Proceedings of ALGORITMY 2005
pp. 1–10

OPTIMAL MULTISTAGE PORTFOLIO MANAGEMENT USING A PARALLEL INTERIOR POINT METHOD[*]

L.HALADA[†], M.LUCKA[‡], AND I.MELICHERCIK[§]

Abstract. We present a multi-stage model for allocation of financial resources to different currencies. The model is tested using a three-stage scenario tree with a mean-reversion property. For solving three-stage stochastic programs the interior point method (IPM) in the frame of the primal-dual path-following formulation is used. Because the matrix of the corresponding linear system is large, sparse and regular, it can be easily stored in a compressed format. The compressed storage is further used in calculating the right-hand side of the linear system and the objective function.
An application of the BQ method to the IPM allows decomposition of the large linear system to smaller blocks allowing thus solving it in parallel. The parallel code is designed for clusters of SMP's and written in the Fortran/MPI language. The linear algebra operations on the small block matrices are executed by the LAPACK library calls.
The achieved results have proved that by increasing the number of stages the quality of the optimal decision could be improved.

Key words. portfolio management, interior point method, parallel computing, large-scale optimization

AMS subject classifications. 82B31, 60H35, 68W10, 90C5

1. Introduction.

Recently an interest in the development of multistage models of portfolio management could be observed. At this place we refer to successful and valuable contributions of [1], [2] and [3]. In [4] a multi-period dynamic model for fixed-income portfolio management under uncertainty, using multi-stage stochastic programming was developed. The scenarios of the term structure of interest rates were generated using Monte Carlo simulations. Their results confirmed that multi-period models outperform classical single-period models.

We present an example of a portfolio management problem. It is a model for allocation of financial resources to bond indices in different currencies. The stochastic properties are represented in the form of a scenario tree. The scenarios contain future possible developments of interest rates and exchange rates. When one deals with several currencies, the realistic scenario trees are "bushy" and the number of scenarios grows exponentially with the number of stages. Thus, the computation of such problems could be extremely large and computationally intractable. Approaches for solving these problems usually either take advantage the problems' matrix structure or decompose the problem into smaller subproblems. In the literature we can also see a considerable research effort to develop efficient parallel methods for solving this problem on parallel computer architectures. In our paper we demonstrate a parallel interior point algorithm (IPM) for solving three-stage stochastic linear problem which comes from a three-period models of portfolio management.

The paper is organized as follows:

[*] This work was supported by the Special Research Program SFB F011 "AURORA" of the Austrian Science Fund FWF and VEGA Agency (1/9154/02, 1/1004/04), Slovakia.

[†] Institute for Informatics, SAS, Dubravska 9, 842 37 Bratislava, (upsyhala@savba.sk).

[‡] Faculty of Education, University of Trnava, Priemyselna 8, 918 43 Trnava, (mlucka@truni.sk).

[§] Department of Economic and Financial Modeling, Faculty of Mathematics, Physics and Informatics, Mlynska dolina, 840 00 Bratislava, (igor.melichercik@fmph.uniba.sk).

In Section 2 we formulate the problem and describe the scenario tree generation. Results of the three-stage portfolio management are presented in Section 3. Section 4 brings the application of the Interior Point Method to solving the three-stage stochastic programs. The Section 5 describes the problems with implementation of the three-stage stochastic problem with large matrices. The last section outlines the future work and briefly summarizes the gained results.

2. Problem formulation. The stochastic properties are represented in the form of a scenario tree. Denote by $\mathcal{F}_\tau,\ 1 \leq \tau \leq \mathcal{T}$ the set of nodes at time τ. For any $\omega \in \mathcal{F}_\tau,\ 1 < \tau \leq \mathcal{T}$, there is a unique element $a(\omega) = \omega' \in \mathcal{F}_{\tau-1}$, which is the unique predecessor of ω.

Financial instruments. Denote the decision variables 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 held in period τ,
and constants as
$c^{(0)}$: Initial cash available,
$h_j^{(0)}$: Composition of the initial portfolio,
$v_j^{(\tau)}(\omega^\tau),\ \omega^\tau \in \mathcal{F}_\tau$: The value of the j-th index in the base currency,
$\xi_j^{(\tau)}(\omega^\tau),\ \chi_j^{(\tau)}(\omega^\tau),\ \omega^\tau \in \mathcal{F}_\tau$: Bid and ask prices of the j-th index in the base currency computed by

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

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

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 we add constraints allowing to sell only β part of any asset or β part of the whole portfolio.

Equations. *Inventory balance and cash-flow accounting for the Period* 1

$$h_j^{(0)} + b_j^{(1)} - s_j^{(1)} = h_j^{(1)} \quad \forall j.$$
$$c^{(0)} + \sum_j \xi_j^{(1)} s_j^{(1)} = \sum_j \chi_j^{(1)} b_j^{(1)}.$$

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

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

Risk reduction
The terminal wealth calculation is given by:

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

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

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

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

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

where

$$E(WT) = \sum_{\omega^{\mathcal{T}} \in \mathcal{F}_{\mathcal{T}}} \pi(\omega^{\mathcal{T}}) WT(\omega^{\mathcal{T}}) \tag{2}$$

($\pi(\omega^{\mathcal{T}})$ is the probability of the scenario $\omega^{\mathcal{T}}$). Using (1) the expected terminal wealth (2) could be written in the form

$$\begin{aligned} E(WT) &= \sum_{\omega^{\mathcal{T}} \in \mathcal{F}_{\mathcal{T}}} \pi(\omega^{\mathcal{T}}) \sum_j \xi_j^{(\mathcal{T})}(\omega^{\mathcal{T}}) h_j^{(\mathcal{T}-1)}(a(\omega^{\mathcal{T}})) \\ &= \sum_{\omega^{\mathcal{T}} \in \mathcal{F}_{\mathcal{T}}} \sum_j \pi(\omega^{\mathcal{T}}) \xi_j^{(\mathcal{T})}(\omega^{\mathcal{T}}) h_j^{(\mathcal{T}-1)}(a(\omega^{\mathcal{T}})) \\ &= \sum_{\omega^{(\mathcal{T}-1)} \in \mathcal{F}_{(\mathcal{T}-1)}} \sum_j h_j^{(\mathcal{T}-1)}(\omega^{\mathcal{T}-1}) \tilde{c}_j(\omega^{\mathcal{T}-1})\,, \end{aligned} \tag{3}$$

where

$$\tilde{c}_j(\omega^{\mathcal{T}-1}) = \sum_{\omega^{\mathcal{T}} \in \mathcal{F}_{\mathcal{T}} : a(\omega^{\mathcal{T}}) = \omega^{\mathcal{T}-1}} \pi(\omega^{\mathcal{T}}) \xi_j^{(\mathcal{T})}(\omega^{\mathcal{T}})\,. \tag{4}$$

Scenario tree. The set of scenarios could be generated in many ways. For practical realization we use a discretization of the process with following properties:

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

$$v_j^{(\tau+1)} = v_j^{(\tau)} \exp(\mu \,\triangle t + \sigma_j \sqrt{\triangle t}\, Z_j)\,, \tag{5}$$

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

2. Suppose we have expected prices of the indices $P_j^{(\tau)}$, $\tau = 2, 3, \ldots, \mathcal{T}$ (in the base currency). These prices could be calculated from predictions of renowned financial institutions. For our purposes (bond indices in different currencies) we need predictions of interest and exchange rates. We suppose that the price process (5) between the periods τ and $\tau + 1$ is mean reverting to the prescribed price $P_j^{(\tau+1)}$ in period $\tau + 1$:

$$E(v_j^{(\tau+1)} | v_j^{(\tau)}) = P_j^{(\tau+1)}\,. \tag{6}$$

The condition (6) is fulfilled, if we take (5) with

$$\mu = \log\left(\frac{P_j^{(\tau+1)}}{v_j^{(\tau)}}\right) \frac{1}{\triangle t} - \frac{1}{2}\sigma_j^2\,.$$

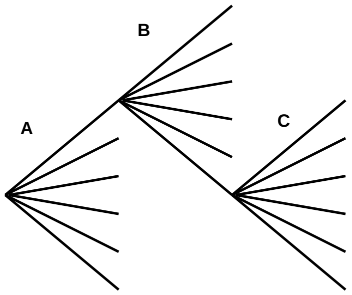

FIG. 1. *Scenario tree of type $A \times B \times C$.*

In our calculations we used a three-stage scenario tree. The scenario tree of type $A\text{x}B\text{x}C$ means that there are A possibilities of the price vector $v_i^{(2)}$ at the end of the first period, B possibilities (for each position $\omega^2 \in \mathcal{F}_2$) of the price vector $v_i^{(3)}$ at the end of the second period and C possibilities (for each position $\omega^3 \in \mathcal{F}_3$) of the price vector $v_i^{(4)}$ at the end of the third period. The procedure of generating of the tree of type $A\text{x}B\text{x}C$ is following:

1. For each node ω^τ, $\tau = 1, 2, 3$ we consider n independent $N(0, 1)$ random variables (n = number of indices) $\tilde{Z}_1, \tilde{Z}_2, \ldots, \tilde{Z}_n$. Each random variable is discretized by k values and probabilities. The discretization is same for all nodes in the same period. Thus, if we discretize the random variables in periods $\tau = 1, 2, 3$ by k_1, k_2, k_3 values respectively, we obtain a scenario tree of type $k_1^n\text{x}k_2^n\text{x}k_3^n$.

2. Using Cholesky factorization we obtain n random variables $Z_1, Z_2, \ldots, Z_n$ with prescribed correlation matrix cor (Z_i, Z_j).

3. Using (5) we obtain a scenario tree of prices.

3. Results of three-stage portfolio management. For testing we chose to calculate the results for a portfolio composed of 4 bond indices in 4 currencies: CHF, USD, GBP and EUR. The base currency was CHF. The time period was 9 months, the scenario tree had 3 stages. Each stage represented a 3-months period.

The expected prices $P_i^{(\tau)}$, $i = 2, 3, 4$ were calculated from expectations of interest rates and exchange rates development published by Meryll Lynch at the beginning of the year 2003. The covariance matrix of the returns was estimated from historical data of the indices and exchange rates. We have performed calculations for scenario trees of types: $16 \times 16 \times 16$ $(2^4 \times 2^4 \times 2^4)$ corresponding to 3276 variables, $81 \times 16 \times 16$ $(3^4 \times 2^4 \times 2^4)$ corresponding to 16536 variables, and $81 \times 81 \times 16$ $(3^4 \times 3^4 \times 2^4)$ corresponding to 79716 variables. One can ask interesting questions concerning the proposed model:

1. Does it make sense to increase the number of stages?

We have compared the results of the model with 16x16x16 (three-stage) scenario tree to the results of the model with single-stage scenario tree with 4096 (=16x16x16) end nodes (identical to the end nodes of the three-stage 16x16x16 tree). The initial (period 1) decisions were different and the objective function was significantly higher for the model with the three-stage tree. This observation is in accordance with financial intuition: if the expectations of the returns of indices in different periods are different, then (unless the transaction costs are too high) it makes sense to invest

	1/99	4/99	7/99	10/99	1/00	4/00	7/00	10/00	1/01	4/01
CHF	0.0	0.0	5.0	0.0	0.0	24.7	0.0	24.7	0.0	5.0
EUR	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0
USD	29.6	29.6	5.0	5.0	29.6	5.0	29.6	0.0	29.6	19.8
GBP	0.0	0.0	19.8	24.7	0.0	0.0	0.0	0.0	0.0	5.0

Table 1

Sensitivity of the optimal point to different tree generations

money in different periods to different indices. The single-stage model represents a buy-and-hold decision without possibility of future re-balancing and therefore the results are worse comparing to the three-stage model.

2. Is there any difference between the multi-stage decision and the successive single-stage decisions?

The results show that if the transaction costs are positive, the initial (period 1) decisions could be different. This is again in accordance with intuition: The single stage (first period) decision does not see the next (second and third period) expectations of the returns of the indices. The transaction costs could cause that it makes sense to choose an index that is not the best one in the first period, but the expectations in the next periods are better. The switch of the investment to this index in the second or third periods could be costly.

3. The optimal decision is (in accordance with financial intuition) sensitive with respect to different generations of the scenario tree. In Tab. 1 we summarize financial decisions for 10 different tree generations. The trees (of the type 16 × 16 × 16) are generated in accordance with 10 successive actual 9-months market developments with starting dates from January 1999 to April 2001. The time difference between starting dates of the successive 9-months periods is 3 months. The initial cash and composition of the portfolio is:

$$c^{(0)} = 1000\,\mathrm{CHF},\ h_j^{(0)} = 5\ \forall j = 1,2,3,4\,.$$

The initial prices of the indices and the transaction costs are:

$$v_j^{(1)} = 100\,\mathrm{CHF}\ \forall j = 1,2,3,4\,,\ \delta^{bid} = \delta^{ask} = 1\%\,.$$

In the columns there are optimal period 1 holding positions $h_j^{(1)}$, $j = 1,2,3,4$ with respect to the corresponding scenario tree generations. If the expected prices are, e.g. in accordance with the market development from January 1, 1999 to September 30, 1999, then the optimal decision is: invest the whole resources to the USD index.

4. Application of IPM. The objective function can be expressed in the form $c^T x$, where x is the vector of decision variables. Nonzero elements of the vector c are the coefficients $\tilde{c}_j(\omega^{\mathcal{T}-1})$ from (4). The aim is to find the solution of the problem defined as:

$$maximize\ c^T x, subj.to\ A^{(3)}x = b. \tag{7}$$

In the three-stage stochastic model the constraint matrix $A^{(3)}$ and the corresponding vector b have the form:

$$A^{(3)} = \begin{pmatrix} A_0^{(3)} & & & & & & \\ T_1^{(3)} & A_1^{(2)} & & & & & \\ T_2^{(3)} & & A_2^{(2)} & & & & \\ T_3^{(3)} & & & A_3^{(2)} & & & \\ \cdots & & & & \cdots & & \\ \cdots & & & & & \cdots & \\ T_{N^{(3)}}^{(3)} & & & & & & A_{N^{(3)}}^{(2)} \end{pmatrix}, \; b = \begin{pmatrix} b_0^{(3)} \\ b_1^{(2)} \\ b_2^{(2)} \\ \vdots \\ b_{N^{(3)}}^{(2)} \end{pmatrix},$$

where the matrix $A_k^{(2)}, k = 1, 2, ...N^{(3)}$ represents a two-stage problem. The right-hand side vector b is split into $N^{(3)}$ sub-vectors each of them corresponding to one two-stage problem. The matrix $A_k^{(2)}$, $k = 1, 2, \ldots, N^{(3)}$ has the same block structure as the matrix $A^{(3)}$. The matrices $A_k^{(2)}$ are substituted with matrices $A_k^{(1)}$ and the matrices $T_{k,j}^{(3)}$, $k = 0, 1, 2, \ldots, N^{(3)}$ are replaced with matrices $T_k^{(2)}$. Matrices $A_k^{(2)}$ are of the size $m_k^{(2)} \times n_k^{(2)}$, whereby $m_k^{(2)} \leq n_k^{(2)}$, for $k = 1, 2, 3, \ldots N^{(3)}$. They have full row rank equal $m_k^{(2)}$. The matrices $T_k^{(3)}$ have the size conforming with the matrices $A_k^{(2)}$. The same is valid for the matrices $A_k^{(1)}$ and $T_{k,j}^{(2)}$, respectively.

There are many possibilities how the problem (7) can be solved. We have used the Interior Point Method (IPM) in the frame of the Mehrotra's Predictor Corrector algorithm MPC defined in [12], p.198. This algorithm, since 1990, has been the basis for most interior point software.
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

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x^{aff} \\ \Delta y^{aff} \\ \Delta z^{aff} \end{pmatrix} = \begin{pmatrix} r_c \\ r_b \\ r_\mu \end{pmatrix}, \tag{8}$$

where $r_b = b - Ax$, $r_c = c - z - A^T y$, $r_\mu = -XZe$; X and Z are diagonal matrices with diagonal entries x and z, respectively.
The solution of (8) can be expressed as follows:

$$\Delta y^{aff} = (ADA^T)^{-1}(r_b + AZ^{-1}(Xr_c - r_\mu)), \tag{9}$$

$$\Delta x^{aff} = Z^{-1}(XA^T \Delta y^{aff} + r_\mu - Xr_c), \tag{10}$$

$$\Delta z^{aff} = X^{-1} r_\mu - X^{-1} Z \Delta x^{aff}, \tag{11}$$

where $D = Z^{-1}X$.

Calculating the centering parameters α_{aff}^{pri}, α_{aff}^{dual} and μ_{aff} and setting $\sigma = (\mu_{aff}/\mu)^3$, where $\mu = x^T z/n$, the linear system (8) is solved again with the right-hand side $r_b = 0$, $r_c = 0$, $r_\mu = \sigma\mu e - \Delta X^{aff} \Delta Z^{aff} e$ for the solution $(\Delta x^{cc}, \Delta y^{cc}, \Delta z^{cc})$.
Computing the search direction and step to boundary from

$$\begin{aligned} (\Delta x^k, \Delta y^k, \Delta z^k) &= (\Delta x^{aff}, \Delta y^{aff}, \Delta z^{aff}) + (\Delta x^{cc}, \Delta y^{cc}, \Delta z^{cc}), \\ \alpha_k^{pri} &= argmax\{\alpha \geq 0; x^k + \alpha \Delta x^k \geq 0\}, \\ \alpha_k^{dual} &= argmax\{\alpha \geq 0; z^k + \alpha \Delta z^k \geq 0\} \end{aligned}$$

and setting the $\alpha_k^{pri} = min(0.99 * \alpha_{max}^{pri}, 1)$, $\alpha_k^{dual} = min(0.99 * \alpha_{max}^{dual}, 1)$, the values of $(x^{k+1}, y^{k+1}, z^{k+1})$ are established as

$$\begin{aligned} x^{k+1} &= x^k + \alpha_k^{pri} \Delta x^k \\ (y^{k+1}, z^{k+1}) &= (y^k, z^k) + \alpha_k^{dual} (\Delta y^k, \Delta z^k). \end{aligned}$$

The crucial step for finding the vector x in (10) is to solve the equation (9). From the computational point of view it is the most time consuming part of the algorithm. Moreover the system must be solved two times in every iteration with different right-hand sides. For the three-stage stochastic problem, it means to solve

$$(A^{(3)} D^{(3)} (A^{(3)})^t) \, \Delta y = (r_b + A^{(3)} Z^{-1} (X r_c - r_\mu)) = r^{(3)}, \tag{12}$$

where matrix $A^{(3)}$ stands as matrix A. We have us concentrated to find an effective parallel algorithm for solving the equation (12).

It has been proven in [7] that the inversion of the matrix $A^{(3)} D^{(3)} (A^{(3)})^t$ can be computed by the Sherman-Morrison-Woodbury formula as follows:

$$(\, A^{(3)} D^{(3)} (A^{(3)})^t \,)^{-1} = (\mathcal{R}^{(3)})^{-1} - (\mathcal{R}^{(3)})^{-1} U^{(3)} \, (G^{(3)})^{-1} \, (V^{(3)})^t \, (\mathcal{R}^{(3)})^{-1}, \tag{13}$$

where

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

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

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

and

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

Thus, the solution $(A^{(3)} D^{(3)} (A^{(3)})^t) \, \Delta y = r^{(3)}$ can be expressed by the inversion on the basis of the validity (13) as $\Delta y = p^{(3)} - s^{(3)}$ while

$$\mathcal{R}^{(3)} p^{(3)} = b^{(3)}, \tag{14}$$

$$G^{(3)} q^{(3)} = (V^{(3)})^t p^{(3)}, \tag{15}$$

$$\mathcal{R}^{(3)} s^{(3)} = U^{(3)} q^{(3)}. \tag{16}$$

The equations (14)-(16) 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 $R_k^{(2)}$ with corresponding right-hand side represent the basic equation of the two-stage stochastic problem [5], [6]. Thus (14) leads to the computation of the independent systems

$$\begin{aligned} p_0^{(3)} &= b_0^{(3)} \\ R_k^{(2)}\, p_k^{(3)} &= b_k^{(3)}, \quad k = 1, 2, ..., N^{(3)} \end{aligned}$$

and by the same way, we obtain from (16) the independent systems

$$\begin{aligned} s_0^{(3)} &= A_0^{(3)} q_1^{(3)} + q_2^{(3)} \\ R_k^{(2)}\, s_k^{(3)} &= T_k^{(3)} q_k^{(3)}, \quad k = 1, 2, ..., N^{(3)}, \end{aligned}$$

where $(q_1^{(3)}, q_2^{(3)})$ is the solution vector of the equation (15) with the very small size of the system matrix. The parallel three-stage procedure has been summarized in the paper [7]. The detail parallel procedure for solving the system (12) has been published in [11].

5. Implementation of large three-stage stochastic problems. Multi-period portfolio management problems lead inevitably to the solving of large linear systems (12), where the matrix of the system is sparse and has a regular structure. As an example, for a problem 81 × 81 × 16, where 4 currencies are considered, the size of the matrix $A^{(3)}$ is 33215×79716, whereby the linear system $A^{(3)} D (A^{(3)})^T$ is a system with 33215 unknowns. Because the matrix $A^{(3)}$ is a sparse matrix, it has just 159428 nonzero from total 2647766940 elements.

Our proposed method for solving system (12) uses instead of the "large" square matrix $A^{(3)} D (A^{(3)})^T$ only the "small" block matrices $A_k^{(2)}$ and $T_k^{(3)}$. Despite of this fact, by establishing the right-hand size for equation (9) we cannot avoid calculation of the matrix-vector product with the matrix $A^{(3)}$. For these reasons we have used sparse operations and stored the matrix $A^{(3)}$ in a compressed row storage form [10] form. We did not make any assumptions about the sparsity structure of the matrix and did not store any unnecessary elements. In general sparse operations are not very efficient, needing an indirect addressing step for every single scalar operation in a matrix-vector product. The compressed row storage (CRS) format puts the subsequent non-zeros of the matrix rows in contiguous memory locations and for the sparse matrix $A^{(3)}$ creates three vectors.

The application of the IPM method to the system (12) decomposes the "large" linear system into many "small" systems. The algorithms used for solving these systems are Cholesky decomposition, solving a system of linear equations, matrix-vector or matrix-matrix multiplications. For solving them we have used the linear algebra package LAPACK [8]. The LAPACK library calls were realized in the frame of an MPI implementation ([9]) targeting a distributed-memory parallel computational model. The details of the parallel implementation of stochastic three-stage algorithm are described in the paper [11]. As it is shown there the three-stage parallel programs can be parallelized in two levels. The higher level does parallel processing of several two-stage problems while of the lower level several one-stage problems are processed in parallel.

The experiments were executed on a Beowulf cluster, University of Vienna, consisting of 16 Pentium 4 (3.06GHz) processors communicating over Gigabit Ethernet. Table 2 and 3 illustrate performance results for three-stage stochastic problems of size 81 × 81 × 16 and 81 × 16 × 16 showing execution times (elapsed time) for different

NP	PxQ								
	1x2	2x1	1x4	2x2	4x1	1x8	2x4	4x2	8x1
2	28.69	15.14							
4			101.61	59.192	10.09				
8						107.84	69.4	37.04	28.44
	1x16	2x8	4x4	8x2	16x1				
16	94.57	67.41	44.98	25.4	11.35				

Table 2

Timing results in seconds for the three-stage problem $81 \times 81 \times 16$. *NP is the number of processors utilized, and PxQ specifies the configuration of the virtual process array.*

NP	PxQ								
	1x2	2x1	1x4	2x2	4x1	1x8	2x4	4x2	8x1
2	5.47	2.89							
4			18.29	10.39	1.86				
8						16.42	13.15	6.07	1.55
	1x16	2x8	4x4	8x2	16x1				
16	15.62	10.96	7.76	4.48	1.53				

Table 3

Timing results in seconds for the three-stage problem $81 \times 16 \times 16$. *NP is the number of processors utilized, and PxQ specifies the configuration of the virtual process array.*

numbers of processors and different values for P and Q. Both experiments have been measured for up to 16 processes, organized as a two-dimensional virtual process array of size $P \times Q$. P denotes the number of processes employed in parallel execution of the two-stage problems $k = 1, 2, \ldots N^{(3)}$. Each of these P processes has employed Q processes for parallel processing of one-stage problems $j = 1, \ldots M_k$ in the frame of a single two-stage problem.

All experiments proved that the crucial fact in achieving good performance is the construction of the virtual processor array. The best results were achieved in experiments with the largest number of processes designated for parallel processing of two-stage problems and so for "almost" independent processing of two-stage problems. This proves the algorithmic independence between the two-stage problems where very few communication between processes take place.

6. Conclusions and Future work. We have presented a multi-stage model for allocation of financial resources to bond indices in different currencies. The model was tested using a three-stage scenario tree with a mean-reversion property. Our results have proved that increasing the number of stages of the problem improves the quality of the solution. If the transaction costs are positive, the multi-stage decision process could cause better results than successive single-stage decisions. In the future we plan to run financial studies using historical data of interest and exchange rates to test the efficiency of the proposed model. For this purpose we plan to realize the parallel three-stage stochastic code as Grid Services using the Vienna Grid Environment (VGE) [13]. The VGE environment is based on a service-oriented architecture and has been built on top of existing standard Grid and Web Services technologies. Under VGE, parallel applications available on various HPC platforms, may be exposed via WSDL [14] as services and securely accessed by multiple remote clients over the Internet using the Simple Object Access Protocol [15]. For realizing a VGE Grid service for calculating the three-stage stochastic programs, the user must provide the input data files in a prescribed structure. As a result the solution of the system (7) will be provided. A more precise description of the whole service as well as the experimental results will be the subject of our next papers.

REFERENCES

[1] 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.

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

[3] 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.

[4] 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.

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

[6] 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.

[7] 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,Vol. 24, No. 2–3, (2003), pp. 251–265.

[8] 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, , (1999), Philadelphia PA.

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

[10] 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. 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.

[11] 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, pp.77-86 , October,23-25, 2002

[12] Stephen. J. Wright, *Primal-Dual Interior Point Methods* , SIAM 1997, ISBN 0-89871-382-X.

[13] S. Benkner, I. Brandic, G. Engelbrecht, R. Schmidt. VGE - A Service-Oriented Environment for On-Demand Supercomputing. Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing (Grid 2004), Pittsburgh, PA, USA, November 2004.

[14] Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/wsdl

[15] SOAP Version 1.2. http://www.w3.org/TR/soap/