Application of Multistage Stochastic Programs Solved in Parallel in Portfolio Management

Maria Lucka [a], Igor Melichercik [b], Ladislav Halada [c]

[a]*Faculty of Education, University of Trnava, Priemyselna 4, 918 43 Trnava, Slovakia*

[b]*Department of Applied Mathematics and Statistics, Faculty of Mathematics, Physics and Informatics, Comenius University, Mlynska Dolina, 842 48 Bratislava, Slovakia*

[c]*Institute for Informatics, Slovak Academy of Sciences, Dubravska 9, 842 37 Bratislava, Slovakia*

Abstract

We present a multistage model for allocation of financial resources to bond indices in different currencies. The model was tested on historical data of interest and exchange rates. We compare a two-stage and a three-stage stochastic programming model from a financial performance point of view.

For solving two-stage and three-stage stochastic programs the interior point method (IPM) in the frame of the primal-dual path following formulation is used. An application of the Birge and Qi factorization to the IPM allows decomposition of large linear system to smaller blocks allowing thus to solve it in parallel.

The parallel code is written in the Fortran programming language, using the Message Passing Interface (MPI) for communication. Parallel and financial performance is illustrated on experiments executed on the IBM 1350 Linux cluster.

Keywords: portfolio management, interior point method, parallel computing, large-scale optimization, multistage stochastic programming

Introduction

Financial institutions face the problem of optimal portfolio decisions under uncertainty. This uncertainty can be represented by a set of possible scenarios of market development. The mathematical framework for optimizing the portfolio decisions could be found in several models. One important class of models represents single period models based on the idea of mean-variance optimization of Markowitz [26]. The important examples are expected utility maximization of Ingersoll [21] and mean absolute deviation minimization of Sharpe [33], Konno and Yamazaki [23], Zenios and Kang [36]; see, e.g., Zenios [38] for classification of these models.

Recently, an interest in the development of multiperiod models of bond portfolio management has been observed. At this place we refer to successful and valuable contributions of Bradley and Cane [7], Kusy and Ziemba [24], Dupačová [11], Dantzig and Infanger [10], Mulvey [28], Hiller and Eckstein [19], Zenios [37], Carino et al. [8], Golub et al. [13], Frauendorfer and Schürle [12].
In Holmer et al. [20] a multiperiod dynamic model for fixed-income portfolio management under uncertainty, using multistage stochastic programming was developed. The scenarios of term structure development were obtained by Monte Carlo simulations. Their results confirmed that multiperiod models outperform single period ones. A multiperiod dynamic model for international bond portfolios was formulated in Beltratti et al. [3].

In [18] we formulated a model for allocation of financial resources to bond indices in different currencies. Such portfolios are the subject to an interest rate risk and an exchange rate risk. The stochastic properties of possible future development of exchange rates and interest rates are represented in the form of a scenario tree. The objective is to maximize the expected value of the portfolio at the defined time horizon. Mathematically, such models lead to the multistage stochastic programming. When one deals with several currencies, the number of scenarios per stage should be sufficient to model the randomness of interest rates in each currency and all exchange rates. Therefore, the realistic trees are ”bushy” and the number of scenarios grows exponentially with the number of stages. The computation of such problems could be extremely large and computationally intractable. Therefore, the parallel computing of such large size problem is preferred. Many studies have been published on how stochastic programming algorithm can be parallelized, such as [25,29,31] and [9,34,35].

We have used the interior point method in the frame of primal-dual path following formulation, due to a good possibility for parallelization. Moreover, the recursive structure of the constraint matrices for two and three stage problems and their Birge and Qi (BQ) factorization has been exploited for modular and independent computation. We have used the factorization technique based on the work of Birge and Qi for two-stage stochastic program [6] and its parallelization suggested in [22]. Pflug and Halada [30] proposed a generalization of this type of factorization for multistage stochastic programs. The parallel computer program for three-stage stochastic program has been proposed, implemented and tested on examples with relative small number of scenarios in papers [4], [17]. Recent contributions to the application of interior point method and its parallelization in multistage stochastic programms can be found in [5], [14], [15], [16].

We have performed financial studies using historical data of interest rates and exchange rates. Two-stage and three-stage stochastic models with different quality of information about future development of financial markets have been compared. The aim of this analysis was to verify whether higher number of stages implies better quality of the decision process. This has been a continuation of [18] where we performed several preliminary calculations with a low number of scenarios and different scenario setting. The results indicated that the increasing number of stages could make sense and that the optimal decision is sensitive with respect to different generations of the scenario tree. Moreover we have augmented the existing three-stage parallel program with the sparse matrix representation needed for solving large optimization problems. However, no financial studies were performed in [18].

In this paper we include the existing parallel three-stage stochastic program into a more complex code, enabling thus testing the portfolio management model on historical data of indices. The scenarios of possible developments were generated by using the Monte Carlo method. The parallel three-stage stochastic algorithm based on the interior point method and using the parallel BQ factorization, allowed us to generate trees with a large number of scenarios.

The paper is organized as follows. The problem formulation with financial instruments, constraint equations and an objective function is formulated in Section 1. We describe the multistage stochastic program used for portfolio management in detail in Section 2. Section 3 contains the algorithm of calculation of the financial study. Parallel BQ factorization for the three-stage stochastic program is given in Section 4. Experimental results, parallel and financial performance of the proposed model are demonstrated in Section 5. Section 6 contains conclusions and final remarks.

1 Problem formulation

The stochastic properties are represented in the form of a scenario tree. Denote by $\mathcal{F}_\tau,\ 1 \le \tau \le \mathcal{T}$ the set of nodes in period τ. For any $\omega \in \mathcal{F}_\tau,\ 1 < \tau \le \mathcal{T}$, there is a unique element $a(\omega) = \omega' \in \mathcal{F}_{\tau-1}$, which is the unique predecessor of ω. Each stage has the length of one period and the decision variables of the stage $\tau,\ 1 \le \tau < \mathcal{T}$ correspond to the beginning of the period τ.

1.1 Financial instruments

Denote the decision variables (for periods $1 \le \tau < \mathcal{T}$) of this process as
$b_j^{(\tau)}(\omega^\tau),\ \omega^\tau \in \mathcal{F}_\tau$: The amount of the index j bought in period τ,
$s_j^{(\tau)}(\omega^\tau),\ \omega^\tau \in \mathcal{F}_\tau$: The amount of the index j sold in period τ,
$h_j^{(\tau)}(\omega^\tau),\ \omega^\tau \in \mathcal{F}_\tau$: The amount of the index j hold in period τ,

and constants (for periods $1 \leq \tau \leq \mathcal{T}$) as
$c^{(0)}$: Initial cash available,
$h_j^{(0)}$: Composition of the initial portfolio,
$\gamma_j^{(\tau)}(\omega^\tau),\ \omega^\tau \in \mathcal{F}_\tau$: The exchange rate base currency/ j-th currency ($\gamma_1^{(\tau)} = 1\ \forall\tau$),
$\vartheta_j^{(\tau)}(\omega^\tau),\ \omega^\tau \in \mathcal{F}_\tau$: The value of the j-th index in the j-th currency,
$\upsilon_j^{(\tau)}(\omega^\tau),\ \omega^\tau \in \mathcal{F}_\tau$: The value of the j-th index in the base currency computed by

$$\upsilon_j^{(\tau)}(\omega^\tau) = \vartheta_j^{(\tau)}(\omega^\tau).\gamma_j^{(\tau)}(\omega^\tau)\,,$$

$\xi_j^{(\tau)}(\omega^\tau),\ \chi_j^{(\tau)}(\omega^\tau),\ \omega^\tau \in \mathcal{F}_\tau$: Bid and ask prices of the j-th index in the base currency computed by

$$\xi_j^{(\tau)}(\omega^\tau) = \upsilon_j^{(\tau)}(\omega^\tau)(1-\delta_j^{bid}),\ \chi_j^{(\tau)}(\omega^\tau) = \upsilon_j^{(\tau)}(\omega^\tau)(1+\delta_j^{ask}) \tag{1}$$

($\delta_j^{ask} > 0$ and $\delta_j^{bid} > 0$ are transaction costs for buying and selling).

1.2 Constraints

The amount of bought and sold units of index should be nonnegative. We forbid short positions. Therefore the number of hold units is nonnegative.
$b_j^{(\tau)}(\omega^\tau) \geq 0, \quad s_j^{(\tau)}(\omega^\tau) \geq 0, \quad h_j^{(\tau)}(\omega^\tau) \geq 0 \quad \forall\, 1 \leq \tau < \mathcal{T}$ and $\omega^\tau \in \mathcal{F}_\tau$.

Possible restrictions for selling
A typical investor is conservative. He/she is not willing to sell a big part of the portfolio. Therefore one can add constraints allowing to sell only β part of any asset or β part of the whole portfolio:

$$\sum_j s_j^{(1)}\xi_j^{(1)} \leq \beta \sum_j h_j^{(0)}\xi_j^{(1)}$$

for the Period 1 and

$$\sum_j s_j^{(\tau)}(\omega^\tau)\xi_j^{(\tau)}(\omega^\tau) \leq \beta \sum_j h_j^{(\tau-1)}(a(\omega^\tau))\xi_j^{(\tau)}(\omega^\tau)$$

for the Period τ, where $1 < \tau < \mathcal{T}$.

Risk reduction
The terminal wealth calculation is given by:

$$WT(\omega^{\mathcal{T}}) = \sum_j \xi_j^{(\mathcal{T})}(\omega^{\mathcal{T}})h_j^{(\mathcal{T}-1)}(a(\omega^{\mathcal{T}})) \quad \forall\omega^{\mathcal{T}} \in \mathcal{F}_{\mathcal{T}}. \tag{2}$$

To reduce the risk one can add the constraint forbidding the terminal wealth to fall below some proper constant C:

$$WT(\omega^{\mathcal{T}}) \geq C \quad \forall \omega^{\mathcal{T}} \in \mathcal{F}_{\mathcal{T}}.$$

1.3 Equations

Inventory balance and cash-flow accounting for the Period 1

$$h_j^{(0)} + b_j^{(1)} - s_j^{(1)} = h_j^{(1)} \quad \forall j. \tag{3}$$

$$c^{(0)} + \sum_j \xi_j^{(1)} s_j^{(1)} = \sum_j \chi_j^{(1)} b_j^{(1)}. \tag{4}$$

Inventory balance and cash-flow accounting for the Period τ*, where* $1 < \tau < \mathcal{T}$*:*

$$h_j^{(\tau-1)}(a(\omega^\tau)) + b_j^{(\tau)}(\omega^\tau) - s_j^{(\tau)}(\omega^\tau) = h_j^{(\tau)}(\omega^\tau) \quad \forall \omega^\tau \in \mathcal{F}_\tau \quad \forall j. \tag{5}$$

$$\sum_j \xi_j^{(\tau)}(\omega^\tau) s_j^{(\tau)}(\omega^\tau) = \sum_j \chi_j^{(\tau)}(\omega^\tau) b_j^{(\tau)}(\omega^\tau) \quad \forall \omega^\tau \in \mathcal{F}_\tau. \tag{6}$$

1.4 Objective function

The objective function maximizes the expected terminal wealth. It can be written as:

$$\text{Maximize}\, E(WT)\,,$$

where

$$E(WT) = \sum_{\omega^{\mathcal{T}} \in \mathcal{F}_{\mathcal{T}}} \pi(\omega^{\mathcal{T}}) WT(\omega^{\mathcal{T}}) \tag{7}$$

($\pi(\omega^{\mathcal{T}})$ is the probability of the scenario $\omega^{\mathcal{T}}$). Using (2) the expected terminal wealth (7) could be written in the form

$$\begin{aligned} E(WT) &= \sum_{\omega^{\mathcal{T}} \in \mathcal{F}_{\mathcal{T}}} \pi(\omega^{\mathcal{T}}) \sum_j \xi_j^{(\mathcal{T})}(\omega^{\mathcal{T}}) h_j^{(\mathcal{T}-1)}(a(\omega^{\mathcal{T}})) \\ &= \sum_{\omega^{\mathcal{T}} \in \mathcal{F}_{\mathcal{T}}} \sum_j \pi(\omega^{\mathcal{T}}) \xi_j^{(\mathcal{T})}(\omega^{\mathcal{T}}) h_j^{(\mathcal{T}-1)}(a(\omega^{\mathcal{T}})) \end{aligned}$$

$$= \sum_{\omega^{(\mathcal{T}-1)} \in \mathcal{F}_{(\mathcal{T}-1)}} \sum_j h_j^{(\mathcal{T}-1)}(\omega^{\mathcal{T}-1}) \tilde{c}_j(\omega^{\mathcal{T}-1}) \, , \tag{8}$$

where

$$\tilde{c}_j(\omega^{\mathcal{T}-1}) = \sum_{\omega^{\mathcal{T}} \in \mathcal{F}_{\mathcal{T}} : a(\omega^{\mathcal{T}}) = \omega^{\mathcal{T}-1}} \pi(\omega^{\mathcal{T}}) \xi_j^{(\mathcal{T})}(\omega^{\mathcal{T}}) \, . \tag{9}$$

1.5 Scenario tree

The set of scenarios could be generated in many ways. Monte Carlo simulations of the process with following properties is used for practical realization:

Property 1.
Between two successive periods the price process follows a log-normal process:

$$\upsilon_j^{(\tau+1)} = \upsilon_j^{(\tau)} \exp(\nu \, \triangle t + \sigma_j \sqrt{\triangle t} \, Z_j) \, , \tag{10}$$

where Z_j is a random variable with $N(0,1)$ distribution, ν an annual return and σ_j a volatility of the $j-$th index. The correlations $\mathrm{cor}\,(Z_i, Z_j)$ and volatilities of the indices are calibrated using historical data of the indices.

Property 2.
We suppose that the price process (10) between the periods τ and $\tau + 1$ is mean reverting to the prescribed price $P_j^{(\tau+1)}$ in period $\tau + 1$:

$$E(\upsilon_j^{(\tau+1)} | \upsilon_j^{(\tau)}) = P_j^{(\tau+1)}, \tag{11}$$

where $P_j^{(\tau)}$, $\tau = 2, 3, \ldots, \mathcal{T}$, are expected prices of the indices (in the base currency).
The condition (11) is fulfilled, if we take (10) with

$$\nu = \log \left(\frac{P_j^{(\tau+1)}}{\upsilon_j^{(\tau)}} \right) \frac{1}{\triangle t} - \frac{1}{2} \sigma_j^2 \, . \tag{12}$$

In our calculations we used two-stage and three-stage scenario trees of types $Sc_1 \times Sc_2$ and $Sc_1 \times Sc_2 \times Sc_3$, respectively. The scenario tree of type $Sc_1 \times Sc_2$, and $Sc_1 \times Sc_2 \times Sc_3$, respectively, means that there are Sc_1 possibilities of the price vector $\upsilon_i^{(2)}$ at the end of the first period, Sc_2 possibilities (for each position $\omega^2 \in \mathcal{F}_2$) of the price vector $\upsilon_i^{(3)}$ at the end of the second period and (in the three-stage case)

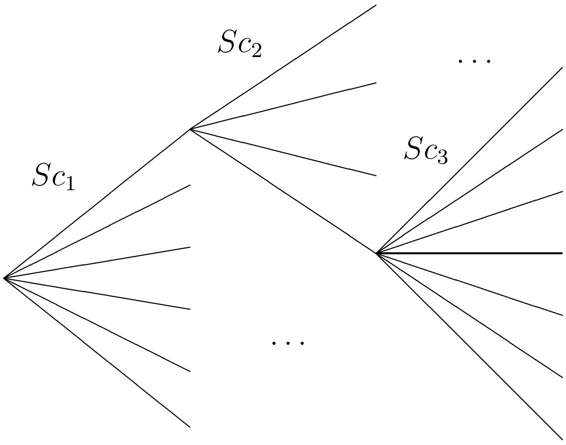

Fig. 1. Scenario tree of type $Sc_1 \times Sc_2 \times Sc_3$. The leaves are enumerated in this form for computational recursivity reasons between stages.

Sc_3 possibilities (for each position $\omega^3 \in \mathcal{F}_3$) of the price vector $v_i^{(4)}$ at the end of the third period.

We do not generate the scenarios in the last period in the practical realization. Using (9) and (11) one gets

$$\tilde{c}_j(\omega^{\mathcal{T}-1}) = \pi(\omega^{\mathcal{T}-1}) P_j^{\mathcal{T}} (1 - \delta_j^{bid})\,. \qquad (13)$$

Thus, instead of generating the prices $v_j^{\mathcal{T}}(\omega^{\mathcal{T}})$ we use their mean values $P_j^{\mathcal{T}}$. Therefore, hereafter we denote the types of the three-stage and two-stage trees by $Sc_1 \times Sc_2 \times *$ and $Sc_1 \times *$, respectively. Denote by n the number of indices. One has to perform $Sc_1.Sc_2.n^2$ Monte Carlo simulations to generate a three-stage scenario tree and $Sc_1.n$ Monte Carlo simulations to generate a two-stage scenario tree. Concerning the number of variables, for each decision node one has $3n$ decision variables (n indices, for each index buy, sell and hold decisions). Thus, a three-stage scenario tree implies $3n(1 + Sc_1 + Sc_1.Sc_2)$ decision variables, a two-stage scenario tree $3n(1 + Sc_1)$ decision variables.

2 Multistage stochastic linear program

The mathematical representation of the financial problem described above leads to the multistage stochastic linear program

$$maximize\ c^T x,\ subj.\ to\ Ax = h\,. \qquad (14)$$

The right-hand side vector h has the form

$$h = (c^{(0)}, -h_1^{(0)}, -h_2^{(0)}, \ldots, -h_n^{(0)}, 0, \ldots, 0)^T \qquad (15)$$

where $c^{(0)}$ is the initial cash available and $h_j^{(0)}$, $j = 1, 2, \ldots, n$ is the composition of the initial portfolio. The time periods in the tree are arranged in the increasing order from the root of the tree up to the leaves. While the computation is executed recursively starting from leaves to the root, we will enumerate the computational levels (cl) successively from the leaves to the root. As a consequence, the time periods and the computational levels have the opposite direction. A simple lattice for a three-stage problem composed from two types of two-stage programs is illustrated in Fig.2. BQS(2,j) represents the two-stage problem for nodes j, $j = 1, 2, 3$, at the second computational level. On the basis of results proved in [30], BQS(3,1) represents again a two-stage problem for the node at the third computational level that can be solved by the BQ factorization when the results of the solver BQS(2,j), $j = 1, 2, 3$, are available. An advantage of this model is the possibility to parallelize the computation on many levels without communication.

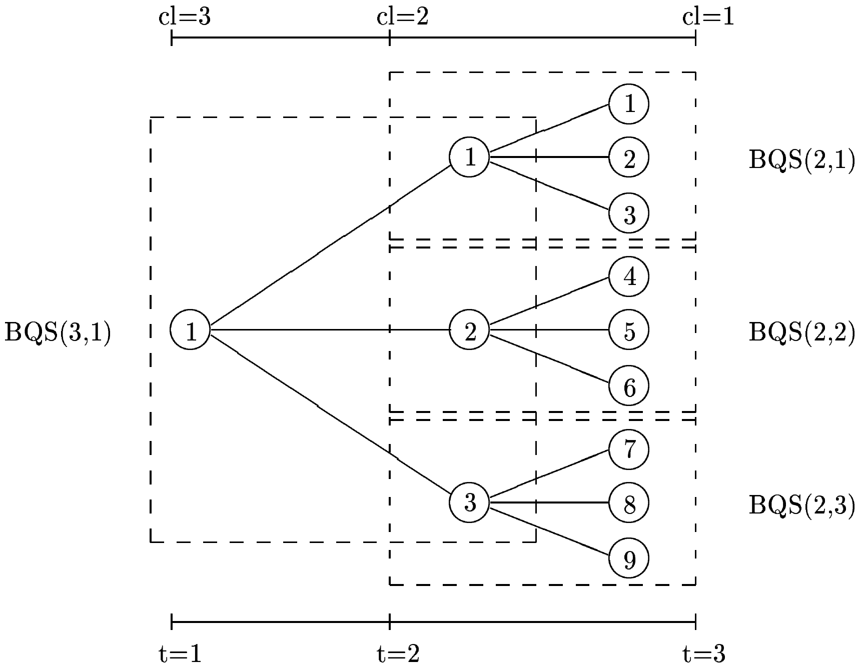

Fig. 2. Scenario tree of a three-stage program

The matrix constraint matrix $A = A^{(3)}$ referring to the three-stage stochastic program has the following form:

$$A^{(3)} = \begin{pmatrix} A_0^{(3)} & & & & & & \\ T_1^{(3)} & A_1^{(2)} & & & & & \\ T_2^{(3)} & & A_2^{(2)} & & & & \\ T_3^{(3)} & & & A_3^{(2)} & & & \\ \cdots & & & & \cdots & & \\ \cdots & & & & & \cdots & \\ T_{Sc_1}^{(3)} & & & & & & A_{Sc_1}^{(2)} \end{pmatrix} \tag{16}$$

where $A_k^{(2)}, k = 1, 2, ...Sc_1$, represent matrices corresponding to Sc_1 two-stage problems in the frame of the whole three-stage problem. The k-th two-stage problem, $k = 1, 2, \ldots, Sc_1$, is determined by the matrix $A_k^{(2)}$,

$$A_k^{(2)} = \begin{pmatrix} A_{k,0}^{(2)} & & & & & & \\ T_{k,1}^{(2)} & A_{k,1}^{(1)} & & & & & \\ T_{k,2}^{(2)} & & A_{k,2}^{(1)} & & & & \\ T_{k,3}^{(2)} & & & A_{k,3}^{(1)} & & & \\ \cdots & & & & \cdots & & \\ \cdots & & & & & \cdots & \\ T_{k,Sc_2}^{(2)} & & & & & & A_{k,Sc_2}^{(1)} \end{pmatrix} \tag{17}$$

In the case of four indices the matrices $A_0^{(3)}$, $A_{k,0}^{(2)}$, $A_{k,r}^{(1)}$, $k = 1, 2, \ldots Sc_1$, $r = 1, 2, \ldots, Sc_2$, have the form:

$$\begin{pmatrix} \chi_1 & \chi_2 & \chi_3 & \chi_4 & -\xi_1 & -\xi_2 & -\xi_3 & -\xi_4 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \end{pmatrix} \tag{18}$$

where $\xi_j = \xi_j^{(\tau)}(\omega^\tau)$, $\chi_j = \chi_j^{(\tau)}(\omega^\tau)$, $j = 1, 2, 3, 4$ represent the bid and ask prices of the indices in the corresponding nodes. The matrices $T_{k,r}^{(2)}$, $k = 1, 2, \ldots Sc_1$,

$r = 1, 2, \dots Sc_2$, have the form:

$$T_{k,r}^{(2)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{19}$$

The matrices $T_k^{(3)}$, $k = 1, 2, \dots Sc_1$ with $5 + 5Sc_2$ rows have the form:

$$T_k^{(3)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & \cdot & \cdot & \cdot & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{20}$$

In the case of the two-stage problem (14) the constraint matrix $A = A^{(2)}$ has the structure similar to $A_k^{(2)}$ and the vector h has a form conforming to matrices $A_k^{(2)}$.

3 Portfolio management modeling using historical data of indices

The proposed portfolio management model was tested using historical daily data of indices and exchange rates from January 1997 (start day: January 6, 1997) to January 2003 (end day: January 6, 2003). At the beginning there was a cash available and the portfolio was bought according to the result of the multistage stochastic program. The portfolio was then daily rebalanced according to the results of the proposed model. The financial study is calculated using the following algorithm:

```
for $D = start\_day, end\_day$ do
  (i) read the values of indices in the corresponding days
      (values $I_j^{(\tau)}, \tau = 1, 2, 3, 4$ used in (36), and values for calculation of the past
      returns $\rho_j^{(-3)}$, $\rho_j^{(-2)}$ and $\rho_j^{(-1)}$ in (37), respectively)
```

(ii) according to equations(10), (11), (12) and (36) ((37), respectively) calculate the expected prices in the determined time periods and generate the scenario-tree

(iii) calculate the bid and ask prices according to (1)

(iv) calculate the vectors h and c and non-zero elements of the matrix A

(v) solve the optimization problem

$$maximize\ c^T x,\ subj.\ to\ Ax = h \tag{21}$$

(vi) determine the composition of the portfolio

(vii) read the values of the indices for $D+1$ and calculate the value of the portfolio

end do D

In the case of the three-stage stochastic model the matrix $A = A^{(3)}$ can be established according to the equations (3) - (6). The right-hand side vector $h = (0, -x(9), -x(10), -x(11), -x(12), 0...0)^T$, whereby $x(j), j = 9, 10, 11, 12$, are the values of the solution vector x of (14) from the previous day. At the beginning for $D = start\_day$ is the vector $h = 0$, except the first element $h(1) = c^{(0)}$ (initial cash). The most computational demanding part of the modeling is the *solving of the optimization problem* (14).

There are many possibilities how the problem (14) can be solved. We have used the Interior Point Method (IPM) in the frame of the Mehrotra's Predictor Corrector algorithm MPC defined in [39], p.198. This algorithm, since 1990, has been the basis for most interior point software. For clarity reasons we present the basic ideas of this method, whereby the matrix A stands for the corresponding matrix $A^{(\tau)}$, where $\tau = 3$ for three-stage and $\tau = 2$ for two-stage stochastic programming model, respectively.
Given (x^0, y^0, z^0) with $x^0 > 0$, $z^0 > 0$, it finds the iterates $(x^{k+1}, y^{k+1}, z^{k+1})$, $k = 0, 1, 2...$, by solving the system

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x^{aff} \\ \Delta y^{aff} \\ \Delta z^{aff} \end{pmatrix} = \begin{pmatrix} r_c \\ r_h \\ r_\mu \end{pmatrix}, \tag{22}$$

where $r_h = h - Ax$, $r_c = c - z - A^T y$, $r_\mu = -XZe$; X and Z are diagonal matrices with diagonal entries x and z, respectively, and e denotes the vector having all entries equal one.
The solution of (22) is expressed as

$$\Delta y^{aff} = (ADA^T)^{-1}(r_h + AZ^{-1}(Xr_c - r_\mu)), \tag{23}$$

$$\Delta x^{aff} = Z^{-1}(XA^T \Delta y^{aff} + r_\mu - Xr_c), \tag{24}$$

$$\Delta z^{aff} = X^{-1} r_\mu - X^{-1} Z \Delta x^{aff}, \tag{25}$$

where $D = Z^{-1}X$. Calculating the centering parameters α_{aff}^{pri}, α_{aff}^{dual} and μ_{aff} and setting $\sigma = (\mu_{aff}/\mu)^3$, with $\mu = x^T z/n$, the linear system (22) is solved again with the right-hand side $r_h = 0$, $r_c = 0$, $r_\mu = \sigma\mu e - \Delta x^{aff}\Delta z^{aff} e$ for the solution $(\Delta x^{cc}, \Delta y^{cc}, \Delta z^{cc})$.
After the search direction and step to boundary is calculated

$$(\Delta x^k, \Delta y^k, \Delta z^k) = (\Delta x^{aff}, \Delta y^{aff}, \Delta z^{aff}) + (\Delta x^{cc}, \Delta y^{cc}, \Delta z^{cc}).$$

Then the proper values α_k^{pri} and α_k^{dual} are determined and the vectors $(x^{k+1}, y^{k+1}, z^{k+1})$ are established as

$$x^{k+1} = x^k + \alpha_k^{pri}\Delta x^k \tag{26}$$

$$(y^{k+1}, z^{k+1}) = (y^k, z^k) + \alpha_k^{dual}(\Delta y^k, \Delta z^k). \tag{27}$$

The crucial step for finding the vector Δx^{aff} in (24) is to solve the equation (23). From the computational point of view it is the most time consuming part of the algorithm. According to the already said, the system of the following form must be solved two times in every iteration with different right-hand sides

$$(ADA^T)\ \Delta y = b. \tag{28}$$

We have used the factorization technique based on the work of Birge and Qi for two-stage stochastic program ([6] and its parallelization suggested in [22]. Pflug and Halada [30] proposed a generalization of this type of factorization for multistage stochastic programs. The parallel computer program for 3-stage stochastic program and its application for parallel computer with distributed memory architecture is suggested in following paragraphs.

4 Parallel BQ factorization for three-stage stochastic programs

We describe the BQ factorization for solving the problem (28), where the matrix $A = A^{(3)}$ comes from the three-stage stochastic program. As an example, the problem BQS(3,1) depicted on the Fig.2, can be described by the equation

$$(A^{(3)}D^{(3)}(A^{(3)})^T)\ \Delta y = b^{(3)} \tag{29}$$

using the two-stage stochastic problems $BQS(2,k),\ k = 1, 2, \dots Sc_1$, defined as

$$(A_k^{(2)}D_k^{(2)}(A_k^{(2)})^T)\ p_k^{(2)} = b_k^{(2)}. \tag{30}$$

For clarity reasons in the next we will denote the number of scenarios of the first period Sc_1 by $N^{(3)}$.

4.1 Three-stage algorithm

It has been proven in [30] that the inversion of the matrix $A^{(3)}D^{(3)}(A^{(3)})^T$ can be computed by the Sherman-Morrison-Woodbury formula as follows:

$$(A^{(3)}D^{(3)}(A^{(3)})^T)^{-1} = (\mathcal{R}^{(3)})^{-1} - (\mathcal{R}^{(3)})^{-1}U^{(3)}\ (G^{(3)})^{-1}\ (V^{(3)})^T\ (\mathcal{R}^{(3)})^{-1} \quad (31)$$

where

$$\mathcal{R}^{(3)} = Diag(I_{m_0^{(3)}},\ R_1^{(2)},\ R_2^{(2)},\ ...,\ R_{N^{(3)}}^{(2)}),$$

$$U^{(3)} = \begin{pmatrix} A_0^{(3)} & I_{m_0^{(3)}} \\ T_1^{(3)} & \\ T_2^{(3)} & \\ \vdots & \\ T_{N^{(3)}}^{(3)} & \end{pmatrix},\ (V^{(3)})^T = \begin{pmatrix} (A_0^{(3)})^T & (T_1^{(3)})^T & (T_2^{(3)})^T & \dots & (T_{N^{(3)}}^{(3)})^T \\ -I_{m_0^{(3)}} & & & & \end{pmatrix},$$

$$G^{(3)} = \begin{pmatrix} \hat{G}^{(3)} & (A_0^{(3)})^T \\ -A_0^{(3)} & 0 \end{pmatrix},$$

and

$$\hat{G}^{(3)} = (D_0^{(3)})^{-1} + (A_0^{(3)})^T A_0^{(3)} + \sum_{k=1}^{N^{(3)}} (T_k^{(3)})^T (R_k^{(2)})^{-1} T_k^{(3)},$$
$$R_k^{(2)} = A_k^{(2)} D_k^{(2)} (A_k^{(2)})^T.$$

Thus, the solution $(A^{(3)}D^{(3)}(A^{(3)})^T)\ \Delta y = b^{(3)}$ can be expressed by the inversion on the basis of the validity (31) as $\Delta y^{(3)}) = p^{(3)} - s^{(3)}$ while

$$\mathcal{R}^{(3)} p^{(3)} = b^{(3)}, \quad (32)$$

$$G^{(3)}q^{(3)} = (V^{(3)})^T p^{(3)}, \tag{33}$$
$$\mathcal{R}^{(3)}s^{(3)} = U^{(3)}q^{(3)}. \tag{34}$$

The equations (32)-(34) represent the decomposition of the original problem into three sub-problems. An advantage of such a decomposition is that $\mathcal{R}^{(3)}$ is the block-diagonal matrix, where the diagonal matrix element $R_k^{(2)}$ represents the two-stage stochastic problem BQS(2,k), $k = 1, 2, ..., N^{(3)}$.
Hence, the parallel three-stage procedure can be summarized as follows :

Parallel three-stage procedure *solve3*$(A^{(3)}D^{(3)}(A^{(3)})^T; b^{(3)})$

```
Step 3.1 ( Solve $\mathcal{R}^{(3)}p^{(3)} = b^{(3)}$)
  $p_0^{(2)} = b_0^{(2)}$
  parallel do $k = 1, 2, \ldots N^{(3)}$
    $p_k^{(2)} = \mathit{solve2}(A_k^{(2)}D_k^{(2)}(A_k^{(2)})^t; b_k^{(2)})$
  end parallel do
Step 3.2 ( Solve $G^{(3)}q^{(3)} = (V^{(3)})^T p^{(3)}$)
  parallel do $k = 1, 2, \ldots N^{(3)}$
    $\bar{G}_k^{(3)}\widehat{T}_k^{(3)} = T_k^{(3)}$
  end parallel do
Step 3.3
  $\widehat{G}^{(3)} = (D_0^{(3)})^{-1} + (A_0^{(3)})^T A_0^{(3)} + \sum_{k=1}^{N^{(3)}}(T_k^{(3)})^T(\widehat{T}_k^{(3)} - T_k^{(3)})$
  Compute Cholesky decomposition of $\widehat{G}^{(3)}$
  Compute $\widehat{v}_1 = (A_0^{(3)})^T p_k^{(0)} + \sum_{k=1}^{N^{(3)}}(T_k^{(3)})^T p_k^{(2)}$
  $\widehat{v}_2 = -p_0^{(2)}$
  $\widehat{G}^{(3)}B_0^{(3)} = (A_0^{(3)})^T$
  $\widehat{G}^{(3)}\tilde{v}_1 = \widehat{v}_1$
  $\bar{G}^{(3)} = A_0^{(3)}B_0^{(3)}$
  Compute Cholesky decomposition of $\bar{G}^{(3)}$
  $\bar{G}^{(3)}q_2^{(3)} = A_0\tilde{v}_1 + \widehat{v}_2$
  $\widehat{G}^{(3)}q_1^{(3)} = \widehat{v}_1 - (A_0^{(3)})^T q_2^{(3)}$
Step 3.4 ( Solve $\mathcal{R}^{(3)}s^{(3)} = U^{(3)}q^{(3)}$)
  $s_0^{(2)} = b_0^{(2)}$
  parallel do $k = 1, 2, \ldots N^{(3)}$
    $s_k^{(2)}$ = solve2 $(A_k^{(2)}D_k^{(2)}(A_k^{(2)})^t; T_k^{(3)}q_1^{(3)})$
  end parallel do
Step 3.5
  parallel do $k = 0, 1, 2, \ldots N^{(3)}$
    $\Delta y_k^{(3)} = p_k^{(2)} - s_k^{(2)}$
  end parallel do
```

The output of this procedure is vector Δy which fulfills the relation

$$\Delta y = (\Delta y_0^{(3)}, \Delta y_1^{(3)}, ..., \Delta y_{N^{(3)}}^{(3)}), \tag{35}$$

where the last two vectors of (35) are the results of the Step 3.1 and Step 3.3, respectively. They are computed by the procedure *solve* $(A_k^{(2)} D_k^{(2)} (A_k^{(2)})^t; rhs)$ with different vectors rhs. The basic steps of this procedure are as follows:

4.2 Parallel two-stage procedure

The problems BQS(2,j), $j = 1, 2, 3$, depicted on the Fig.2, are the examples of the two-stage stochastic problems that could be described by the equation (30). The algorithm for solving the two-stage stochastic problem has been suggested in [6], and its parallelization in [22]. Therefore in the next we will describe only the computational steps of the two-stage parallel procedure that could be helpful to the reader. For clarity reasons we will denote the number of the scenarios of the second period of the three-stage program Sc_2 by $N^{(2)}$.

Parallel two-stage procedure *solve2*$(A^{(2)} D^{(2)} (A^{(2)})^T; b^{(2)})$

```
Step 2.1
   parallel do $j = 1, ...N^{(2)}$
      $R_j = A_j^{(1)} D_j^{(1)} (A_j^{(1)})^T$
      Calculate Cholesky decomposition of $R_j = (L_j)(L_j)^T$
      Solve system of linear equations $(L_j)(L_j)^T p_j^{(1)} = b_j^{(1)}$
      $p_0^{(1)} = b_0^{(1)}$
   end parallel do
Step 2.2
   parallel do $j = 1, ...N^{(2)}$
      $(L_j)(L_j)^T \widehat{T}_j^{(2)}(:, l) = T_j^{(2)}(:, l)$, for $l = 1, ...12$
      $\widehat{H}_j^{(2)} = (T_j^{(2)})^T \widehat{T}_j^{(2)}$
      $t_j = (T_j^{(2)})^T p_j^{(1)}$
   end parallel do
Step 2.3
   $\widehat{G}^{(2)} = (D_0^{(1)})^{-1} + (A_0^{(2)})^T A_0^{(2)} + \sum_{j=1}^{N^{(2)}} \widehat{H}_j^{(2)}$
   $\widehat{v}_1 = (A_0^{(2)})^T p_0^{(1)} + \sum_{j=1}^{N^{(2)}} t_j$
   $\widehat{v}_2 = -p_0^{(1)}$
   Calculate Cholesky decomposition of $\widehat{G}^{(2)}$
   Solve systems of linear equations $(\widehat{G}^{(2)}) B(:, l) = (A_0^{(2)})^T$, for $l = 1, 2, ...5$.
   Solve the system of linear equations $(\widehat{G}^{(2)}) \tilde{v} = \widehat{v}_1$
   $\bar{G}^{(2)} = A_0^{(2)} B,$
   $\bar{R}^{(2)} = A_0^{(2)} \tilde{v} + \widehat{v}_2$
   Calculate Cholesky decomposition of $\bar{G}^{(2)}$
   Solve $\bar{G}^{(2)} q_2 = \bar{R}^{(2)}$
   $\widehat{G}^{(2)} q_1 = \widehat{v}_1 - (A_0^{(2)})^T q_2$
```

```
Step 2.4
  $s_0^{(1)} = (A_0^{(2)})^T q_1 + q_2$
  parallel do $j = 1, ...N^{(2)}$
    $u_j = T_j^{(2)} q_1$
    $R_j s_j^{(1)} = u_j$
  end parallel do
Step 2.5
  parallel do $j = 0, 1, ...N^{(2)}$
    $\Delta y_j^{(2)} = p_j^{(1)} - s_j^{(1)}$
  end parallel do
```

5 Experimental results

5.1 Parallel implementation and performance

Multiperiod portfolio management problems inevitably lead to the solving of large linear systems (22). The size of the matrix depends on the number of $\mathcal{T}$ time periods considered. For four currencies and the scenario tree of the size $Sc_1 \times Sc_2 \times *$, the size of the matrix $A^{(3)}$ is $[(Sc_1 * (Sc_2 + 1) + 1) * 5] \times [(Sc_1 * (Sc_2 + 1) + 1) * 12]$ in the three-stage case. As an example, for the scenario tree $300 \times 300 \times *$, with 4 currencies considered, the size of the matrix $A^{(3)}$ is $451505 \times 1\ 083\ 612$ and the linear system (23) has 451 505 unknowns. The matrix $A^{(3)}$ is a sparse matrix and has only $Sc_1 * (Sc_2 + 1) * 20 + 20 + 4 * Sc_1 * (Sc_2 + 1)$ nonzero from total $(Sc_1 * (Sc_2 + 1) * 5 + 5) \times (Sc_1 * (Sc_2 + 1) * 12 + 12)$ elements. In the given example it is 2 167 220 nonzero from the total 203 856 765 025 elements.

The application of the IPM method for solving the optimization problem (14) enables to decompose the large three-stage stochastic problem represented by matrix $A^{(3)}$ to Sc_1 two-stage problems. Each of the two-stage problems determined by a matrix $A^{(2)}$, is composed again of Sc_2 one-stage problems represented in our experiments by matrices $A^{(1)}$ of the size 5×12.

The use of the BQ decomposition for solving of the system (29) by means of the procedure *solve3*() enables the decomposition of the large three-stage problem represented by matrix $A^{(3)}$ into Sc_1 smaller problems, whereby each of them corresponds to one two-stage problem. As it is apparent from the designed parallel algorithm in the procedure *solve3*(), the most of the computational steps (except the Step 3.3) can be executed in parallel. The same is true for executing of the two-stage procedure *solve2*(), that is called two times from *solve3(),* more precisely in the Step 3.1 and Step 3.4.

This fact points out to the possibility of parallel calculations on two levels: the higher level enabling parallel execution of all steps assigned for parallel execution in the procedure *solve3*(). On the lower level, the same holds true for each of the Sc_2 two-stage problems solved by *solve2*(). Also there, the most computational steps are marked with the ”parallel do” clause.

The interprocessor communication is needed only for establishing of the matrices $\widehat{G}^{(3)}$ and $\widehat{G}^{(2)}$, and vectors $\widehat{v}_1$ in the Step 3.3 and Step 2.3, respectively.

Suppose, there are $NPROC$ processors available and two integers P, Q, so that $P \times Q = NPROC$. The $NPROC$ processors can be decomposed into P groups with the rank of Q processors each. Then in the procedure *solve3*(), the parallel loops are executed using the P processors, whereby Q processors are used for parallel computations inside the procedure call of *solve2*() (Step 3.1 and Step 3.4).

For example, let's solve a three-stage stochastic problem with $Sc_1 = 300$ and $Sc_2 = 200$ by $NPROC = 8$ processors. Let $P = 4$ and $Q = 2$. Then 4 groups with 2 processors in each, are formed. To each group of processors is assigned $300/P = 75$ two-stage problems. Each of the 300 two-stage problems is using $Q = 2$ processors for parallel execution of 200 one-stage problems according to the procedure *solve2*(). In this example, the ”parallel do” instruction is executed on 4 processors in *solve3*() and on 2 processors in *solve2*().

The algorithms used for solving one-stage problems are as follows: Cholesky decomposition, solving a system of linear equations, and matrix-vector or matrix-matrix multiplications. For these operations, performed on the matrices of the size 5×12, the sequential version of the linear algebra package LAPACK [1] has been used. The LAPACK routines are called in the frame of the Fortran code using the Message Passing Interface (MPI) [27] for interprocessor communication.

Despite the fact, that the IPM decomposition enabled to decompose the large linear system into smaller ones, while evaluating the right-hand size of equation (23) one cannot avoid the calculation of the matrix-vector product with the whole matrix $A^{(3)}$. Due to these reasons we have stored the matrix $A^{(3)}$ also in a compressed row storage form [2] and used it for a sequential sparse matrix-vector multiplication. Other experimental details and results of the decomposition and the parallel implementation of the stochastic three-stage algorithm may be found also in our previous papers [4], [17], [18].

In this paper we present the experiments executed on the IBM Linux cluster, University of Technology, Vienna, consisting of 144 Pentium 4 (3.6GHz) processors (2 processors per node) communicating over Fast Ethernet node interconnect.

Table 1 illustrates performance results of the three-stage stochastic problems of the size $300 \times 300 \times *$. P denotes here the number of two-stage problems solved in parallel, while Q processors were employed in the parallel computation inside every two-stage problem. The time of execution for $P = Q = 1$, was 203.2 seconds. All experiments showed, that the best performance results were achieved, when the number P of processors designated for parallel processing of two-stage problems in the procedure *solve3()* was as high as possible.

NP	PxQ								
	1x2	2x1	1x4	2x2	4x1	1x8	2x4	4x2	8x1
2	171.7	97.67							
4			119.33	73.33	60.92				
8						95.58	71.42	48.33	41.74
	1x16	2x8	4x4	8x2	16x1				
16	71.92	52.4	47.17	36.4	23.4				

Table 1
Timing results in seconds for one model day calculations with the three-stage problem of the size $300 \times 300 \times *$. NP is the number of processors utilized, and $P \times Q$ specifies the configuration of the virtual processor array.

5.2 Financial performance

The tests were performed on historical data from January 1997 to January 2003. The data included prices of bond indices of 10 years government bonds in USD, EUR, CHF and GBP and corresponding exchange rates. The domestic currency was USD and therefore the values of portfolio were recalculated to USD. At the beginning there was a cash in USD available and the portfolio was bought according to the result of the multistage stochastic program. The portfolio was then daily rebalanced according to the results of the stochastic program. The scenario tree contained 300 scenarios per stage. We used the same time horizon of 6 weeks for two-stage and three-stage models. This implies the period length 2 weeks in the three-stage model and 3 weeks in the two-stage model. The important parameters of scenarios are future expected prices of indices $P_j^{(\tau)}$, $\tau = 2, 3, \ldots, \mathcal{T}$. In our experiments we supposed that the returns of expected prices were known future returns of indices $j = 1, 2, 3, 4$ with normal perturbation:

$$\begin{aligned} P_j^{(2)}/I_j^{(1)} &= I_j^{(2)}/I_j^{(1)} + \sigma_e Z_{1,j}\,, \\ P_j^{(3)}/P_j^{(2)} &= I_j^{(3)}/I_j^{(2)} + \sigma_e Z_{2,j}\,, \\ P_j^{(4)}/P_j^{(3)} &= I_j^{(4)}/I_j^{(3)} + \sigma_e Z_{3,j} \end{aligned} \tag{36}$$

where $I_j^{(\tau)}, \tau = 1, 2, 3, 4$ denote the known root and future prices of indices. The known future returns of indices are perturbated (to simulate an incomplete market information) by mutually independent N(0,1) distributed random variables $Z_{\tau,j}$, σ_e stands for the standard deviation of the error. Such a model of incomplete information is inspired by modeling of asset returns with normal random variables often used in finance. $\sigma_e = 0$ represents the perfect information about the future behavior of the markets. The higher the σ_e, the worse the information about the future development of the markets. We have also tested a model with ”no information”. Suppose that we are calculating the decision at time t and denote the period length by $\triangle t$. Denote by $\rho_j^{(-3)}$, $\rho_j^{(-2)}$ and $\rho_j^{(-1)}$, $j = 1, 2, 3, 4$, the realized returns of the indices in the last past successive periods, i.e. $[t - 3\triangle t, t - 2\triangle, t]$, $[t - 2\triangle t, t - \triangle t]$, $[t - \triangle t, t]$, respectively. In the ”no information” model we suppose that

$$\begin{aligned} P_j^{(2)}/I_j^{(1)} &= 1 + \rho_j^{(-3)} \, , \\ P_j^{(3)}/P_j^{(2)} &= 1 + \rho_j^{(-2)} \, , \\ P_j^{(4)}/P_j^{(3)} &= 1 + \rho_j^{(-1)} \, . \end{aligned} \tag{37}$$

Using (36) or (37) one can calculate the expected future prices of indices and generate the scenario tree according to Section 1.5.

Tables Tab. 2 and Tab. 3 contain values of the portfolio at annual dates achieved with two-stage and three-stage models for different quality of the information. The portfolio value development for 2-stage and 3-stage models with different levels of information could be seen in Fig. 3 and Fig. 4. One can conclude that two-stage model is better when the information about the future development of the markets is worse ($\sigma_e = 20\%$, No info). With better information ($\sigma_e = 0$, $\sigma_e = 5\%$, $\sigma_e = 10\%$) the three stage model outperforms the two-stage model. One would expect better results with more stages because of more possibilities to rebalance. However, mathematical models in finance must be properly calibrated to deliver acceptable results. A good example is the well-known Martkowitz portfolio selection model ([26]), which needs good estimations of the expected asset returns and their covariances. In our model higher number of stages cannot overcome bad estimates of the future expectations ($\sigma_e = 20\%$, no info). In a properly calibrated model a higher number of scenarios makes sense for two reasons. First, one cannot estimate precisely the future development of the prices. Taking into account more possibilities increases a chance that one of the scenarios will fit closely the future

Date	$\sigma_e = 0$	$\sigma_e = 5\%$	$\sigma_e = 10\%$	$\sigma_e = 20\%$	No info
Jan-97	100.00	100.00	100.00	100.00	100.00
Jan-98	120.21	120.44	117.64	115.88	115.37
Jan-99	168.59	156.53	157.15	155.63	129.37
Jan-00	156.72	147.73	148.88	145.26	101.38
Jan-01	192.55	192.73	186.33	165.92	113.15
Jan-02	214.38	208.07	193.46	166.67	113.91
Jan-03	307.86	288.15	257.30	212.22	155.98

Table 2
Values of the portfolio: two-stage model

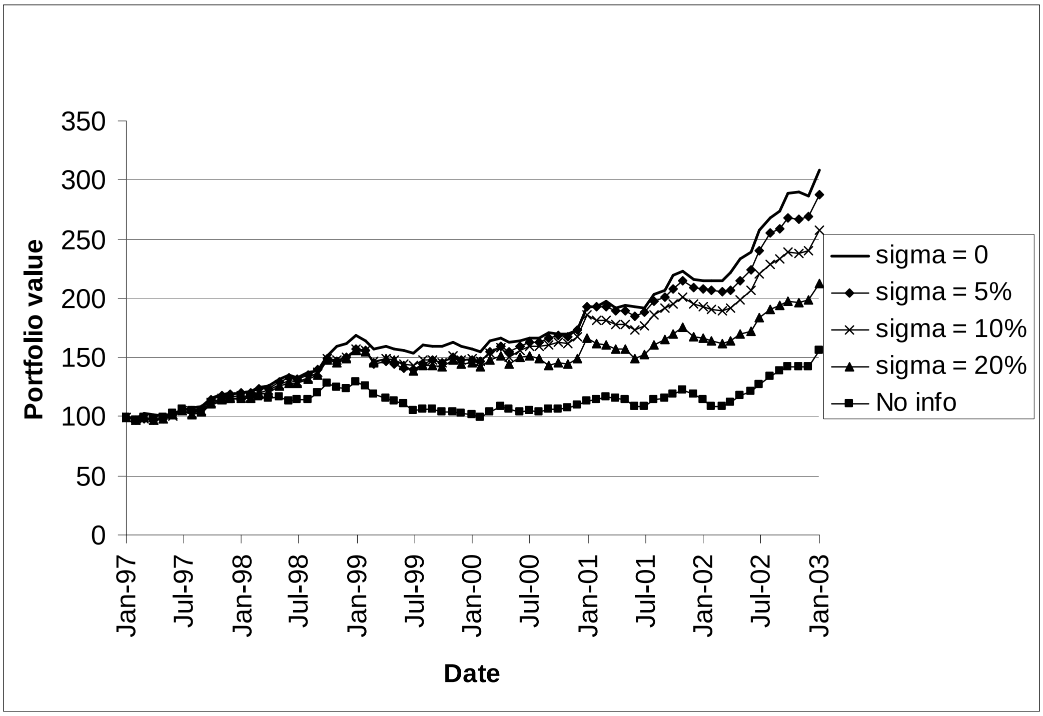

Fig. 3. Portfolio value development using the 2-stage model.

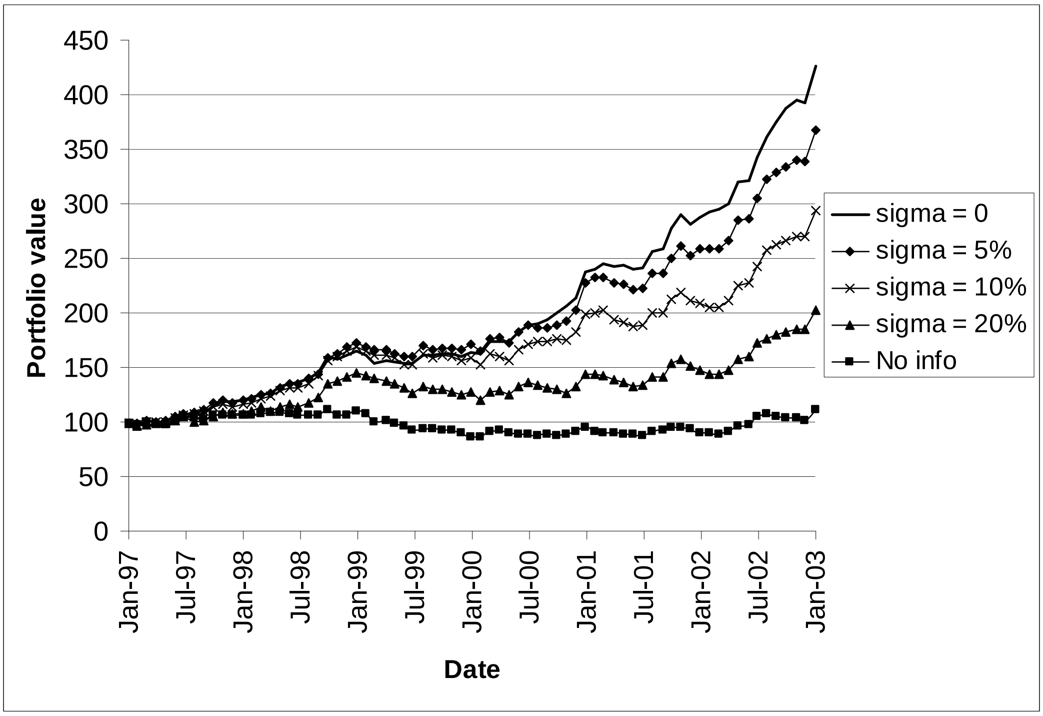

Fig. 4. Portfolio value development using the 3-stage model.

Date	$\sigma_e = 0$	$\sigma_e = 5\%$	$\sigma_e = 10\%$	$\sigma_e = 20\%$	No info
Jan-97	100.00	100.00	100.00	100.00	100.00
Jan-98	119.81	120.06	116.49	108.00	106.40
Jan-99	164.85	172.79	168.57	145.29	110.15
Jan-00	163.68	171.15	158.35	127.77	86.51
Jan-01	237.82	227.24	198.61	143.47	94.76
Jan-02	287.31	258.49	208.89	147.09	90.17
Jan-03	426.09	367.13	293.39	202.02	111.54

Table 3
Values of the portfolio: three-stage model

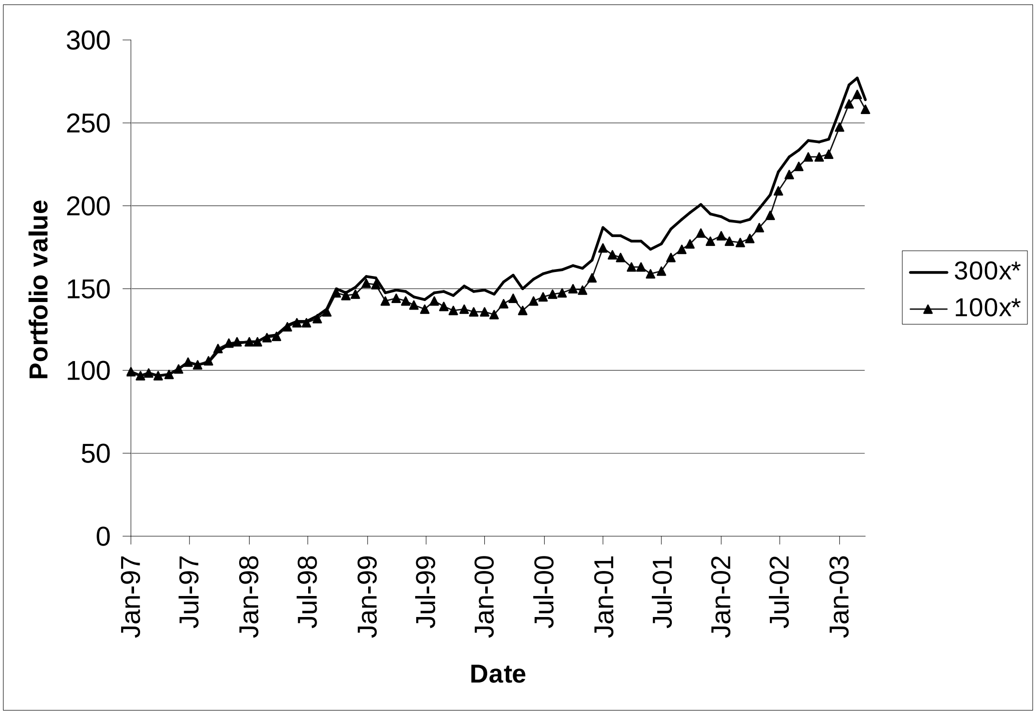

Fig. 5. Sensitivity of the portfolio value development to the number of scenarios per stage for the 2-stage model with $\sigma = 10\%$.

real prices. Second, the increase of the number of scenarios per stage makes the discrete model closer to the log-normal process (10). The sensitivity of the portfolio value development to the number of scenarios per stage for the 3-stage and 2-stage models with $\sigma = 10\%$ could be seen in Fig. 5 and Fig. 6. The results are calculated for 100 and 300 scenarios per stage. One can see that in both models the results are better for higher number of scenarios. Therefore, increasing the computational performance gives the possibility of higher number of scenarios and this could (in a properly calibrated model) imply better financial performance.

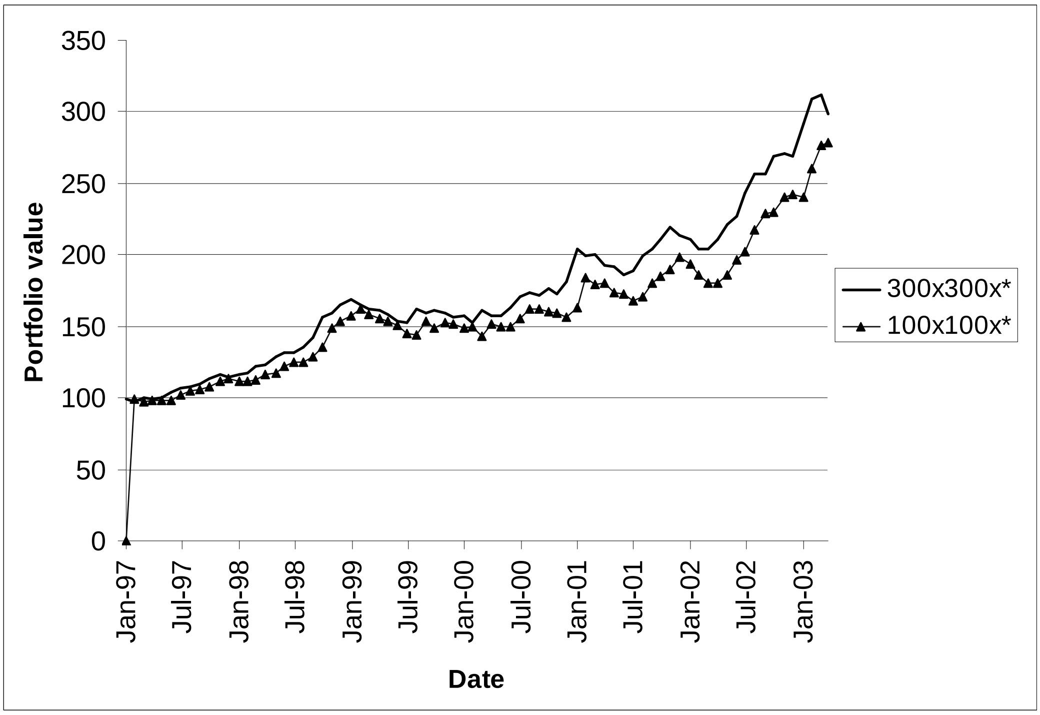

Fig. 6. Sensitivity of the portfolio value development to the number of scenarios per stage for the 3-stage model with $\sigma = 10\%$.

6 Conclusions

We have presented a multistage stochastic model for allocation of financial resources to bond indices in different currencies. The model was tested on historical data containing the values of indices of government bonds and exchange rates. We have compared the two-stage and the three-stage stochastic models from the point of view of financial performance. We have showed that it make sense to increase the number of stages if we have ”a good information” about the future development of financial markets. Increasing the number of stages need not improve the quality of the results in the case of a low quality of information. Furthermore, in a properly calibrated model a higher number of scenarios could improve the financial performance.

For solving two-stage and three-stage stochastic programs the interior point method (IPM) was used. An application of the BQ method to the IPM allowed decomposition of the large linear system to smaller blocks. The three-stage parallel programs were parallelized in two levels. On the higher level several two-stage problems were processed in parallel, while on the lower level several one-stage problems were processed in parallel. By creating a virtual processor array we have showed that the best performance results were achieved, when the highest possible number of processors was employed in the parallel processing of two-stage problems. The parallel code was written in the Fortran programming language using the MPI.

Acknowledgments

This work was supported by the Special Research Program SFB F011 ”AURORA” of the Austrian Science Fund FWF and VEGA Grant Agency (1/9154/02, 1/1004/04, 1/3767/06), Slovakia.

References

[1] E. Anderson, Z. Bai, C. H. Bischof, S. Blackford, J. W. Demmel, J. J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. C. Sorensen, Lapack *Users' Guide*, 3rd edn. SIAM Press, Philadelphia PA, 1999.

[2] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, editors. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000.

[3] A. Beltratti, A. Consiglio and S. A. Zenios, *Scenario Modeling for the Management of International Bond Portfolios*, Annals of Oper. Res. 85, (1999), pp. 227-247.

[4] S. Benkner and L. Halada and M. Lucka, *Parallelization Strategies of Three-Stage Stochastic Program Based on the BQ Method*, Parallel Numerics'02, Theory and Applications. Edited by R. Trobec, P. Zinterhof, M. Vajtersic and A. Uhl, (2002), pp. 77-86.

[5] J. Blomvall and P. O. Lindberg, *A Riccati-based primal interior point solver for multistage stochastic programming*, European Journal of Operational Research 143 (2002), pp. 452-461.

[6] J. R. Birge and L. Qi, *Computing block-angular Karmarkar projections with applications to stochastic programming*, Management Sci. 34, 1988, pp. 1472-1479.

[7] A. J. Bradley and D. B. Crane, *A Dynamic Model for Bond Portfolio Management*, Management Science, 19, (1972), pp.139-151.

[8] D. R. Carino, T. Kent, D. H. Myers, C. Stacy, M. Sylvanus, A. L. Turner, K. Watanabe and W. T. Ziemba, *The Russell-Yasuda Kasai Model: an Asset/Liability Model for a Japanese Insurance Company Using Multistage Stochastic Programming*, Interfaces 24(1), (1994), pp. 29-49.

[9] Y. Censor, S. A. Zenios, *Parallel Optimization* , Oxford University Press, New York, (1993).

[10] G. B. Dantzig and G. Infanger, *Multistage Stochastic Linear Programs for Portfolio Optimization*, Annals of Operations Research 45, (1993), pp. 59-76.

[11] J. DUPAČOVÁ, *Applications of Stochastic Programming in Finance*, In: Atti del XVI Convegno A.M.A.S.E.S, Treviso, (1992), pp. 13-30.

[12] K. FRAUENDORFER AND M. SCHÜRLE, *Barycentric Approximation of Stochastic Interest Rate Process*, In Ziemba, W.T. (ed.) et al., Worldwide asset and liability modeling. Research seminar, Cambridge Univ., Cambridge University Press, (1995), pp. 231-262.

[13] B. GOLUB, M. HOLMER, R. MC KENDALL AND S. A. ZENIOS, *Stochastic Programming Models for Money Management*, European Journal of Operations Research, 85, (1995), pp. 282-296.

[14] J. GONDZIO AND A. GROTHEY, *Parallel interior point solver for structured quadratic programs: Application to financial planning problems*, Annals of Operations Research, 152 (2007), pp. 319-339.

[15] J. GONDZIO AND A. GROTHEY, *Solving nonlinear portfolio optimization problems with the primal-dual interior point method*, European Journal of Operational Research, 181 (2007), pp. 1019-1029.

[16] J. GONDZIO AND R. SARKISSIAN, *Parallel interior point solver for structured linear programs*, Mathematical Programming, 96 (2003), pp. 561-584.

[17] L. HALADA, S. BENKNER, M. LUCKA, *Parallel Birge and Qi Factorization of Three-stage Stochastic Linear Programs*, Parallel and Distributed Computing Practices **5**, No.3, Nova Science Publishers, (2002), pp. 301-311.

[18] L. HALADA, M. LUCKA AND I. MELICHERCIK, *Optimal Multistage Portfolio Management Using a Parallel Interior Point Method*, Proc. of ALGORITMY’2005, High Tatras, Slovakia, (2005), pp. 359-368.

[19] R. S. HILLER AND J. ECKSTEIN, *Stochastic Dedication: Designing Fixed Income Portfolios Using Massively Parallel Benders Decomposition*, Management Science 39, (1994), pp. 1422-1438.

[20] M. HOLMER, R. MC KENDALL, CH. VASSIADOU-ZENIOU AND S. A. ZENIOS, *Dynamic Models for Fixed-Income Portfolio Management under Uncertainty*, Journal of Economic Dynamics and Control, 22, (1998), pp. 1517-1541.

[21] J. INGERSOLL, *Theory of Financial Decision Making*, Studies in Financial Economics. Rowman & Littlefield, Totowa, New Jersey, (1987).

[22] E. R. JESSUP, D. YANG, AND S. A. ZENIOS, *Parallel factorization of structured matrices arising in stochastic programming*, SIAM J.Optimization, Vol. 4, No.4, (1994), pp. 833-846.

[23] H. KONNO AND H. YAMAZAKI, *Mean-absolute Deviation Portfolio Optimization Model and its Applications to Tokyo Stock Market* , Management Science, 37(5), (1991), pp. 519-531.

[24] M. I. KUSY AND W. T. ZIEMBA, *A bank Asset and Liability Management Model*, Operations Research, 34, (1986), pp. 256-276.

[25] H. W. MORITSCH AND G. CH. PFLUG, *Using a Distributed Active Tree in Java for the Parallel and Distributed Implementation of a Nested Optimization Algorithm*, In: IEEE Proceedings of the 5th Workshop on High Performance Scientific and Engineering Computing with Applications (HPSECA-03), October 2003, Kaohsiung, Taiwan, ROC.

[26] H. M. MARKOWITZ, PORTFOLIO SELECTION, *Journal of Finance*, **7**, (1952), pp. 77-91.

[27] Message Passing Interface Forum. MPI: A Message-Passing Interface Standard, Vers. 1.1, June 1995. MPI-2: Extensions to the Message-Passing Interface, 1997.

[28] J. M. MULVEY, *Incorporating Transaction Costs in Models for Asset Allocation*, In S. A. Zenios, (Ed.), Financial Optimization, Cambridge University Press, (1993), pp. 243-259.

[29] SOREN S. NIELSEN, STAVROS A. ZENIOS, *Scalable parallel Benders decomposition for stochastic linear programming*, Parallel Computing 23, (1997), pp. 1069-1088.

[30] G. CH. PFLUG, L. HALADA, *A Note on the Recursive and Parallel Structures of the Birge and Qi Factorization for Tree Structured Linear Programs*, Computational Optimization and Applications**24**, No. 2-3, (2003), pp. 251-265.

[31] G. CH. PFLUG, A. SWIETANOVSKY, *Selected parallel optimization methods for financial management under uncertainty*, Parallel Computing 26, (1), (2000), pp. 3-25.

[32] G. H. PFLUG, L. HALADA, M. LUCKA, *Parallel Implementation of the Birge and Qi Method for three-stage stochastic programs using IPM*, Proc. of the conference Algoritmy'2002, High Tatras, Slovakia, (2002), pp. 102-112.

[33] W. F. SHARPE, *Mean-absolute-deviation Characteristic Lines for Securities and Portfolios*, Management Science, 18(2), (1971).

[34] H. VLADIMIROU, S. A. ZENIOS, *Scalable parallel computations for large-scale stochastic programming*, Annals of Operations Research 90, (1999), pp. 87-129.

[35] S. A. ZENIOS, *High-performance computing in finance: the last 10 years and the next*, Parallel Computing 25, (1999), pp. 2149-2175.

[36] S. A. ZENIOS AND P. KANG, *Mean-absolute Deviation Portfolio Optimization for Mortgage backed Securities*, Annals of Operations Research, 45, (1993), pp. 433-450.

[37] S. A. ZENIOS, *A Model For Portfolio Management with Mortgage Backed Securities*, Annals of Operations Research 42, (1994), pp. 223-233.

[38] S. A. ZENIOS, *Asset/Liability Management under Uncertainty for Fixed-income Securities*, Annals of Operations Research, 95, (1995), pp. 77-98.

[39] STEPHEN J. WRIGHT, *Primal-Dual Interior Point Methods*, ISBN 0-89871-382-X, SIAM, 1997.