Multiperiod Portfolio Management Using Parallel Interior Point Method

L. Halada[a] *, M. Lucka[b] and I. Melichercik[c]

[a]Institute of Informatics, Slovak Academy of Sciences,
Dubravska 9, 845 07 Bratislava, Slovakia

[b]Institute for Software Science, University of Vienna,
Liechtensteinstrasse 22, A-1090 Vienna, Austria

[c]Depart. of Economic and Financial Modeling, Faculty of Mathematics, Physics and Informatics ,
Mlynska dolina, 840 00 Bratislava, Slovakia

Computational issues occurring in finance industry demand high-speed computing when solving problems such as option pricing, risk analysis or portfolio management. In order to respond adequately on changes in the market it is necessary to evaluate information as fast as possible and to update the appropriate portfolio changes. As a consequence at present one can observe increasing interest in the development of multi-period models of portfolio management. These models introduce intermediate reallocation opportunities connected to transaction costs, which affect the composition of the portfolio at each decision instant. Stochastic programs provide an effective framework for sequential decision problems with uncertain data, when uncertainty can be modelled by a discrete set of scenarios.
In this paper we present an algorithm for solving a three-stage stochastic linear program based on the Birge and Qi factorization of a constraint matrix product in the frame of the primal-dual path-following interior point method. We outline the parallelization of the method for distributed-memory machines using Fortran/MPI and the linear algebra package LAPACK.

1. Introduction

Problems of portfolio management can be viewed as multi-period dynamic decision problems where transactions take place at discrete points in time. At each point in time a portfolio manager has to make the decision taking into account market conditions (e.g. exchange rates, interest rates) and the contemporary composition of the portfolio. Using this information the manager could sell some assets from the portfolio and using the cash from selling and other possible resources he/she buys new assets.

We present an example of a portfolio management problem. It is a model for allocation of financial resources to bond indices in different currencies. In each currency we have one index that consists of bonds issued in this currency. The whole portfolio is evaluated in the base currency. The risk one faces when making the decision is twofold: interest rate risk and exchange rate risk (future interest rates and exchange rates are uncertain). The stochastic properties are represented in the form of a scenario tree. The scenarios contain future possible developments of interest rates and exchange rates. The objective is to maximize the expected value of the portfolio at the time horizon taking into account future reallocation opportunities connected to transaction costs. 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 efforts to develop efficient parallel methods for solving this problem on parallel computer architectures [1],[2],[3],[4]. In our paper we demonstrate a parallel interior point

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

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: Section 2 establishes the problem formulation. Section 3 presents the application of the IPM to the three-stage stochastic programs. The last section discuss the issues of parallel implementation.

2. 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 at time 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.

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,
$\gamma_j^{(\tau)}(\omega^\tau),\ \omega^\tau \in \mathcal{F}_\tau$: The exchange rate base currency/ j-th currency ($\gamma_1^{(\tau)} = 1\ \forall\tau$),
$v_j^{(\tau)}(\omega^\tau),\ \omega^\tau \in \mathcal{F}_\tau$: The value of the j-th index in the j-th currency,
$\xi_j^{(\tau)}(\omega^\tau),\ \omega^\tau \in \mathcal{F}_\tau$: Bid price of the j-th index in the base currency computed by

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

$\chi_j^{(\tau)}(\omega^\tau),\ \omega^\tau \in \mathcal{F}_\tau$: Ask price of the j-th index in the base currency computed by

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

($\delta_j^{ask} > 0$ 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. $b_j^{(\tau)}(\omega^\tau) \ge 0,\quad s_j^{(\tau)}(\omega^\tau) \ge 0,\quad h_j^{(\tau)}(\omega^\tau) \ge 0\quad \forall\ 1 \le \tau < \mathcal{T}$ 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.

Inventory balance and cash-flow accounting for the Period 1

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

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

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

Risk reduction
The terminal wealth calculation is given by:

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

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:

$$\text{Maximize} \sum_{\omega^{\mathcal{T}} \in \mathcal{F}_{\mathcal{T}}} \pi(\omega^{\mathcal{T}}) WT(\omega^{\mathcal{T}}),$$

where $\pi(\omega^{\mathcal{T}})$ is the probability of the scenario $\omega^{\mathcal{T}}$.
Now, if we express the objective function in the form $c^T x$, where x is the vector of all decision variables. Our aim is to find the solution of the problem

$$\text{Maximize } c^T x, \qquad \text{subj. to} \quad A^{(3)} x = b.$$

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

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

where the matrices $A_k^{(2)}, k = 1, 2, ...N^{(3)}$ represent a two-stage problem in the frame of the whole three-stage problem. The right-hand side vector b is split on sub-vectors in accordance with matrix $A^{(3)}$. The k-th two-stage problem determined by the matrix $A_k^{(2)}$, $k = 1, 2, \ldots, N^{(3)}$ has the form

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

where $A_{k,0}^{(2)}$ is an $m_{k,0}^{(2)} \times n_{k,0}^{(2)}$ matrix and $A_{k,j}^{(1)}$ are $m_{k,j}^{(1)} \times n_{k,j}^{(1)}$ matrices. We suppose also that $m_{k,j}^{(1)} \leq n_{k,j}^{(1)}$ for all matrices $A_{k,j}^{(1)}$, $j = 1, 2, \ldots M_k$. Matrices $T_{k,j}^{(2)}$ have the size conformable to matrices $A_{k,0}^{(2)}$ and $A_{k,j}^{(1)}$. We assume moreover, that the matrices $A_0^{(3)}$, $A_{k,0}^{(2)}$ and all matrices $A_{k,j}^{(1)}$ have full row rank and $m_{k,j}^{(1)} \leq n_{k,j}^{(1)}$ for every k and j.

3. Application of the Interior Point Method

One approach for solving the problem defined above is to use the interior-point method (IPM). We have chosen the Mehrotra's Predictor Corrector algorithm MPC defined in [5], 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

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x^{aff} \\ \Delta y^{aff} \\ \Delta z^{aff} \end{pmatrix} = \begin{pmatrix} r_c \\ r_b \\ r_\mu \end{pmatrix}, \tag{1}$$

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. Calculating the centering parameters

$$\begin{aligned} \alpha_{aff}^{pri} &= argmax\{\alpha \in [0,1]; x^k + \alpha \Delta x^{aff} \geq 0\}, \\ \alpha_{aff}^{dual} &= argmax\{\alpha \in [0,1]; z^k + \alpha \Delta z^{aff} \geq 0\}, \\ \mu_{aff} &= (x^k + \alpha_{aff}^{pri} \Delta x^{aff})^T (z^k + \alpha_{aff}^{dual} \Delta z^{aff})/n \end{aligned}$$

and setting $\sigma = (\mu_{aff}/\mu)^3$, where $\mu = x^T z/n$, the linear system (1) 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

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

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

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

From the computational point of view the most time consuming part of this algorithm is solving of the system (1) with different right-hand sides. Therefore an effective parallelization of this process is very suitable. With respect to this, let us express (1) as follows:

$$\begin{aligned} \Delta y^{aff} &= (ADA^T)^{-1}(r_b + AZ^{-1}(Xr_c - r_\mu)), \\ \Delta x^{aff} &= Z^{-1}(XA^T \Delta y^{aff} + r_\mu - Xr_c), \\ \Delta z^{aff} &= X^{-1} r_\mu - X^{-1} Z \Delta x^{aff}, \end{aligned}$$

where $D = Z^{-1}X$. The crucial step for finding the unknown vectors is to solve the first equation. For our three-stage stochastic problem, it means to solve

$$(A^{(3)} D^{(3)} (A^{(3)})^t)\ \Delta y = (r_b + A^{(3)} Z^{-1}(Xr_c - r_\mu)) = r^{(3)}, \tag{2}$$

where matrix $A^{(3)}$ stands as matrix A. It has been proven in [6] that the inversion of the matrix $A^{(3)} D^{(3)} (A^{(3)})^t$ can be computed by the Sherman-Morrison-Woodbury formula as follows:

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

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

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

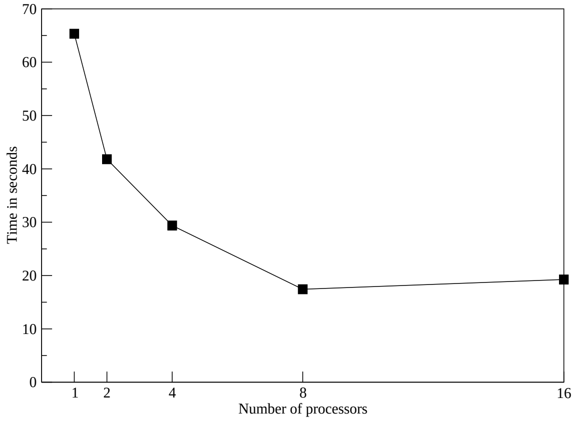

Figure 1. Execution times of experiments where the matrix $A^{(3)}$ has on the main diagonal $N^{(3)} = 9$ block matrices $A_k^{(2)}$. Every matrix $A_k^{(2)}$, $k = 1, 2, ...N^{(3)}$ has again $M_k = 9$ matrices $A_{k,j}^{(1)}$ of the size 3×6, $j = 1, 2, ...M_k$, on the main diagonal. The size of the linear system (2) in was in this case 273×546.

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 (3) as $\Delta y = p^{(3)} - s^{(3)}$ while

$$\mathcal{R}^{(3)}p^{(3)} = b^{(3)}, \tag{4}$$

$$G^{(3)}q^{(3)} = (V^{(3)})^t p^{(3)}, \tag{5}$$

$$\mathcal{R}^{(3)}s^{(3)} = U^{(3)}q^{(3)}. \tag{6}$$

The equations (4)-(6) 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 [7], [1]. The parallel three-stage procedure has been summarized in the paper [6]. The detail parallel procedure for solving the system (2) has been published in [8].

4. Parallel implementation

The implementation of the IPM method based on three-stage algorithm rely on basic algorithms of linear algebra: Cholesky decomposition, solving a system of linear equations, matrix-vector and matrix-matrix multiplication, and summation of matrices or vector, respectively. These algorithms are in the core of every multistage stochastic model and have a profound influence on the performance. For solving of linear algebra problems we have used the program package LAPACK [9], because the LAPACK library has been designed for high-performance workstations and shared-memory multiprocessors [10]. Parallel implementation of the three-stage algorithm in the frame of the IPM method is based on the Message Passing Interface (MPI) [11].

The solving of the system (2) is targeting distributed-memory parallel computers and relies on the Single Program Multiple Data (SPMD) model. The computational structure of both three-stage and two-stage algorithms for solving the above metioned system of linear equations, is very similar. Most computations are independent and can be performed in parallel. Collective communications are required only in two computational steps in every two-stage linear system and also twice in the three-stage linear system, where the collective gathering from all processes takes place. The careful analysis of the parallel three-stage algorithm is shown and explicitly described in the paper [8]. As it is shown there, all "block" rows $j = 1, 2, \ldots, M_k$ in solving the two-stage procedure $A_k^{(2)}$, can be processed in parallel. The same is true for $k = 1, 2, \ldots, N^{(3)}$ "block" rows in the matrix $A^{(3)}$, so two levels of parallel processing are possible. The above mentioned paper presents also the performance results achieved by experiments on the Beowulf cluster, University of Vienna. The algorithm based on the parallel BQ decomposition used in the frame of the MPC algorithm for three-stage stochastic problems, was implemented in the Fortran 90 programming language and executed on cluster of SMP's. The performance results of one of the experiments are illustrated on the Fig.1. Almost linear speed-up can be observed for smaller number of processors; the slow-down achieved for 16 processors was caused by increasing overhead. The size of the test problem was too "small" for this number of processors and the overhead overcame the execution. The performance results for larger problems and further implementation details will create the subject of our next paper.

5. Conclusions

We have presented a parallel method used for solving of the linear programs raised from portfolio management problems. The algorithm is based on the BQ factorization technique for three-stage stochastic programs in the context of the interior point method. Because the structure of the corresponding matrix for both three-stage and two-stage stochastic problems is regular, parallel execution in both hierarchical levels is possible. The algorithm is scalable and enables to solve large linear programs raising from the portfolio management problems.

REFERENCES

[1] 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.

[2] G. Ch. Pflug, A. Swietanovsky, *Selected parallel optimization methods for financial management under uncertainty* , Parallel Computing 26, 2000, pp. 3-25.

[3] Soren S. Nielsen, Stavros A. Zenios, *Scalable parallel Benders decomposition for stochastic linear programming*, Parallel Computing 23, 1997, pp.1069-1088.

[4] G. Ch. Pflug, H. Moritsch, M. Siomak, *Asynchronous Nested Optimization algorithms and their Parallel Implementation*, Proceedings of the International Symposium on Parallel Algorithms, Architectures and Networks, Dallas, Texas, USA, 2002.

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

[6] 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.

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

[8] 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

[9] 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.

[10] L. S. Blackford and J. J. Dongarra, *Lapack Working Note 81*, Revised version 3.0, June 30, 1999.

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