	10.6371%	1 058 701.56	10.5870%
	3	1 033 526.53	10.3353%	1 061 662.09	10.6166%	1 058 502.08	10.5850%
	4	1 016 870.20	10.1687%	1 063 633.47	10.6363%	1 058 948.34	10.5895%
	5	1 019 635.54	10.1964%	1 064 087.16	10.6409%	1 055 489.54	10.5549%
	6	1 021 264.20	10.2126%	1 062 526.66	10.6253%	1 058 590.35	10.5859%
	7	1 028 153.55	10.2815%	1 062 236.40	10.6224%	1 055 148.91	10.5515%
	8	1 017 961.11	10.1796%	1 059 742.09	10.5974%	1 059 059.59	10.5906%
	9	1 025 284.12	10.2528%	1 061 227.87	10.6123%	1 056 897.98	10.5690%
	10	1 033 054.21	10.3305%	1 061 936.83	10.6194%	1 058 965.10	10.5897%
	Avg	1 025 593.29	10.2559%	1 062 398.99	10.6240%	1 058 008.66	10.5801%
	StdDev	6 274.32	0.0627%	1 333.53	0.0133%	1 594.11	0.0159%
	Min	1 016 870.20	10.1687%	1 059 742.09	10.5974%	1 055 148.91	10.5515%
	Max	1 033 526.53	10.3353%	1 064 087.16	10.6409%	1 059 783.10	10.5978%
	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
28	1	1 019 371.76	10.1937%	1 062 316.94	10.6232%	1 060 321.62	10.6032%
	2	1 022 465.51	10.2247%	1 063 713.45	10.6371%	1 061 857.27	10.6186%
	3	1 039 377.18	10.3938%	1 062 054.41	10.6205%	1 063 327.08	10.6333%
	4	1 014 147.70	10.1415%	1 058 502.38	10.5850%	1 064 554.91	10.6455%
	5	1 024 943.54	10.2494%	1 062 813.74	10.6281%	1 062 880.21	10.6288%
	6	1 018 616.16	10.1862%	1 064 483.19	10.6448%	1 063 140.41	10.6314%
	7	1 028 880.74	10.2888%	1 061 039.33	10.6104%	1 061 023.33	10.6102%
	8	1 020 928.69	10.2093%	1 059 512.50	10.5951%	1 062 261.42	10.6226%
	9	1 012 586.72	10.1259%	1 059 665.26	10.5967%	1 063 410.91	10.6341%
	10	1 040 716.17	10.4072%	1 062 302.10	10.6230%	1 062 807.09	10.6281%
	Avg	1 024 203.42	10.2420%	1 061 640.33	10.6164%	1 062 558.43	10.6256%
	StdDev	9 601.57	0.0960%	1 928.16	0.0193%	1 235.91	0.0124%
	Min	1 012 586.72	10.1259%	1 058 502.38	10.5850%	1 060 321.62	10.6032%
	Max	1 040 716.17	10.4072%	1 064 483.19	10.6448%	1 064 554.91	10.6455%

Table A.6: Losses of the equity portfolio generated by Cube-MC

		Ell − MC		Ell − QMC	
	δ	1.0		1.0	
M	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
1	1	62 109.32	6.2347%	61 366.10	6.1601%
	2	62 112.64	6.2351%	62 318.24	6.2557%
	3	62 244.22	6.2483%	62 167.64	6.2406%
	4	62 217.15	6.2456%	62 240.83	6.2479%
	5	62 144.61	6.2383%	61 722.70	6.1959%
	6	61 229.19	6.1464%	62 084.53	6.2323%
	7	61 835.32	6.2072%	62 017.16	6.2255%
	8	61 402.44	6.1638%	61 990.22	6.2228%
	9	61 338.55	6.1574%	61 365.60	6.1601%
	10	61 429.21	6.1665%	61 220.76	6.1455%
	Avg	61 806.27	6.2043%	61 849.38	6.2087%
	StdDev	410.78	0.0412%	402.28	0.0404%
	Min	61 229.19	6.1464%	61 220.76	6.1455%
	Max	62 244.22	6.2483%	62 318.24	6.2557%

		Cube − MC		Cube − QMC	
	δ	1.0		1.0	
M	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
1	1	61 698.51	6.1935%	62 085.72	6.2324%
	2	61 314.04	6.1549%		
	3	61 799.79	6.2037%		
	4	61 982.81	6.2220%		
	5	62 316.99	6.2556%		
	6	61 191.55	6.1426%		
	7	62 192.49	6.2431%		
	8	61 716.30	6.1953%		
	9	62 366.35	6.2605%		
	10	62 000.20	6.2238%		
	Avg	61 857.90	6.2095%	62 085.72	6.2324%
	StdDev	395.21	0.0397%	-	-
	Min	61 191.55	6.1426%		
	Max	62 366.35	6.2605%		

Table A.7: Losses of the option portfolio generated by Ell-MC, Ell-QMC, Cube-MC, Cube-QMC.

		Ell – MC		Ell – QMC	
	δ	1.0		1.0	
M	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
1	1	669 928.89	6.6993%	770 366.08	7.7037%
	2	682 946.48	6.8295%	739 819.77	7.3982%
	3	624 094.09	6.2409%	752 521.00	7.5252%
	4	711 788.16	7.1179%	768 584.82	7.6858%
	5	640 872.01	6.4087%	776 683.36	7.7668%
	6	730 146.76	7.3015%	771 317.00	7.7132%
	7	640 649.83	6.4065%	745 246.42	7.4525%
	8	708 588.40	7.0859%	726 436.08	7.2644%
	9	690 073.11	6.9007%	767 276.53	7.6728%
	10	674 446.71	6.7445%	773 547.93	7.7355%
	Avg	677 353.44	6.7735%	759 179.90	7.5918%
	StdDev	34 512.09	0.3451%	17 077.32	0.1708%
	Min	624 094.09	6.2409%	726 436.08	7.2644%
	Max	730 146.76	7.3015%	776 683.36	7.7668%

		Cube – MC		Cube – QMC	
	δ	1.0		1.0	
M	Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
1	1	648 121.23	6.4812%	697 073.09	6.9707%
	2	644 619.10	6.4462%		
	3	619 106.14	6.1911%		
	4	630 312.47	6.3031%		
	5	629 707.12	6.2971%		
	6	651 659.28	6.5166%		
	7	617 538.61	6.1754%		
	8	645 588.10	6.4559%		
	9	620 944.31	6.2094%		
	10	638 180.99	6.3818%		
	Avg	634 577.74	6.3458%	697 073.09	6.9707%
	StdDev	12 770.34	0.1277%	-	-
	Min	617 538.61	6.1754%		
	Max	651 659.28	6.5166%		

Table A.8: Losses of the equity portfolio generated by Ell-MC, Ell-QMC, Cube-MC and Cube-QMC without zooming-in procedure.

	Surf − MC		Surf − QMC	
Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
1	50 279.18	5.0472%	50 689.20	5.0884%
2	52 275.50	5.2476%		
3	49 885.54	5.0077%		
4	50 959.68	5.1155%		
5	49 460.59	4.9650%		
6	50 385.79	5.0579%		
7	51 113.77	5.1310%		
8	50 367.08	5.0560%		
9	49 421.33	4.9611%		
10	50 656.28	5.0850%		
Avg	50 480.48	5.0674%	50 689.20	5.0884%
StdDev	847.35	0.0851%	-	-
Min	49 421.33	4.9611%		
Max	52 275.50	5.2476%		

Table A.9: Losses of the option portfolio generated by MC and QMC with searching on a surface of the ellipsoid

	Surf − MC		Surf − QMC	
Run	Abs. Loss	Rel. Loss	Abs. Loss	Rel. Loss
1	88 468.17	0.8847%	107 943.01	1.0794%
2	94 796.34	0.9480%		
3	106 896.98	1.0690%		
4	98 507.26	0.9851%		
5	100 055.94	1.0006%		
6	89 732.45	0.8973%		
7	92 624.35	0.9262%		
8	91 917.54	0.9192%		
9	92 218.51	0.9222%		
10	104 872.51	1.0487%		
Avg	96 009.00	0.9601%	107 943.01	1.0794%
StdDev	6 324.75	0.0632%	-	-
Min	88 468.17	0.8847%		
Max	106 896.98	1.0690%		

Table A.10: Losses of the equity portfolio generated by Surf-MC and Surf-QMC.

Bibliography

[1] Acerbi C., C. Nordio, C. Sirtari (2001): "Expected Shortfall as a Tool for Financial Risk Management". Working paper.
Accessible at http://www.gloriamundi.org/picsresources/cacncs.pdf

[2] Artzner Ph., Delbaen F., Eber J.-M. and Heath D. (1997): Thinking Coherently. *Risk*, **10**(11).

[3] Artzner Ph., Delbaen F., Eber J.-M. and Heath D. (1999): "Coherent measures of risk", *Mathematical Finance* **9** (3), 203–228.

[4] Basle Committee on Banking Supervision (1996): *Amendment to the Capital Accord to Incorporate Market Risks*, Bank for International Settlements.
Accessible at http://www.bis.org/publ/cgfs18.htm

[5] Boss M. Breuer T., Krenn G. (2002): "Using the Suchmaschine", Internal Paper, Oesterreichische Nationalbank, 2002

[6] Breuer T., Krenn G. (1999): Stress Testing, Guidelines on Market Risk, Volume 5, Oesterreichische Nationalbank.
Accessible at http://www.oenb.co.at/pdfdown/bankenauf/band5-ev40.pdf

[7] Breuer T., Krenn G., Pistovčák F. (2002): "Stress Tests, Maximum Loss, and Value at Risk", 301–313 in B. Britzelmaier, S. Geberl, H.-R. Kaufmann, M. Menichetti (eds.): *Regulierung oder Deregulierung der Finanzmärkte*, Heidelberg: Physica

[8] Campbell J. Y., Lo A. W., MacKinlay A. C. (1997): *The Econometrics of Financial Markets.* Princeton University Press, Princeton, New Jersey.

[9] Derivatives Policy Group (1995): *Framework for Voluntary Oversight.*
Accessible at http//risk.ifci.ch/137790.htm

[10] Eiselt H. A., Pederzoli G., Sandblom C. L. (1987): *Continuous Optimization Models* Berlin: Walter de Gruyter

[11] Fishman G. S. (1996): *Monte Carlo: Concepts, Algorithms, and Applications* Springer Series in Operations Research, New York: Springer-Verlag

[12] Hull J. C. (1999): *Options, Futures, & Other Derivatives*, 4th ed., Prentice-Hall

[13] Huyer W., Neumaier A. (1999): "Global Optimization by Multilevel Coordinate Search", *Journal of Global Optimization* **14**, 331–355

[14] Mina J., Jerry Yi Xiao (2001): "Return to RiskMetrics: The Evolution of a Standard", New York: RiskMetrics.
Accessible at http://www.riskmetrics.com

[15] Niederreiter H. and Peart P. (1986): "Localization of Search in Quasi-Monte Carlo Methods for Global Optimization", *SIAM J. Sci. Stat. Comput.***7**(2), 660–664

[16] Niederreiter H., Xing C. P. (1995): "A Construction of Low-Discrepancy Sequences Using Global Function Fields", *Acta Arith.* **73**, 87–102

[17] Niederreiter H., Xing C. P. (1995): "Low-Discrepancy Sequences Obtained from Algebraic Function Fields Over Finite Fields", *Acta Arith.* **72**, 281–298

[18] OECD (2002): *Main Economic Indicators*, November 2002, Available at http://www.oecd.org

[19] Pirsic G. (2002):"A Software Implementation of Niederreiter-Xing Sequences", 434–445 in H. Niederreiter, K.-T. Fang, F.J. Hickernell (eds.): *Monte Carlo and Quasi-Monte Carlo Methods 2000*, Berlin: Springer-Verlag

[20] RiskMetrics (1996): "RiskMetrics Technical Document", 4th ed. New York: RiskMetrics
Accessible at http://www.riskmetrics.com

[21] Shaw J. (1997): "Beyond VAR and Stress Testing", in *VAR - Understanding and Applying Value-at-Risk.* KPMG Risk Publications, 221-224.

[22] Studer G. (1997): *Maximum Loss for Measurement of Market Risk*, Ph.D. thesis No. 12397, ETH Zurich. Accessible at http://www.ifor.math.ethz.ch/ studer/

[23] Uryasev S. (2000): "Conditional Value at Risk: Optimisation Algorithms and Applications", *Financial Engineering News* **2**(3)

Weitere Arbeiten
Forschungszentrum Prozess- und Produkt-Engineering

ANWENDUNGEN

Kurzfristige Prognose des Stromverbrauchs in Vorarlberg auf Stunden- und Viertelstundenbasis
Thomas Steinberger, 2004

Weiterbildungs- und Qualifizierungsbedarf kleinerer und mittlerer Unternehmen in Vorarlberg bezüglich Prozess- und Projektmanagement, Führung, Strategie und Innovationsmanagement
Markus Reichart, Julia Schneider, Isabella Gratzer, 2004

Netzwerke für Innovationen
Martin Meusburger, Markus Reichart, Karin Feurstein, 2005

Neue Technologien im Produktinnovationsprozess
Julia Schneider, Markus Reichart, 2005

Bezug von externen Leistungen in der Produktentwicklung Aktueller Stand - Trends - Verbesserungspotenziale
Julia Schneider, 2005

METHODEN

Identifying Worst Case Scenarios of Security Portfolios with Quasi-Random Search Algorithms
Thomas Breuer, Filip Pistovcak, 2004

A General Noise Model and Its Effects on Evolution Strategy Performance
Hans-Georg Beyer, Dirk V. Arnold, 2004

Using Quasi-Monte Carlo Scenarios in Risk Management
Thomas Breuer, Filip Pistovcak, 2004

An Explicit Characterization of Calogero-Systems
Fritz Gesztesy, Karl Unterkofler, Rudi Weikard, 2004

Reliability of new Fibrillation Detection Algorithms for Automated External Defibrillators
Anton Amann, Robert Tratnig, Karl Unterkofler, 2004

Towards an Integrated Measurement of Credit and Market Risk
Thomas Breuer, Martin Jandacka, Gerald Krenn, 2005

Umgang mit Szenarien
Falko E. P. Wilms, 2005

Umgang mit unscharfen Informationen
Falko E. P. Wilms, 2005

Fachhochschule Vorarlberg
Forschungszentrum
Prozess- und Produkt-Engineering
Hochschulstrasse 1
A-6850 Dornbirn

T +43 (0)5572 792 7100
F +43 (0)5572 792 9510

www.fhv.at/res/ppe

Fachhochschule Vorarlberg
Forschungszentrum
Prozess- und Produkt-Engineering