Optimization of performance measures based on Expected Shortfall

Tom Fischer, Heriot-Watt University, Edinburgh[*]
Armin Roehrl, Approximity GmbH[†]

July 01, 2005

Abstract

We explain how to optimize portfolios with respect to RORAC and RORC based on Expected Shortfall. Recent results from the theories of performance measurement and Swarm Intelligence are used for numeric optimization. We combine and correlate geometric Brownian motions for stocks with a two-factor Cox-Ingersoll-Ross (CIR-2) model for interest rates such that portfolios of bonds and stocks can be optimized. Examples for German market data as well as an analysis of the scalability of the algorithms to assure fast run-times on clusters of computers for large real-life portfolios are given. Differences between RORAC- and RORC-optimized portfolios are demonstrated.

JEL: G11, G31
Keywords: Cox-Ingersoll-Ross Model, Expected Shortfall, Parallel Computing, Portfolio Optimization, RORAC, RORC, Swarm Intelligence

1 Introduction

Portfolio optimization has a long history starting with Markowitz' considerations in the early 1950s. During the last ten years, interest in portfolio optimization has shifted towards optimization with respect to risk measures like Value-at-Risk (VaR) and the more sophisticated Expected Shortfall (ES), in its various slightly differing forms also known as Conditional VaR (CVaR),

[*] Heriot-Watt University, Department of Actuarial Mathematics and Statistics, Edinburgh EH14 4AS, United Kingdom; Tel: +44 131 451 4371; Email: fischer@ma.hw.ac.uk. The author gratefully acknowledges partial funding by the German Federal Ministry of Education and Research (BMBF, 03LEM6DA).

[†] Hauser Muehlberg 21, 93444 Koetzting, Germany. Tel.: +49 170 1639874; Email: aroehrl@approximity.com. The authors thank Angelika May for support.

Tail-VaR or Average VaR. Portfolio-VaR optimization using gradient search methods can already be found in a Ph.D. dissertation of Lemus Rodriguez (1999) and in an article by Gouriéroux, Laurent and Scaillet (2000). Since then, gradient search has lost a lot of its attraction, especially as CVaR/ES was more and more seen as the much more sophisticated risk measure coming along with a computationally attractive optimization method. The articles of Rockafellar and Uryasev (2000, 2002) that derived a linear optimization "short-cut" formula for CVaR gave the risk management community a comprehensive and efficient optimization tool at hand.

Although there exists a lot of literature on CVaR/ES-optimization, to our knowledge there is hardly any on portfolio optimization with respect to the performance measures Return-On-Risk-Capital and Return-On-Risk-Adjusted-Capital (RORC and RORAC) in the case where both are calculated with respect to a modern risk measure like ES. In particular, for this type of problem there seems to be no optimization shortcut as in the case of CVaR/ES-optimization. There also seems to be no literature about the use and implications of specific financial market models for such performance optimization. Furthermore, while the focus of our paper lies on the detailed explanation of the presented methods, we also want to demonstrate how fast developing computer technology enables to solve portfolio optimization problems numerically even for complex market models and big portfolios.

After briefly introducing the risk measure Expected Shortfall (ES) and the performance measures RORC and RORAC with respect to ES, the paper explains two concepts, Gradient Search (GS) and (Particle) Swarm Intelligence (SI), for portfolio optimization with respect to one of these measures. After that, a specific financial market model is proposed. This model enables optimization of portfolios consisting of stochastically developing bonds and stocks. The model combines and correlates stocks represented by geometric Brownian motions with a two-factor Cox-Ingersoll-Ross (CIR-2) model for the interest rates. The model choice is not arbitrary, but motivated by our experience that the mentioned models have (often separately and in largely simplified forms) been considered in major German life insurance companies. The reader is provided with explicit model equations and formulas and also explicit algorithms for GS- and SI-optimization. In particular, the gradients of the RORC and RORAC performance measures under budget constraints are derived. The reader is shown examples of optimized portfolios for German market data. These examples give strong evidence that the choice between RORAC and RORC can make a significant difference for the optimal portfolios even though these performance measures look quite similar on the first view. We furthermore analyze the scalability of our algorithms with respect to multi-processor computers, computer clusters or Global Grid environments to assure fast run-times for large real-life portfolios.

The paper is organized as follows. Section 2 introduces the considered risk and performance measures. Section 3 explains the general approach to the optimization problem. Furthermore, Gradient Search, the Swarm Intelligence optimization method and the role of stochastic simulation in these approaches are described. After this, Section 4 introduces and discusses the proposed financial market model. Section 5 is dedicated to the stochastic simulation part, i.e. the generation of market scenarios. In Section 6, information on parameter estimation can be found. Section 7 gives a brief chronological overview of all steps concerning the proposed optimization methodology. In Section 8 we present our findings for German market data. Section 9 shows the scalability of our approaches. In Section 10 we conclude. Section 11 contains tables with numeric results.

2 Risk and performance measures

2.1 General definitions

Let us define the *total payoff* (or *total risk*) of a portfolio by

$$X = X(u) := \sum_{i=1}^{n} u_i X_i, \tag{1}$$

where the *portfolio* $u = (u_i)_{1 \leq i \leq n} \in \mathbb{R}^n$ represents $n \in \mathbb{N}$ different *payoffs* X_i $(1 \leq i \leq n)$ with weights $u_i \in \mathbb{R}$. The X_i are assumed to be one-dimensional real-valued random variables. We call $B = (X_1, \ldots, X_n)$ a *portfolio base* (cf. Fischer, 2003) as the payoff of any considered portfolio $u \in \mathbb{R}^n$ will be described by (1). As random variables the components of B do *not* have to be linearly independent.

Now, consider a financial market with n securities. Assume that the prices of these securities at time $s \in \mathbb{R}_0^+$ (the positive real numbers including 0) are given by random variables

$$V_1(s), V_2(s), \ldots, V_n(s). \tag{2}$$

At $s = 0$, which is the present, we assume to have constants. We will use the notation

$$V_i = V_i(0) \tag{3}$$

and

$$V = V(u) := \sum_{i=1}^{n} u_i V_i(0) \tag{4}$$

for the value of the portfolio u at time 0. For the rest of the paper, we will consider specific payoffs X (or X_i), namely *returns* defined by

$$X_i := V_i(t) - V_i(0), \tag{5}$$

where the variable $t > 0$ is the considered time horizon for which performance or risk management is performed. Hence, the X_i are wins or losses due to the i-th asset during the time interval $[0, t]$, and the portfolio u has a *total return* for the time interval from 0 to t of exactly $X(u)$.

A *risk measure* ρ is usually defined as a mapping that assigns to a random variable X (e.g. payoff or return) a real number $\rho(X)$, its risk:

$$\rho : X \longmapsto \rho(X). \tag{6}$$

The amount $\rho(X)$ is commonly interpreted as the minimum cash such that the “risk” of X is “acceptable” to the holder of the portfolio whenever he/she has the additional amount $\rho(X)$ stored as *risk capital* (cf. Artzner et al., 1999). Working with a portfolio base $B = (X_1, \ldots, X_n)$, a risk measure ρ on the payoffs $X(u)$ for $u \in \mathbb{R}^n$ implies a risk measure ρ_B on the portfolios u. In particular, we can define

$$\rho_B : u \longmapsto \rho(X(u)). \tag{7}$$

We also write $\rho(u)$ for $\rho_B(u)$. Based on the context, no confusion should arise.

A *performance measure* can also be represented by functions as just considered above. However, in contrast to risk measures, performance measures are ususally intended to describe ratios like the relation of the expected return to the risk capital or invested risk-adjusted capital. We have a look at conrete examples, namely RORC and RORAC, in Section 2.3.

2.2 Expected Shortfall

For $0 < \alpha < 1$, we define the α-*Value-at-Risk* by

$$\mathrm{VaR}_\alpha(X) := -\inf\{x : P(X \leq x) \geq \alpha\}. \tag{8}$$

Hence, VaR is a negative α-quantile of the distribution of the random variable X, i.e. -$\mathrm{VaR}_\alpha(X)$ is a threshold which is fallen short of in $\alpha \cdot 100\%$ of all cases. *Expected Shortfall (ES)* is defined by

$$\mathrm{ES}_\alpha(X) := -\mathbf{E}[X|X \leq -\mathrm{VaR}_\alpha(X)]. \tag{9}$$

i.e. -$\mathrm{ES}_\alpha(X)$ is the expectation (i.e. the mean) of the losses under the condition that the threshold mentioned above has already been fallen short of.

2.3 RORC and RORAC

The performance measure RORC, the *Return-On-Risk-Capital*, is defined by

$$\varphi(X) := \frac{\mathbf{E}[X]}{\rho(X)}. \tag{10}$$

RORC measures the mean return per unit risk capital. For $\rho = \mathrm{ES}_\alpha$, we have

$$\varphi_\alpha(X) := \frac{\mathbf{E}[X]}{\mathrm{ES}_\alpha(X)}, \tag{11}$$

the ES-RORC. Carrying out numerical RORC-optimization, one might face the problem that the optimal portfolio (although the portfolio value is constant) could imply a huge amount of risk capital together with a huge expected return. However, practical reasons might imply an upper bound for the risk capital, e.g. in case of the ES-RORC $\mathrm{ES}_\alpha(u) \leq \mathrm{ES}_{\max}$ might be imposed.

The performance measure RORAC, the *Return-On-Risk-Adjusted-Capital*, is defined by

$$\psi(X) := \frac{\mathbf{E}[X]}{V + \rho(X)}. \tag{12}$$

Indeed, ψ measures the mean (or expected) return per unit engaged capital, since $V + \rho$ is the value of the invested capital plus the cost of risk (cf. (4)). Hence, in contrast to RORC (10), RORAC considers not only the risk capital but the risk-*adjusted* investment capital. For $\rho = \mathrm{ES}_\alpha$, i.e.

$$\psi_\alpha(X) := \frac{\mathbf{E}[X]}{V + \mathrm{ES}_\alpha(X)}, \tag{13}$$

we talk of the ES-RORAC. As in the case of RORC, an upper bound for the risk capital might be imposed in the case of RORAC-optimization.

3 Portfolio optimization

3.1 The problem

We assume a fixed budget, i.e. portfolio value, of V which must be fully invested at the present time $t = 0$. We are looking for the portfolio in $\mathbb{R}^n$ with value V which minimizes (maximizes) the risk (performance). For convenience, we assume that the risk ρ has to be minimized. As V_i is the price of asset i at time 0, this implies the following constraint for the portfolios u:

$$V = \sum_{i=1}^{n} u_i V_i. \tag{14}$$

A possible solution of the optimization problem is given by a portfolio $u^* \in \mathbb{R}^n$, such that $\rho(u^*)$ is minimal (on $\mathbb{R}^n$) under (14). Defining

$$u'_n := (V - \sum_{i<n} u_i V_i)/V_n, \tag{15}$$

and ρ' as

$$\rho'(u_1, \ldots, u_{n-1}) := \rho(u_1, \ldots, u_{n-1}, u'_n) \tag{16}$$

it follows from (14) that we can express the solution u^* by

$$(u^*_1, \ldots, u^*_{n-1}) = \text{argmin } \rho'(u_1, \ldots, u_{n-1}), \tag{17}$$

together with $u^*_n = (V - \sum_{i<n} u^*_i V_i)/V_n$. Working with real data, we discovered that candidate portfolios for extremal points due to the considered risk or performance measures can contain tremendous amounts of short-sold assets, i.e. the portfolio as a vector of real numbers contains huge negative components. For this reason, we introduce a further constraint: For $a, b \geq 0$, we require

$$-b\frac{V}{V_i} \leq u_i \leq a\frac{V}{V_i} \quad \text{for all } 1 \leq i \leq n. \tag{18}$$

For instance, $b = 0$ implies portfolios which allow no short-selling. The values $a = b = 1$ guarantee that the amount of capital or debts in no asset is bigger than the total value V of the portfolio.

3.2 Gradient Search

Let the risk measure ρ be differentiable on $\mathbb{R}^n$. From standard analysis we obtain for $1 \leq i \leq n-1$

$$\frac{\partial \rho'}{\partial u_i}(u_1, \ldots, u_{n-1}) = \frac{\partial \rho}{\partial u_i}(u_1, \ldots, u_{n-1}, u'_n) - \frac{V_i}{V_n}\frac{\partial \rho}{\partial u'_n}(u_1, \ldots, u_{n-1}, u'_n). \tag{19}$$

```
maxits=bigNumber
its=0
while its < maxits    # fixed number of iterations
  ptf=RandomPortfolio # satisfying constraints
  loop
    if Rho(ptf)<Rho(bestPtf)     # small Rho wanted
      bestPtf=ptf
    end
    grd=gradient(ptf)
    ptfnew=ptf-grd*stepsize      # if constraints, adapt this step
  until Rho(ptfnew)>Rho(ptf)
  its+=1
end
```

Figure 1: Simple pseudo-code for portfolio optimization using brute-force gradient search. Clever varying the step-size can give further speed-up. Instead of chosing random portfolios, one can use a grid-based search.

Using the partial derivates (19), one can start looking for the (local) extreme points of ρ' in $\mathbb{R}^{n-1}$ by applying Gradient Search (GS) methods. This might be a comfortable approach to solve the optimization problem (17) as long as the considered measures have sufficient differentiability properties. However, the proof of such differentiability properties can be rather difficult (see e.g. Tasche, 2000). This is one reason for our proposal of Swarm Intelligence optimization methods (see Subsection 3.3).

Outline of a gradient minimum-search (Figure 1):

(1) Evaluate the gradient for the current portfolio. (2) If the gradient is zero, exit. We have found a local or global minimum. (3) Follow the negative gradient of the current portfolio one small step. Modify the portfolio to satify the constraints. Continue with step 1.

On a one-processor computer this gradient algorithm has to be started many times with different portfolios ("brute-force"), as one might be stuck in a local minimum.

We need the partial derivatives (19) of ES, ES-RORC and ES-RORAC.

Expected Shortfall

Assuming sufficient differentiability properties, one can show that

$$\frac{\partial \mathrm{ES}_\alpha}{\partial u_i}(u_1, \ldots, u_n) = -\mathbf{E}[X_i | X \leq -\mathrm{VaR}_\alpha(X)]. \tag{20}$$

We refer to Tasche (2000) for further information on the differentiation of ES. For $\rho(u) = \mathrm{ES}_\alpha(X(u))$, equation (19) can be written as

$$\begin{aligned} \frac{\partial \mathrm{ES}'_\alpha}{\partial u_i}(u_1, \dots, u_{n-1}) &= -\mathbf{E}[X_i | X \leq -\mathrm{VaR}_\alpha(X)] \\ &\quad + \frac{V_i}{V_n}\mathbf{E}[X_n | X \leq -\mathrm{VaR}_\alpha(X)]. \end{aligned} \tag{21}$$

This expression is suitable for numerical evaluation by Monte-Carlo methods.

RORC

The partial derivatives (19) of the ES-RORC under the budget constraint (14) are obtained from standard differentiation rules:

$$\begin{aligned} \frac{\partial \varphi'_\alpha}{\partial u_i} &= -\mathbf{E}[X] \cdot \frac{\mathbf{E}[X_i | X \leq -\mathrm{VaR}_\alpha(X)]}{\mathrm{ES}_\alpha(X)^2} + \frac{\mathbf{E}[X_i]}{\mathrm{ES}_\alpha(X)} \\ &\quad - \frac{V_i}{V_n}\left(-\mathbf{E}[X] \cdot \frac{\mathbf{E}[X_n | X \leq -\mathrm{VaR}_\alpha(X)]}{\mathrm{ES}_\alpha(X)^2} + \frac{\mathbf{E}[X_n]}{\mathrm{ES}_\alpha(X)} \right). \end{aligned} \tag{22}$$

RORAC

As V is a constant, the partial derivatives (19) of the ES-RORAC under the budget constraint (14) are similar to those of RORC:

$$\begin{aligned} \frac{\partial \psi'_\alpha}{\partial u_i} &= -\mathbf{E}[X] \cdot \frac{\mathbf{E}[X_i | X \leq -\mathrm{VaR}_\alpha(X)]}{(V + \mathrm{ES}_\alpha(X))^2} + \frac{\mathbf{E}[X_i]}{V + \mathrm{ES}_\alpha(X)} \\ &\quad - \frac{V_i}{V_n}\left(-\mathbf{E}[X] \cdot \frac{\mathbf{E}[X_n | X \leq -\mathrm{VaR}_\alpha(X)]}{(V + \mathrm{ES}_\alpha(X))^2} + \frac{\mathbf{E}[X_n]}{V + \mathrm{ES}_\alpha(X)} \right). \end{aligned} \tag{23}$$

Again we see that (22) and (23) are relatively simple expression of expectations and therefore suitable for Monte-Carlo methods.

3.3 Swarm Intelligence

Swarm Intelligence (SI) is a property of a system where the collective behaviours of unsophisticated agents interacting locally with their environment cause coherent functional global patterns to emerge. SI provides a basis with which it is possible to explore collective or distributed problem solving without centralized control or the provision of a global model (cf. Kennedy et al., 2001). The three underlying principles of SI are: evaluate, compare and imitate. Living organisms can learn by evaluating stimuli and rate them as positive or negative. In our case this is the metric (i.e. risk or performance measure) we want to optimize. As practiced in the Adaptive Culture Model (cf. Shibanai, Yasuno and Ishiguro, 2001) and in real life, people compare themselves to others and imitate only those neighbours that are superior to themselves. SI offers a tradeoff between individual and group learning.

We give a brief outline of the algorithm (cf. Kennedy et al. (2001), Kennedy and Eberhart (1995)) using standard notation. Let y_i be the position of particle i. In our case, the position represents a specific portfolio ($y_i \in \mathbb{R}^n$). The change of portfolio is called v. v traditionally stands for velocity. Each clockstep t particles move from one stop to another by $y_i(t) = y_i(t-1) + v_i(t)$ and sample the search space by modifying the velocity term. The direction of movement is a function of the current position (y_i), velocity (v_i), the location of the individual's previous best success (p_i), and the best position found by any member of the neighborhood (p_g):

$$y_i(t) = f(y_i(t-1), v_i(t-1), p_i, p_g). \tag{24}$$

One possible implementation is

$$v_i(t) = v_i(t-1) + n_1(p_i - y_i(t-1)) + n_2(p_g - y_i(t-1)) \tag{25}$$

with

$$y_i(t) = y_i(t-1) + v_i(t). \tag{26}$$

The n variables are random variables defined by an upper limit, so that the particles cycle around the two best bets: p_i and p_g. The random numbers (n_1 and n_2) are updated in every iteration. With real data the velocity v very quickly becomes too large and one has to set limits.

As the value of a portfolio has to remain constant, two minor modifications in the choice of v are required.

In simulation studies on typical portfolios it proves successful to inject about 10% of new particles with random speeds and locations from time to time and to remove the 10% worst performing particles. The exact population size is an open research problem with experts having different opinions. A rule of thumb is to keep the population size small, but to rely on a high number of iterations. As this can take a long time for higher dimensional problems, parallel computing solutions are an easy way out of the dilemma, following Kent Thompson's (co-inventor of Unix) famous quote: ”When in doubt, use brute force”.

```
ys=generateInitialPortfolios     #satisfying the constraint
p=ys
loop
  # best portfolios's fitness so far
  ys.each_with_index{ |y,i|
    p[i]=y, if Rho(y)<Rho(p[i])  # small Rho wanted
  }

  i=rand size                    # arbitrary Rho
  g=i
  for j=indexes of neighbors
    g=j if Rho(p[j])<Rho(p[g])   # g: index of best performer
                                 # in the neighborhood
  end
  #assuming delta t=1
  v[i]=[i]+n1*(p[i]-ys[i])+n2*(p[g]-ys[i])
  v[i]=Vmax if v_id>Vmax
  v[i]=Vmin if v_id<Vmin
  ys[i]=ys[i]+v[i]

  fixPortfolio                # constraint
  if loopCount mod 10000==0   # big number here
    removeWorstPortfolio      # remove 10% worst portfolios
    injectNewPortfolios       # inject 10% new portfolios
  end
until some criterion
```

Figure 2: Simple pseudo-code for portfolio optimization using swarm particles based on Kennedy et al. (2001). This basic algorithm is implemented more efficiently.

3.4 How stochastic simulation fits in

Whether GS or SI methods are used to solve the portfolio optimization problem, a way must be found to determine the distributions of the payoff functions X_i since finally any optimization routine must compute the measures (9), (11) or (13). The X_i (cf. (5)) are wins or losses due to the i-th asset in the market where we assumed to have n numbered assets. It is clear that ES, ES-RORC, ES-RORAC, but also their derivatives, crucially depend on the model for the future prices $V_i(t)$ of these assets. The particular stochastic model we use will be introduced in Section 4.

Once the model for the price processes is chosen and a way to get possible parameters is found, one can theoretically evaluate the risk and performance measures (9), (11) and (13) and their partial derivatives under budget constraint (21), (22) and (23). However, one often encounters models (also in our case) where it is not possible or quite difficult to calculate these values directly. A more realistic assumption is that one succeeds in doing a stochastic simulation of the model which computes $m \in \mathbb{N}^+$ (e.g. $m = 10^4$) market scenarios, i.e. finally one has for each i numerical realizations (here given in increasing order)

$$x_i^1, x_i^2, \dots, x_i^m \tag{27}$$

of the random variable X_i defined by (5). The realizations (also in increasing order)

$$x^1, x^2, \dots, x^m \tag{28}$$

for any $X = X(u)$ follow immediately. Given these realizations, estimates for the stochastic expressions in the considered measures can be used. In particular, we compute estimates using the "empirical" distribution given by the simulation output, e.g.

$$\widehat{\mathbf{E}}[X] = \frac{1}{m}\sum_{j=1}^{m} x^j, \tag{29}$$

$$\widehat{\mathrm{VaR}}_\alpha(X) = -x^{\lceil \alpha m \rceil} \tag{30}$$

or

$$\widehat{\mathrm{ES}}_\alpha[X] = \frac{-\sum_{x^j \le -\widehat{\mathrm{VaR}}_\alpha(X)} x^j}{\mathrm{card}\{j : x^j \le -\widehat{\mathrm{VaR}}_\alpha(X)\}}, \tag{31}$$

where $\lceil r \rceil$ denotes the smallest integer which is greater or equal the real number r. One could use other perhaps more sophisticated estimators. Nonetheless, replacing all stochastic expressions in (9), (11), (13), (21), (22) and (23) as suggested by (29) to (31), one obtains approximations for the respective measures and their gradients which are easy to implement in any suitable programming language. Gradient Search methods or Swarm Intelligence optimization methods can now be executed using the obtained approximations.

4 A particular market model

Until now, the presented theory has been completely independent from a particular market model eventually used. In the following sections we apply the above ideas to a concrete model and data setup.

We model stocks and non-defaultable bonds. All stochastics evolve from a $(d+2)$-dimensional Brownian motion (Wiener process) $(W_i)_{i=1,\dots,d+2}$, where the first two components drive the dynamics of the two-factor interest rate model for the bonds and the last d drive the dynamics of d stocks. The Brownian motions W_i are correlated by a covariance matrix Σ where for $i = 1, \ldots, d+2$

$$\Sigma_{i,i} = 1, \quad \text{and} \quad \Sigma_{1,0} = \Sigma_{0,1} = 0, \tag{32}$$

i.e. each W_i is a one-dimensional standard Brownian motion, and W_1 and W_2 are uncorrelated.

4.1 Interest rates and bonds

The relation between the price $p(t, \tau)$ of a *unit zero-coupon bond* with maturity τ at time t, i.e. the price at time t of the guaranteed payoff 1 at time $t + \tau$, and the corresponding *spot (interest) rate* $R(t, \tau)$ is

$$e^{R(t,\tau)\cdot\tau} = \frac{1}{p(t,\tau)}. \tag{33}$$

Hence, $R(t, \tau)$ is the at t guaranteed continuously compounded yearly rate of interest during the time interval $[t, t + \tau]$. For future points in time ($t > 0$), $p(t, \tau)$, respectively $R(t, \tau)$, are assumed to be random variables. We can now turn to the considered interest rate model of Chen and Scott (1992) with two stochastic factors. The model is usually called Cox-Ingersoll-Ross-2 (CIR-2) as it heavily relies on the work of Cox, Ingersoll and Ross (1985).

The model features two stochastic factors (x_1, x_2) fulfilling the stochastic differential equations

$$dx_i = (b_i - a_i \cdot x_i)dt + \sigma_i \sqrt{x_i} dW_i \quad (i = 1, 2) \tag{34}$$

where b_i, a_i and σ_i are positive constants. One has $x_i > 0$ if $2b_i > \sigma_i^2$. By (32) the Brownian motions W_1 and W_2 are independent. Equation (34) defines a so-called *mean reversion process.* The parameter a is called the *strength* of the mean reversion and b/a the *mean reversion level*, i.e. the long-term mean of the process x_i. The implied spot rate at time t for a maturity τ is

$$R(t, \tau) = \sum_{i=1}^{2} \left(-\frac{\log A_i(\tau)}{\tau} + \frac{B_i(\tau)}{\tau} x_i(t) \right), \tag{35}$$

the implied unit zero-coupon bond price by (33)

$$p(t, \tau, x(t)) = \prod_{i=1}^{2} A_i(\tau) e^{-B_i(\tau) x_i(t)}. \tag{36}$$

The functions A_i and B_i are given by

$$A_i(\tau) = \left[\frac{2h_i e^{(a_i+\lambda_i+h_i)\tau/2}}{2h_i + (a_i+\lambda_i+h_i)(e^{\tau h_i}-1)}\right]^{2b_i/\sigma_i^2} \tag{37}$$

and

$$B_i(\tau) = \left[\frac{2(e^{\tau h_i}-1)}{2h_i + (a_i+\lambda_i+h_i)(e^{\tau h_i}-1)}\right], \tag{38}$$

where

$$h_i = \sqrt{(a_i+\lambda_i)^2 + 2\sigma_i^2}. \tag{39}$$

The parameter λ_i is related to the change of measure (physical to martingale measure) and can together with all other parameters be estimated from historical interest rates. Prices of coupon bonds can now be calculated as the sums of prices of properly chosen zero-coupon bonds.

4.2 Stocks

We model d stocks by geometric Brownian motions, i.e. price processes S_j ($j = 1, \ldots, d$) with

$$S_j(t) = S_j(0)e^{\mu_j t + \sigma_j W_{j+2}(t)}, \tag{40}$$

where $\mu_j \in \mathbb{R}$ is the drift and $\sigma_j \in \mathbb{R}^+$ the diffusion coefficient of the Brownian motion in the exponent. In terms of stochastic differential equations we have

$$d\ln S_j = \mu_j dt + \sigma_j dW_{j+2}. \tag{41}$$

4.3 Comments

The choice of our model is based on experience with practitioners. A couple of German life insurance companies have been considering one-factor Cox-Ingersoll-Ross models together with geometric Brownian motions in the context of Asset Liability Management. Usually, the models are independently used to model debt securities and stocks. The combination and correlation of the models as proposed in this paper and also the incorporation of multifactor interest rate models seems to be an improvement of these approaches. For insurance companies (known to be conservative), an important aspect of such models is the acceptance by the scientific public. This explains somewhat the decision for "standard" models like geometric Brownian motions or the CIR-1 model. As mentioned in Fischer, May and Walther (2004), the Vasicek-2

(Gaussian) model for instance behaves in some way better than CIR-2 (concerning parameter estimation; see also Babbs and Nowman (1998)). Nonetheless, insurance companies seem to prefer CIR as under the respective parameter constraints CIR assures positive interest rates.

5 Market scenario generation

As described earlier, we are going to carry out a stochastic simulation to obtain an "empirical" distribution of the considered random payoffs. A simulation requires discretization. The width of one time step is the constant Δ (e.g. one day, one month etc.), i.e. we consider points $t_{m+1} = t_m + \Delta$ ($m \in \mathbb{N}$) and $t_0 = 0$. Increments

$$\delta W_{i,m} := W_i(t_m) - W_i(t_{m-1}) \tag{42}$$

of the $(d+2)$ Brownian motions have to be simulated. For fixed m, the $\delta W_{i,m}$ are correlated by the covariance matrix $\Delta\Sigma$ (cf. (32)). For fixed i, the increments $\delta W_{i,m}$ are independent normally distributed random variables with variance Δ and expectation 0. Hence, all discretized dynamics are driven by a series of standard normally distributed random variables $N_{i,m}$ ($i = 1, \ldots, d+2; m \in \mathbb{N}^+$), where for each m the random variables $(N_{i,m})_{i=1,\ldots,d+2}$ are correlated by the covariance matrix Σ which will later be estimated from real data.

5.1 Simulation of correlated normal random variables

Simulation of i.i.d. normal random variables is standard. Let us consider the Cholesky decomposition

$$\Sigma = CC^t \tag{43}$$

of the covariance matrix Σ. If $Z = (Z_i)_{i=1,\ldots,d+2}$ are $d+2$ i.i.d. normal random variables, then

$$(N_i)_{i=1,\ldots,d+2} = N = C \cdot Z \tag{44}$$

contains $d+2$ normally distributed random variables with covariances Σ.

5.2 Interest rates and bonds

From (34), a so-called Euler-approximation gives the recursion

$$x_{i,m} = x_{i,m-1} + (b_i - a_i \cdot x_{i,m-1})\Delta + \sigma_i \sqrt{x_{i,m-1}}\sqrt{\Delta} N_{i,m} \quad (i = 1, 2) \tag{45}$$

where the $N_{i,m}$ (fixed i or alternatively fixed m) are i.i.d. $N(0,1)$. Plugging the computed values into (35), resp. (36), returns the desired interest rates, resp. bond prices.

5.3 Stocks

From (41), we obtain the Euler-scheme

$$\ln S_{j,m} - \ln S_{j,m-1} = \mu_j \cdot \Delta + \sigma_j \cdot \sqrt{\Delta} \cdot N_{j+2,m} \quad (1 \le j \le d) \tag{46}$$

where the $N_{j+2,m}$ (fixed j) are i.i.d. $N(0,1)$. This implies for $M \in \mathbb{N}^+$

$$S_{j,M} = S_{j,0} \exp\left(\mu_j M\Delta + \sigma_j \sum_{m=1}^{M} \sqrt{\Delta} N_{j+2,m}\right). \tag{47}$$

6 Estimation of parameters

6.1 Interest rates and bonds

Parameter estimation for the CIR-2 model and detailed descriptions of the used methods are subject of several existing articles, e.g. Chen and Scott (1993), Duan and Simonato (1999), Bolder (2001), Beletsky and Szimayer (2002) and Fischer, May and Walther (2004). The problem is not trivial. The most efficient method seems to be pseudo maximum-likelihood estimation with Kalman-filtering. In particular, we used the machinery as explained in Fischer, May and Walther (2004). The interested reader can find further information in this paper and the references therein.

A comment on the data: we use the historical yield structure of the German debt securities market (monthly, taken at the end of each month). The values for spot rates with maturities $\tau > 0$ up to 28 years can be computed via a parametric presentation of yield curves (the so-called Svensson-method; cf. Svensson (1994) and Schich (1997)) for which the historical parameters can be taken from the homepage of the German Federal Reserve (*Deutsche Bundesbank*; `http://www.bundesbank.de`). The implied Bundesbank values R' are estimates of *discrete* interest rates on notional zero-coupon bonds based on German Federal bonds and treasuries (cf. Schich, 1997) and have to be converted into continuous interest rates by $R = \ln(1 + R')$.

6.2 Stocks

Given the market data $S_{j,m}$ ($j = 1, \ldots, d$; $m = -M, \ldots, 0$; time step $= \Delta$; $t = 0$ is the present), Equation (46) is used to obtain the estimators

$$\hat{\mu}_j = \frac{1}{M\Delta} \ln(S_{j,0}/S_{j,-M}) \tag{48}$$

and

$$\hat{\sigma}_j = \sqrt{\frac{1}{M\Delta} \sum_{i=1-M}^{0} (\ln(S_{j,m}/S_{j,m-1}) - \Delta\hat{\mu}_j)^2} \tag{49}$$

for the parameters of the stock price dynamics.

6.3 The covariance matrix

After having plugged in historical data, solving the equations (45) and (46) for $N_{i,m}$ gives us a time series $(N_{i,m})$ ($i = 1, \ldots, d+2$; $m = 1-M, \ldots, 0$) of hypothetical historical realizations of the normal random variables (44). The values

$$\widehat{\Sigma}_{i,j} = \frac{1}{M} \sum_{m=1-M}^{0} N_{i,m} N_{j,m} \tag{50}$$

can be used as estimates for the entries of the covariance matrix Σ. However, there is still something missing since we can not get the historical realizations $x_{i,m}$ of the stochastic factors of the interest rate model (45) directly from the market. Instead, we use the affine term structure (35) to derive them from the interest data. One has

$$\begin{array}{cccccccl}
\underbrace{\begin{pmatrix} R(t,\tau_1) \\ R(t,\tau_2) \end{pmatrix}} & = & \underbrace{\begin{pmatrix} -\frac{\log A_1(\tau_1)}{\tau_1} & -\frac{\log A_2(\tau_1)}{\tau_1} \\ -\frac{\log A_1(\tau_2)}{\tau_2} & -\frac{\log A_2(\tau_2)}{\tau_2} \end{pmatrix}} & + & \underbrace{\begin{pmatrix} \frac{B_1(\tau_1)}{\tau_1} & \frac{B_2(\tau_1)}{\tau_1} \\ \frac{B_1(\tau_2)}{\tau_2} & \frac{B_2(\tau_2)}{\tau_2} \end{pmatrix}} & \cdot & \underbrace{\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}} & (51) \\
R_t & = & M_A & + & M_B & \cdot & x(t). & (52)
\end{array}$$

From

$$x(t) = M_B^{-1}(R_t - M_A) \tag{53}$$

we obtain a time series $x_{i,m}$ ($i = 1, 2$; $m = -M, \ldots, 0$) by inserting the time series of the respective spot rates into (53). Slightly different from Fischer, May and Walther (2004), our suggestion is

$$\tau_1 = 0.5 \text{ years}, \quad \tau_2 = 10.0 \text{ years}. \tag{54}$$

Equation (53) also returns the starting values $x(0) = (x_1(0), x_2(0))$ for the simulation of the factors x_1 and x_2. The calculation of the values $x(0)$

implies a mathematically continuous continuation of the history of the spot rates $R(., \tau_1)$ and $R(., \tau_2)$ by the CIR-2 model. For other maturities than τ_1 and τ_2 there might be jumps in the respective sport rate (cf. Fischer, May and Walther, 2004). A simulation study of the same authors showed that for realistic time horizons the starting values have significant influence on the means of the simulated interest rates. Hence, a proper calculation of starting values is important.

Having executed the explained procedure, one can calculate the empirical covariance matrix $\widehat{\Sigma}$ by (50). At this point, a further problem arises. The CIR-2 model works with *uncorrelated* Brownian motions (cf. subsection 4.1). Nonetheless, the upper left 2×2-submatrix of $\widehat{\Sigma}$, which theoretically should be the two-dimensional identity, may differ from the theoretical values. To stay in the proposed model, one can adjust the estimate $\widehat{\Sigma}$ by setting the upper left 2×2-submatrix to the identity matrix. Doing this, it is important to check whether the new matrix is still positively definite as we afterwards have to carry out the Cholesky decomposition. In cases where positive definiteness gets lost, one should choose a symmetric positively definite matrix close to the proposed matrix with the identity in the upper left corner.

The proposed technique for the calculation of the covariance matrix and the starting values should be suitable for any stochastic interest rate model with an affine term structure (e.g. Vasicek-2).

7 Chronological overview

I. Estimation

• Get data • estimate parameters of stock prices; (48) and (49) • estimate parameters of interest rate dynamics (cf. Subsection 4.1) • compute the historical time series $x_{i,m}$ $(m \leq 0)$ by (53) • solve equations (46) and (45) for the historical $N_{j,m}$ $(m \leq 0)$ • compute the covariance matrix $\widehat{\Sigma}$; (50) • compute the Cholesky decomposition of $\widehat{\Sigma}$; (43)

II. Simulation

• Simulate future i.i.d. normal random variables and plug them into (44) to get the simulated $N_{i,m}$ $(m > 0)$ • plug the $N_{i,m}$ into (46) and (45) to get the simulated scenario of stock prices and interest rate model factors $x_{i,m}$ $(m > 0)$ • plug the factors $x_{i,m}$ into (35) or (36) to get spot rates or bond prices • reiterate the above three steps to get a large set of market scenarios

III. Evaluation
• Choose (or assume to be given) a certain portfolio • compute portfolio values (e.g. by (1)) using the scenarios generated in step II • compute the risk and performance measures (9), (11) and (13) by the empirical portfolio distributions obtained; cf. (29) to (31) • if necessary, compute the partial derivatives (21), (22) and (23)
IV. Optimization
• Use a GS or SI method repeating step III for each new portfolio

Note that step II must only be done once. The optimization iterations use the same set of scenarios for alternating portfolios.

8 First results

8.1 Gradient Search vs. Swarm Intelligence

(Particle) Swarm Intelligence is a powerful tool to solve optimization problems in a fixed search-space. SI is computationally appealing as simple to implement and computationally robust with respect to local minima and maxima, provided enough iterations are performed. As an additional bonus, SI is inherently suitable for parallel computing and can be implemented in a massively parallel way (cf. Auslander et al. (1995), Fabiunke (2002)). Gradient (Grid) Search methods are superior to random-guessing algorithms like SI if the search-space is e.g. a sphere, but on highly multi-dimensional surfaces, the gradient method gets stuck too often in local extreme points and therefore becomes computationally expensive, as one has to start from many different starting points. In higher-dimensional problems, SI seem fitter than GS methods. However, one has to be careful with such statements, as according to the No Free Lunch (NFL) theorem (cf. Wolpert and Macready, 1996), when performance is averaged over all possible search spaces, all search algorithms perform equally well.

Ultimately, we decided to stick with SI algorithms, which seem to converge faster for large real-life size portfolios. Combining these evolutionary algorithms with a selected Gradient Search for selected good intermediate solutions provides further speed-up. One should also mention that depending on the chosen market model it might be not easy to prove that the possibly for optimization purposes used derivatives (21), (22) and (23) really exist. This, however, is no real drawback when using a combined SI-GS-method since the GS-optimization routines are now subject to “back-testing” by the SI-methods. Problems coming in line with differentiation of quantile expressions could also be avoided by using risk measures which have more suitable differentiability properties like e.g. the risk measures proposed in Fischer (2003).

One caveat with all numerical solutions, without further assumptions about the search-space is that there is no guarantee that the optimal solution is found. In practice, one monitors the rate of convergence and dedicates enough search-time. Looking at the number of idle PCs and workstations in the typical investment bank or insurance company, one can be on the save side and farm out the work to a large number of processors or a dedicated cluster or Global Grid. However, it should also be mentioned that financial institutions often have no interest in global optimization, but local one since existing portfolios are often optimized in small steps and not by a complete restructuring.

8.2 Discussion of numeric outcomes

For our numeric examples the time horizon is one month where the simulation takes 20 steps per month. The number of iterations is 1000. We consider portfolios which have a present value of exactly 1000 EUR. We optimize using a local GS-method and a combined SI-GS-method. The second one is run with and without constraint $b = 0$, i.e. with and without shortselling in the portfolio (cf. (18)). The considered confidence level for ES is $\alpha = 5\%$. Two types of portfolios are examined. The smaller one contains two bonds and two stocks, the bigger one 10 bonds and 10 stocks. In particular, we considered the following bonds and stocks (which are here listed in the same order as in the portfolio vectors in the tables in the Section 11):

(A) 2 zero-coupon bonds: Maturity 1 year and 10 years.

2 “stocks”: Xetra DAX and Allianz

(B) 10 zero-coupon bonds: Maturity 1 year up to 10 years.

10 stocks: Allianz, BASF, BMW, Bayer, Commerzbank, Daimler-Chrysler, Deutsche Bank, Lufthansa, E.ON, Hypovereinsbank

All stocks are elements of the Xetra DAX and had their IPO at least 10 years ago. Data was taken from `http://de.finance.yahoo.com`. The estimates are calculated from monthly data from May 2002 to April 2003. We obtain the model parameters listed in Table 1. The same time interval and discretization was taken for the estimation of the term structure model parameters. Maturities from 1 to 10 years were taken into consideration. Results are in Table 2. The entries of the (adjusted) covariance matrix in Table 3 confirm the use of correlations between the interest rates and the stock market model factors to obtain a more realistic combined model.

For each of our setups we computed the ES-, ES-RORC- and ES-RORAC-optimal portfolio. The mean, VaR, ES, ES-RORC and ES-RORAC for 1-month returns these portfolios are listed in the Tables 4-9 in Section 11 (together with the optimal portfolios themselves in the four assets case). The optimized portfolios are compared with "normed" portfolios where the same capital is invested in each of the four, respectively 20 assets. As expected, the combined SI-GS-method is superior to the local GS-method starting at the normed portfolio. Local extreme points seem to exist in most of the considered cases. A situation as in Table 5 where the ES of the ES-optimized portfolio is higher than (but close to) the ES of the RORC-optimized portfolio could be a symptom for the need of more or finer iterations.

A reasonable optimization output is that the results imply that bonds of longer maturities bear more risks. This can be seen in decreasing weights of bonds with higher maturities in the ES-optimized portfolios (e.g. Tables 5 & 6). This was also observed for the portfolios with 20 assets which are not listed in detail. Massive short-selling and possible absence of global extreme points (e.g. in the RORAC case, cf. Table 4) strongly suggest the use of the constraint $b = 0$ (no short-selling). Roughly speaking, the implication seems to be that optimized portfolios under the constraint contain almost no stocks which is reasonable under the market conditions given between May 2002 and April 2003. A remarkable observation in the constrained case is that ES- and ES-RORC-optimal portfolios seem to be quite similar and prefer bonds at the short end, whereas the ES-RORAC-optimal portfolios tend to prefer the long end of the term structure (cf. Tables 6). This was also observed for portfolios with 10 bonds and 10 stocks. The reason must be that RORAC, in contrast to RORC, rewards under the assumption of the same given RORC assets with lower prices since prices are included in the denominator of its definition. This observation, although made for a very specific market situation, shows how sensitive optimization is towards the choice between the (on the first look quite similar) performance measures RORC and RORAC. Financial institutions should bear that in mind.

In summary, all obtained results seem to be reasonable from the economic point of view and confirm the proposed methods. We can not really judge the impact of the choice of the market model at this stage of our research. However, we guess that other reasonable models (e.g. such using Vasicek-2) would imply results similar to ours.

9 Parallel programming and scalability

9.1 Bulk synchronous parallel computing

Since 1944 von Neumann's model for sequential computing has been widely accepted, but there is no standard model for parallel computing. Most approaches nowadays are based on message-passing, but they are often inadequate, since the potential danger of deadlock, in which each possible activity is blocked, waiting on some other activity that is also blocked, increases dramatically with the complexity of software. Furthermore, models based on message-passing, e.g. MPI (Message passing interface), do not easily allow performance prediction. The Bulk synchronous parallel computing model (BSP) however abstracts low-level program structure in favour of so-called supersteps. This allows easy debugging, removes the problem of deadlock and allows a reasoning of the correctness of the code nearly as easily as in sequential code.

A *BSP computer* consists of a set of processor/memory pairs, a global communication network and a mechanism for the efficient barrier synchronisation of the processors. In real life, this could be anything: a single/multi-processor PC, a cluster of workstations or a real parallel machine like the Cray T3D.

The fundamental idea of BSP is the notion of a *superstep*. In a superstep, computation and communication are decoupled. This avoids deadlock. First the processes perform as many calculations as possible using their local data. If one processor needs data from another, communication starts only after all the computation has stopped. When communication is finished, barrier synchronisation is called and the next superstep begins.

9.2 Cost modelling and performance prediction

A cost model helps to guide the choice of programming algorithm.

The separation of communication from synchronisation and the inherent simplicity of the superstep structure make it relatively easy to find a suitable cost-model. The cost is expressed in terms of *steps* or *floating point operations (FlOps)* for each portion of the program. The cost parameters are the BSP parameters for the machine and parameters determined by the choice of algorithm and their implementation. As a BSP program consists of a sequence of supersteps, the “cost” of an entire program is the sum of the contributions from its supersteps.

What are the key parameters that determine performance? Extensive research by the originators of the BSPlib showed that the following four key parameters are sufficient (cf. Hill and McColl, 1996):

- the number of processors, p;
- processor speed, s (number of steps per second);
- the cost l (steps), of achieving barrier synchronisation (which depends on network latency, which is a measurement of delay from one end of a network to another). Basically l is the cost of telling all processors to wait till all communication has been performed; and
- the cost g (steps per word), of delivering message data. This captures the interprocess communication speed.

Since the processor speed s is essentially a normalising factor, there are only three independent parameters: p, l and g. The cost of one superstep is

$$\max(w_i) + g \cdot \max(h_i) + l \tag{55}$$

where i ranges over processors ($i = 1, \ldots, p$), w_i is the time for the local computation in processor i and h_i is the number of incoming or outcoming messages per processor. The values of the parameters are determined by measurement using suitable benchmarks that mimic average computation and communication loads (cf. Hill, 1996). The dependence on a specific platform enters the cost function only through the parameters p, l and g.

We follow convention and count every floating point operation as 1.

The BSP approach offers a simple cost model. In general, cost-modeling applications give a rough ball-park figure of the cost on any parallel machine and configuration size. The role of profiling tools like bsprof aids simplistic pencil and paper cost modeling, and it effectively predicts the cost of an algorithm on any parallel machine (cf. Hill, Crumpton and Burgess, 1996).

9.3 Scalability of optimization algorithms

Programmers take the burden of writing parallel programs to increase speed and memory. The aim of every parallel algorithm designer is to write code that scales linearly, i.e. runs p times as fast on a p-processor machine. This clearly constitutes an upper bound, if the sequential algorithm is already optimal. Linear scalability is achieved by using good load-balancing, keeping all processors busy all the time and communication costs are minimized.

Data dependency can make optimal speed-up impossible. It determines parallel complexity, the minimum number of steps an algorithm would need to run on a PRAM-computer. This constitutes an upper bound on the maximal speed-up that can be achieved.

There are many different and more sophisticated layouts of parallel implementations possible. The right choice depends on the size of the portfolio and available hardware. For the sake of simplicity in this article we have chosen the brute force approach.

Sketch of the scalability for a parallel brute-force GS: To avoid local extrema, one has to start many times from different grid-points:

1. Superstep: Broadcast the initial portfolio structure and search-areas, or only the portfolio structure and use random startpoints. Depending on the network architecture (reflected in the value of g), one might use several supersteps and use e.g. a tree-shaped communication form. Asymptotic cost for a 1 phase broadcast: $l + npg$, where n is the size of the initial portfolio structure.

2. Superstep: Now work out gradient searches on all processors for a given time. E.g. every processors performs a set-number of searches. On average this will balance out. Asymptotic cost: $1/p\times$ sequential time, as if p processors work out k/p searches, k searches are performed in total. The sequential time is the all dominating factor.

3. Superstep: Each processor sends its best grid point back to processor 1, which sorts them and gives the final result. Asymptotic cost: $l + ng + p$. The extra p arise from chosing the point with best fitness.

As the communication cost, sorting, etc. is negligible for any reasonable number of searches, this algorithm clearly scales linearly with the number of processors used.

Sketch of the scalability for SI:

> Superstep: As in the 1-processor mode (see Figure 2), but now performed on all p processors. Every 1000 or 10000 iterations fit values are exchanged, then the next superstep starts.

Since the cost of data interchange is negligible compared to the cost of the iterations in each superstep, we have scalability as in the GS case.

One typical schoolbook error in this context is not to use a high quality random number generator, assuring independent random number streams on all processors (cf. Mascagni, Ceperley and Srinivasan (1998, 1999)).

Large clusters as well as the rise of grid-computing requires analytic forecastig of run-times to chose the appropriate hardware for the task. There are many potential trade-offs (cf. Jarvis et al. (2002, 2003) and Roehrl (1998)): time versus money, etc. Our paper has shown that a pragmatic approach can take advantage of developments in computerscience to enable the exploration of new portfolio optimization techniques using parallel computing techniques.

10 Conclusion

In Section 1 to 3 we described two general methodologies of ES-, ES-RORC- and ES-RORAC-optimization: gradient search and swarm intelligence optimization, both suitable independently from the considered market model. Later, we proposed a particular market model which seems to be suitable to describe at the same time bonds and stocks, as well as dependencies between them. As we know, practicioners have been using similar models to forecast separately bonds and stocks. The model was used for our numerical examples with German market data. We thoroughly explained the proposed model and the respective simulation and optimization procedures. The economically reasonable results from our numeric examples and a scalability analysis show the suitability and practicability of the proposed methodology. Furthermore, differences between RORAC- and RORC-optimization have been demonstrated. The impact of alternative models could not be considered and leaves questions open for future research.

11 Tables and figures

Share	$V_i(0)$	$\hat{\mu}_i$	$\hat{\sigma}_i$
Xetra DAX	2942.04	-0.54	0.45
Allianz	56.17	-1.46	0.78
BASF	38.16	-0.29	0.31
BMW	29.06	-0.50	0.32
Bayer	16.75	-0.80	0.66
Commerzbank	8.32	-0.92	0.79
DaimlerChrysler	28.90	-0.65	0.36
Deutsche Bank	44.86	-0.59	0.47
Lufthansa	8.79	-0.56	0.48
E.ON	41.80	-0.31	0.29
Hypovereinsbank	10.20	-1.41	0.92

Table 1: Stock market parameters (05/2002 - 04/2003). In the portfolio with 4 assets, the Xetra DAX is treated like a single stock.

$\hat{a}_1$	$\hat{b}_1$	$\hat{\sigma}_1$	$\hat{\lambda}_1$
0.2648	0.0120	0.1236	-0.0647

$\hat{a}_2$	$\hat{b}_2$	$\hat{\sigma}_2$	$\hat{\lambda}_2$
1.7563	0.0145	0.1704	0.4968

Table 2: Estimates for the CIR-2-model (05/2002 - 04/2003)

$$\begin{pmatrix} 1 & 0 & 0.7333 & 0.5860 \\ 0 & 1 & -0.4180 & -0.3799 \\ 0.7333 & -0.4180 & 1 & 0.9062 \\ 0.5860 & -0.3799 & 0.9062 & 1 \end{pmatrix}$$

Table 3: Adjusted covariance matrix $\hat{\Sigma}$ (4 assets)

Portfolio	Mean	VaR	ES	ES-RORC	ES-RORAC
normed	-29.21	148.56	169.90	-0.1719	-0.0250
Units	(256.03; 382.88; 0.0850; 4.4508)				
Capital	(250.0; 250.0; 250.0; 250.0)				
ES-opt.	2.20	8.02	11.67	0.1888	0.0022
Units	(575.22; 595.47; 0.0284; -0.6059)				
Capital	(561.67; 388.81; 83.54; -34.03)				
RORC-opt.	8.73	18.61	28.39	0.3071	0.0085
Units	(506.74; 650.08; 0.0856; -3.0470)				
Capital	(494.81; 424.47; 251.87; -171.15)				
RORAC-opt.	14,535.32	34,795.68	56,428.72	0.2576	0.2531
Units	(89,585.98; -170,200.78; 102.93; -4,952.51)				
Capital	(87,476.85; -111,132.65; 302,838.23; -278,182.42)				

Table 4: 4 assets; $\alpha = 0.05$; no constraints; locally GS-optimized portfolios; GS started at normed portfolio

Portfolio	Mean	VaR	ES	ES-RORC	ES-RORAC
normed	-29.21	148.56	169.90	-0.1719	-0.0250
Units	(256.03; 382.88; 0.0850; 4.4508				
Capital	(250.0; 250.0; 250.0; 250.0)				
ES-opt.	2.75	1.11	2.19	1.2555	0.0027
Units	(1,758.10; -1,049.55; -0.0115; 0.0413)				
Capital	(1,716.71; -685.31; -33.73; 2.32)				
RORC-opt.	3.07	0.84	1.96	1.5643	0.0031
Units	(1,533.04; -730.19; -0.0036; -0.1729)				
Capital	(1,496.95; -476.78; -10.46; -9.71)				
RORAC-opt.	29,454.12	70,484.98	114,339.65	0.2576	0.2554
Units	(186,074.79; -353,548.05; 208.71; -10,038.90)				
Capital	(181,693.00; -230,849.31; 614,040.41; -563,885.10)				

Table 5: 4 assets; $\alpha = 0.05$; no constraints; SI-GS-optimized portfolios

Portfolio	Mean	VaR	ES	ES-RORC	ES-RORAC
normed	-29.21	148.56	169.90	-0.1719	-0.0250
Units	(256.03; 382.88; 0.0850; 4.4508)				
Capital	(250.0; 250.0; 250.0; 250.0)				
ES-opt.	1.81	4.19	6.26	0.2894	0.0018
Units	(1,020.03; 0.22; 0.0013; 0.0001)				
Capital	(996.01; 0.14; 3.84; 0.01)				
RORC-opt.	1.88	4.29	6.37	0.2957	0.0019
Units	(1,020.60; 4.22; 0.0000; 0.0119)				
Capital	(996.57; 2.76; 0.01; 0.67)				
RORAC-opt.	2.13	19.43	25.32	0.0843	0.0021
Units	(84.47; 1,404.48; 0.0001; 0.0027)				
Capital	(82.48; 917.06; 0.31; 0.15)				

Table 6: 4 assets; $\alpha = 0.05$; constraint $b = 0$; SI-GS-optimized portfolios

Portfolio	Mean	VaR	ES	ES-RORC	ES-RORAC
normed	-19.29	115.36	134.59	-0.1433	-0.0170
ES-opt.	0.32	6.93	8.87	0.0361	0.0003
RORC-opt.	7.13	8.28	13.79	0.5172	0.0070
RORAC-opt.	653.55	884.70	1,369.21	0.4773	0.2759

Table 7: 20 assets; $\alpha = 0.05$; no constraints; locally GS-optimized portfolios; GS started at normed portfolio

Portfolio	Mean	VaR	ES	ES-RORC	ES-RORAC
normed	-19.29	115.36	134.59	-0.1433	-0.0170
ES-opt.	2.52	2.23	3.56	0.7064	0.0025
RORC-opt.	4.40	2.92	4.82	0.9129	0.0044
RORAC-opt.	651.75	842.94	1,357.35	0.4802	0.2765

Table 8: 20 assets; $\alpha = 0.05$; no constraints; SI-GS-optimized portfolios

Portfolio	Mean	VaR	ES	ES-RORC	ES-RORAC
normed	-19.29	115.36	134.59	-0.1433	-0.0170
ES-opt.	-2.10	18.59	22.75	-0.0922	-0.0021
RORC-opt.	-2.22	23.76	27.84	-0.0799	-0.0021
RORAC-opt.	-2.1723	19.01	23.17	-0.0937	-0.0021

Table 9: 20 assets; $\alpha = 0.05$; constraint $b = 0$; SI-GS-optimized portfolios

References

[1] Acerbi, C., Tasche, D. (2002) - *On the coherence of Expected Shortfall*, Journal of Banking and Finance 26 (7), 1487-1503

[2] Artzner, P., Delbaen, F., Eber, J., Heath, D. (1999) - *Coherent measures of risk*, Mathematical Finance 9 (3), 203-228

[3] Auslander, J., Fukunaga, A., Partovi, H., et al. (1995) - *Further Experience with Controller-Based Automatic Motion Synthesis for Articulated Figures (1995)*, ACM Transactions on Graphics

[4] Babbs, S.H., Nowman, K.B. (1998) - *Econometric Analysis of a Continuous Time Multi-Factor Generalized Vasicek Term Structure Model: International Evidence*, Asia-Pacific Financial Markets 5, 159-183

[5] Beletski, T., Szimayer, A. (2002) - *Estimating Exponential-Affine Term Structure Models by Kalman Filtering: Studying the DEM/Euro-Market*, Arbeitspapier, Forschungszentrum caesar, Bonn

[6] Bolder, D. (2001) - *Affine Term-Structure Models: Theory and Implementation*, Working Paper 2001-15, Bank of Canada

[7] Chen, R., Scott, L. (1992) - *Pricing Interest Options in a Two-Factor Cox-Ingersoll-Ross Model of the Term Structure*, The Review of Financial Studies 5, No. 4, 613-636

[8] Chen, R., Scott, L. (1993) - *Maximum Likelihood Estimation For A Multifactor Equilibrium Model Of The Term Structure Of Interest Rates*, Journal of Fixed Income 4, 14-31

[9] Cox, J., Ingersoll, J., Ross, S. (1985) - *A theory of the term structure of interest rates*, Econometrica 53, 385-407

[10] Deutsche Bundesbank (1997) - *Schätzung von Zinsstrukturkurven*, Monatsbericht Oktober, 61-66

[11] Duan, J., Simonato, J. (1999) - *Estimating Exponential-Affine Term Structure Models by Kalman Filter*, Review of Quantitative Finance and Accounting 13, 111-135

[12] Fabiunke, M. (2002) - *A Swarm Intelligence Approach to Constraint Satisfaction*, Proceedings of the conference on Integrated Design and Process Technology, June

[13] Fischer, T. (2003) - *Risk capital allocation by coherent risk measures based on one-sided moments*, Insurance: Mathematics and Economics 32 (1), 135-146

[14] Fischer, T., May, A., Walther, B. (2004) - *Anpassung eines CIR-k-Modells mit Hilfe der Kalman-Filter-Methode*, Blätter der DGVFM XXVI (3), 369-387

[15] Gouriéroux, C., Laurent, J., Scaillet, O. (2000) - *Sensitivity analysis of Values at Risk*, Journal of Empirical Finance 7, 225-245

[16] Kennedy, J., Eberhart, R. (1995) - *Particle swarm optimization*, Proc. IEEE International Conference on Neural Networks, Perth, Australia, Nov. 27-Dec. 1, 1942-1948.

[17] Kennedy, J., Russell, C., Eberhart, C., Yuhui, S. (2001) - *Swarm Intelligence*, Morgan Kaufmann

[18] Hill, J. (1996) - *BSP cost parameters, sorted by megaflop/s rate, for the Oxford BSP toolset*, Volume 4, Computing Laboratory, Oxford University

[19] Hill, J., Crumpton, P., Burgess, D. (1996) - *The theory, practice, and a tool for BSP performance prediction applied to a CFD application*, In Proc. Euro-Par '96. LNCS, Passau, Volume 1124, Springer

[20] Hill, J., McColl, W. (1996) - *Questions and answers about BSP*, Computing Laboratory, Oxford University

[21] Jarvis, S., Spooner, D., Nudd, G., Cao, J, Saini, S. (2003) - *Local Grid Scheduling Techniques using Performance Prediction*, Computers and Digital Techniques, 150 (2), 87-96

[22] Jarvis, S., Hill, J., Vasilev, V., Siniolakis, C. (2002) - *Portable and architecture independent parallel performance tuning using BSP*, Parallel Computing, 28 (11), 1587-1609

[23] Lemus Rodriguez, G.J. (1999) - *Portfolio optimization with quantile-based risk measures*, Ph.D. thesis, Massachusetts Institute of Technology

[24] Mascagni, M., Ceperley, D., Srinivasan, A. (1998) - *SPRNG: User's guide*, http://www.ncsa.uiuc.edu/Apps/SPRNG/www/generators.html

[25] Mascagni, M., Ceperley, D., Srinivasan, A. (1999) - *Random Number Generators for Parallel Application*, Monte Carlo Methods in Chemical Physics, Volume 205

[26] Rockafellar, R.T., Uryasev, S. (2000) - *Optimization of Conditional Value-At-Risk*, Journal of Risk 2 (3), 21-41

[27] Rockafellar, R.T., Uryasev, S. (2002) - *Conditional Value-at-Risk for General Loss Distributions*, Journal of Banking and Finance 26 (7), 1443-1471

[28] Roehrl, A. (1998) - *Fast, portable, predictable and scalable bootstrapping*, Computer Science and Statistics: Proceedings of the Conference on the Interface

[29] Schich, S.T. (1997) *Schätzung der deutschen Zinsstrukturkurve*, Diskussionspapier 4/97, Volkswirtschaftliche Forschungsgruppe der Deutschen Bundesbank

[30] Shibanai, Y., Yasuno, S. Ishiguro, I. (2001) - *Effects of Global Information Feedback on Diversity: Extensions to Axelrod' s Adaptive Culture Model*, Journal of Conflict Resolution, 45, 80-96.

[31] Svensson, L. (1994) - *Estimating and interpreting forward interest rates: Sweden 1992 - 94*, IWF Working Paper 114, September

[32] Tasche, D. (2000) - *Risk contributions and performance measurement*, Working Paper, TU München, http://www-m4.mathematik.tu-muenchen.de/m4/pers/tasche/riskcon.pdf

[33] Wolpert, D., Macready, W. (1996) - *No Free Lunch Theorems for Optimization*, IEEE Transactions on Evolutionary Computation