A Primal–Dual Decomposition-Based Interior Point Approach to Two-Stage Stochastic Linear Programming

Arjan Berkelaar * Cees Dert † Bart Oldenkamp ‡ Shuzhong Zhang §

April 1999. Revision July 2000.

Abstract

Decision making under uncertainty is a challenge faced by many decision makers. Stochastic programming is a major tool developed to deal with optimization with uncertainties that has found applications in, e.g. finance, such as asset-liability and bond-portfolio management. Computationally however, many models in stochastic programming remain unsolvable because of overwhelming dimensionality. For a model to be well solvable, its special structure must be explored. Most of the solution methods are based on decomposing the data. In this paper we propose a new decomposition approach for two-stage stochastic programming, based on a direct application of the path-following method combined with the homogeneous self-dual technique. Numerical experiments show that our decomposition algorithm is very efficient for solving stochastic programs. In particular, we apply our decomposition method to a two-period portfolio selection problem using options on a stock index. In this model the investor can invest in a money-market account, a stock index, and European options on this index with different maturities. We experiment our model with market prices of options on the S&P500.

AMS Classification: 49M27 (Decomposition Methods), 90C06 (Large-Scale Problems), 90C15 (Stochastic Programming).

JEL Classification: C61 (Optimization Techniques), G11 (Portfolio Choice)

*Econometric Institute, Erasmus University Rotterdam, P.O.Box 1738, 3000 DR Rotterdam, The Netherlands, berkelaar@few.eur.nl.

†ABN-AMRO Asset Management; Faculty of Economic Sciences and Econometrics, Free University of Amsterdam, The Netherlands, cees.dert@nl.abnamro.com.

‡ABN-AMRO Asset Management; Econometric Institute, Erasmus University Rotterdam, The Netherlands, oldenkamp@few.eur.nl.

§Corresponding Author. Department of Systems Engineering & Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong, zhang@se.cuhk.edu.hk.

1 Introduction

Stochastic programming plays an increasingly important role in many applications of mathematical optimization, especially in financial optimization models such as asset-liability and bond-portfolio management (the interested reader is referred to the recent book on Asset Liability Management by Mulvey and Ziemba [14]). However, efficiently solving large-scale stochastic programming problems still remains a major challenge (see [3] for an introduction to stochastic programming). A successful solution method for stochastic programming should exploit the special structure of the problem in order to cut down computational times. For this purpose, most of the solution methods in the area are based on specialized decomposition; we refer to [9] and the references therein for a survey along this direction. For multi-stage stochastic programming, the so-called L-shaped method and its variants, based on the simplex method, are very popular. With the rapid growth and development in interior point methods in recent years (cf. [17] for various survey articles on interior point methods), this traditional approach to stochastic programming needs to be reconsidered. In [5] Birge and Qi showed how decomposition can be achieved based on Karmarkar's original interior point method for two-stage stochastic linear programming. Within the interior point method realm, in fact, two types of decomposition methods have appeared. The first type, including [5], exploits the structure of two-stage stochastic linear programming which is viewed as large size linear programming; see e.g. [6, 13]. The second type of interior point decomposition methods typically specializes some of the interior point methodology, such as cutting planes or barrier methods, to solve stochastic programming; see e.g. [2, 4, 21].

In this paper we consider a new decomposition method for two-stage stochastic programming. The new method is of the first type, i.e., it is based on the homogeneous self-dual interior point method, and exploits the problem structure. The homogeneous self-dual method (HSD) for linear programming was proposed by Xu, Hung and Ye [19] as a simplification of the self-dual embedding technique of Ye, Todd and Mizuno [20]. This technique proves to be very efficient in solving linear programs (a refined version of the HSD method is actually implemented by Andersen and Andersen [1] in an optimization package called MOSEK). One of the advantages of the HSD method is that it requires no feasibility phase, allowing one to freely select any interior starting point (possibly infeasible). Moreover, the method is capable of detecting infeasibility which can be of great importance for stochastic programs. As a general merit of interior point methods, the number of iterations required to solve a linear program is typically low and insensitive to the dimension of the problem. This is an important property for solving large-scale stochastic programs. The main concern is how to implement each step of an interior point method efficiently. A great deal of attention is to be paid to this issue in the current paper. We observe that it is possible to completely decompose the direction-finding problem into subproblems, therefore enabling a decomposition-based implementation of the HSD technique. In a sense, our efforts can also be seen as trying to exploit the sparsity structure of the constrained matrix. However, unlike other general-purposed sparse matrix techniques in linear algebra, we make full (and exact) use of the structure and therefore the decomposition is best possible. We report numerical results which unambiguously show the speed-up attained when applying our decomposition algorithm compared to solving the deterministic equivalent directly by the HSD method.

As an application we consider a portfolio optimization problem. In this problem an investor wants to buy options on a given stock index, in such a way that the value of his portfolio is guaranteed to be higher than a certain level, and the probability of reaching another given level is guaranteed as well. Moreover, the expected return at the end of the investment horizon is to be maximized. We assume that there is an intermediate date at which the investor may revise his portfolio. This problem is modeled by two-stage stochastic linear programming. We solve the model using the decomposition algorithm proposed in this paper.

This paper is organized as follows. In Section 2 we discuss the generic two-stage stochastic linear program. Section 3 is dedicated to the homogeneous self-dual technique and provides a generic description of a predictor-corrector algorithm based on this HSD technique. In Section 4 we show that it is possible to completely decompose the direction-finding problem into subproblems which involve only low dimensional matrix operations. In Section 5 we report numerical results for some random test-problems. Section 6 discusses a real-world application. We solve a two-stage portfolio optimization model using options on a stock index. We conclude the paper with a summary in Section 7.

2 Two-stage stochastic programming

In this section we introduce the so-called two-stage stochastic linear programming. Interested readers are referred to two recent books on stochastic programming [3] and [10] for more details. Consider the following situation. There are two phases in a decision-making process. At the beginning of the first phase, one has to make a decision, e.g. decide the level of the inventory, or the location of a warehouse etc., without precise knowledge about the state of the world in the second stage. However, the uncertain future possibilities should be taken into account in our decision. Thus, as the reality unfolds we make a recourse decision at the second stage in order to cope with the reality being revealed so far. As an example, when the true demand of customers becomes known, the inventory and production level need to be adjusted accordingly.

In mathematical terms our problem is to find x under the constraints $Ax = b$ and $x \geq 0$. After having made this decision, one of K possible scenarios might occur. Suppose that scenario k will occur with probability π_k ($\pi_k > 0$ and $\sum_{k=1}^{K} \pi_k = 1$). In scenario k, our recourse problem, with decision variable y_k, is as follows:

$$\begin{array}{ll} \min & c_k^T y_k \\ \text{s.t.} & W_k y_k = h_k - B_k x \\ & y_k \geq 0. \end{array}$$

For technical reasons we assume that the matrices A and W_k have full row ranks. The optimal value of the above problem is a function of x. Let us denote it by $Q_k(x)$. Hence, taking into account every scenario, the expected costs under the decision x are $c^T x + \sum_{k=1}^{K} \pi_k Q_k(x)$. Putting the first and second stage decision variables all together, the optimization problem can

be formulated as:

$$\begin{array}{lll} \min & c^T x & + \ \sum_{k=1}^{K} \pi_k c_k^T y_k \\ \text{s.t.} & Ax = b & \\ & x \geq 0 & \\ & & W_k y_k = h_k - B_k x, \\ & & y_k \geq 0,\ k = 1, ..., K \end{array}$$

with its dual

$$\begin{array}{llllll} \max & b^T u & + \sum_{k=1}^{K} h_k^T v_k & & & \\ \text{s.t.} & A^T u & + \sum_{k=1}^{K} B_k^T v_k & + s & & = c \\ & & W_k^T v_k & & + z_k & = \pi_k c_k, \\ & & & s \geq 0, & z_k \geq 0, & k = 1, ..., K. \end{array}$$

In general, this can be a large size linear programming model. For practical purposes we may assume that each of the matrices A, B_k, and W_k $(k = 1, ..., K)$ are reasonably sized. However, the number of scenarios, K, might be very large.

Most of the known methods for solving the problem are based on exploiting the stair-case type structure of the constraints. For example, the so-called L-shaped method of Van Slyke and Wets [18] is a variant of Benders decomposition (dual version of the Dantzig-Wolfe decomposition). A drawback, however, of most such simplex-based methods is that the recourse matrices W_k are assumed to be constant for all k (i.e. fixed recourse). This is too restrictive in many applications. The decomposition algorithm we propose in this paper does not suffer from this restriction: whether W_k is constant or not is irrelevant.

3 The homogeneous self-dual technique

In this section we introduce the so-called homogeneous self-dual path-following method for linear programming, to put our approach in perspective. Most of the material covered in this section can be found in [19]. To make our discussion self-contained the method is reproduced here. We start by considering the following standard linear programming problem:

$$\begin{array}{lll} (P) & \min & c^T x \\ & \text{s.t.} & Ax = b \\ & & x \geq 0. \end{array}$$

The above problem has a dual:

$$\begin{array}{lll} (D) & \max & b^T y \\ & \text{s.t.} & A^T y + s = c \\ & & s \geq 0. \end{array}$$

For most optimization methods solving either (P) or (D), it is important to have an initial feasible solution to start with. This can be achieved by considering an artificial feasibility problem. Methods of this type include the two-phase method, and, in disguise, the big M-method.

In recent years, interior point methods have received intensive research in the area of optimization. It turns out that an efficient implementation of interior point methods should properly combine the primal and the dual information. The issue of initialization has led to the so-called homogeneous self-dual embedding technique, which was first proposed by Ye, Todd and Mizuno [20]. Using this technique a linear program can completely and efficiently be solved without resorting to any type of phase-one procedure. Later, this technique was generalized to more general classes of convex optimization; see [11] and the references therein. The homogeneous self-dual embedding technique of Ye, Todd and Mizuno was later simplified (and also generalized in a sense) by Xu, Hung and Ye. In their paper [19], no optimization problem is explicitly solved; instead a system of homogeneous linear equations and inequalities are approximated. This method proves to be very efficient indeed: theoretically, it retains the best known $O(\sqrt{n}L)$ iteration bound, and in practice Andersen and Andersen [1] implemented this idea in MOSEK, which is a very efficient code.

The idea of dealing with homogeneous self-dual systems can be traced back to Goldman and Tucker [8]. In [8] the following system is considered:

$$\begin{array}{rrrcl} & Ax & -b\tau & = & 0 \\ -A^Ty & & +c\tau & \geq & 0 \\ b^Ty & -c^Tx & & \geq & 0 \\ & x \geq 0, & \tau \geq 0. & & \end{array}$$

Clearly, this system is homogeneous and has a skew-symmetric constraint matrix leading to the notion of self-duality. For convenience, additional variables are introduced to replace the inequality constraints, yielding

$$(H)\left\{\begin{array}{rrrrrcl} & Ax & & -b\tau & & = & 0 \\ -A^Ty & & -s & +c\tau & & = & 0 \\ b^Ty & -c^Tx & & & -\kappa & = & 0 \\ & x \geq 0, & s \geq 0, & \tau \geq 0, & \kappa \geq 0. & & \end{array}\right.$$

If system (H) has a solution $(y^*, x^*, s^*, \tau^*, \kappa^*)$ such that $\tau^* > 0$ and $\kappa^* = 0$, then an optimal solution to (P) is simply x^*/τ^* and an optimal solution to (D) is $(y^*/\tau^*, s^*/\tau^*)$.

However, (H) also contains trivial solutions such as $(y, x, s, \tau, \kappa) = (0, 0, 0, 0, 0)$, from which no information concerning solutions for (P) and (D) can be deduced. To avoid trivial solutions, we note the following fundamental result concerning (H) due to Goldman and Tucker [8].

Theorem 1 *There exists a solution $(y^*, x^*, s^*, \tau^*, \kappa^*)$ for (H) such that*

$$x^* + s^* > 0 \quad \text{and} \quad \tau^* + \kappa^* > 0.$$

It is elementary to check that any solution (y, x, s, τ, κ) to (H) necessarily satisfies

$$x^Ts + \tau\kappa = 0.$$

That is why the Goldman-Tucker type solution is called a strictly complementary solution, since it implies that either x_i^* or s_i^* is zero for all i (and not both), and either τ^* or κ^* is zero (and not both). Based on a strictly complementary solution for (H), solutions for the original linear programming problems (P) and (D) can easily be found, as the next lemma demonstrates.

Lemma 1 *Let $(y^*, x^*, s^*, \tau^*, \kappa^*)$ be as in Theorem 1. If $\tau^* > 0$, then x^*/τ^* is an optimal solution to (P) and $(y^*/\tau^*, s^*/\tau^*)$ is an optimal solution to (D). If $\tau^* = 0$, then $\kappa^* > 0$, i.e. $b^T y^* - c^T x^* > 0$. In this case, if $b^T y^* > 0$, then (P) is infeasible, if $c^T x^* < 0$, then (D) is infeasible.*

The proof is an application of the duality theorems and Farkas' lemma. We omit the details here. Having established Theorem 1 and Lemma 1 we now concentrate on finding a strictly complementary solution for (H).

Consider an arbitrary vector $(\bar{y}, \bar{x}, \bar{s}, \bar{\tau}, \bar{\kappa})$ with $\bar{x} > 0$, $\bar{s} > 0$, $\bar{\tau} > 0$ and $\bar{\kappa} > 0$. The homogeneous self-dual algorithm ([19]) applies a modified Newton step based on that solution. To be precise, we try to find a displacement, $(d_y, d_x, d_s, d_\tau, d_\kappa)$, from the following system of linear equations:

$$(S)\quad \left\{\begin{array}{rrrrrcl} & Ad_x & & -bd_\tau & & = & \eta r_p \\ -A^T d_y & & -d_s & +cd_\tau & & = & -\eta r_d \\ b^T d_y & -c^T d_x & & & -d_\kappa & = & \eta r_g \\ & \bar{S}d_x & +\bar{X}d_s & & & = & \gamma\mu e - \bar{X}\bar{s} \\ & & & \bar{\kappa}d_\tau & +\bar{\tau}d_\kappa & = & \gamma\mu - \bar{\tau}\bar{\kappa} \end{array}\right.$$

where

$$r_p = \bar{\tau}b - A\bar{x}, \quad r_d = \bar{\tau}c - A^T\bar{y} - \bar{s} \quad \text{and} \quad r_g = c^T\bar{x} - b^T\bar{y} + \bar{\kappa}$$

are the feasibility residuals, η and γ are two parameters, and $\mu = (\bar{x}^T\bar{s} + \bar{\tau}\bar{\kappa})/(n+1)$. In this expression we used e to indicate the all-one vector, and $\bar{X}$ and $\bar{S}$ to indicate the diagonal matrices with $\bar{x}$ and $\bar{s}$ respectively on their diagonals.

Observe that when $\eta = 1$ and $\gamma = 0$, (S) is the Newton system yielding a complementary solution of (H)

$$(y', x', s', \tau', \kappa') := (\bar{y} + d_y, \bar{x} + d_x, \bar{s} + d_s, \bar{\tau} + d_\tau, \bar{\kappa} + d_\kappa).$$

This solution satisfies all the equality constraints of (H), but may fail to satisfy the non-negativity constraints and the complementarity constraints. Observe that this search direction is similar to the primal-dual affine-scaling direction. By choosing different parameters however, a procedure similar to the primal-dual path following algorithm can be constructed.

The generic homogeneous self-dual algorithm of Xu, Hung and Ye works as follows. Suppose that we have an iterate $(y^k, x^k, s^k, \tau^k, \kappa^k)$ with $x^k > 0$, $s^k > 0$, $\tau^k > 0$ and $\kappa^k > 0$. Let

$$(\bar{y}, \bar{x}, \bar{s}, \bar{\tau}, \bar{\kappa}) := (y^k, x^k, s^k, \tau^k, \kappa^k)$$

and let $\eta \in [0, 1]$ and $\gamma \in [0, 1]$. Solve the system (S) to get search directions $(d_y, d_x, d_s, d_\tau, d_\kappa)$. Choose a step-length $\alpha > 0$ such that

$$y' \quad = \quad \bar{y} + \alpha d_y$$

$$\begin{aligned} x' &= \bar{x} + \alpha d_x > 0 \\ s' &= \bar{s} + \alpha d_s > 0 \\ \tau' &= \bar{\tau} + \alpha d_\tau > 0 \\ \kappa' &= \bar{\kappa} + \alpha d_\kappa > 0. \end{aligned}$$

Let

$$(y^{k+1}, x^{k+1}, s^{k+1}, \tau^{k+1}, \kappa^{k+1}) := (y', x', s', \tau', \kappa')$$

and $k := k + 1$. Repeat the procedure until a given precision is reached.

The following lemma is proven in [19].

Lemma 2 *At each iteration of a generic homogeneous self-dual algorithm it holds that*

$$(d_x)^T d_s + d_\tau d_\kappa = \eta(n+1)(1-\gamma-\eta)\mu$$

$$\mu' = (1-\alpha\eta)[1-\alpha(1-\gamma-\eta)]\mu$$

and

$$\begin{aligned} r_p' &= (1-\alpha\eta) r_p \\ r_d' &= (1-\alpha\eta) r_d \\ r_g' &= (1-\alpha\eta) r_g. \end{aligned}$$

Based on Lemma 2, it can be shown that a predictor-corrector type implementation of the algorithm solves the problem in $O(\sqrt{n}L)$ iterations. In particular, we call a step *predictor* if $\gamma = 0$ and $\eta = 1$; a step is called *corrector* if $\gamma = 1$ and $\eta = 0$. In order to control the step-length α the following β-neighborhood is introduced:

$$\mathcal{N}(\beta) = \{(y, x, s, \tau, \kappa) \mid \| \begin{pmatrix} Xs \\ \tau\kappa \end{pmatrix} - \mu e \| \leq \beta\mu\}$$

where the norm can be either Euclidean or l_∞, corresponding to the narrow or wide neighborhood algorithms respectively. Most $O(\sqrt{n}L)$ iteration algorithms use a narrow neighborhood, except for the wide region algorithm of Sturm and Zhang [16]. In this paper we only use a narrow neighborhood in the implementation. In implementing the predictor-corrector scheme we essentially follow Lustig, Marsten, and Shanno [12]. We first compute the predictor direction d_P (with $\eta = 1$ and $\gamma = 0$). Based on this predictor direction we compute a centering parameter σ and a centered corrector direction d_C (with $\gamma = \sigma$ and $\eta = 1 - \sigma$). The centering parameter is computed as in Lustig, Marsten, and Shanno [12].

- If primal or dual feasibility has not been attained and

$$\frac{\|r_p\|_1 + \|r_d\|_1 + \|r_g\|_1}{\mu(n+1)} > 10^3.$$

  Then $\sigma = \tilde{\sigma}\mu$ (we choose $\tilde{\sigma} = 0.1$ in our implementation).

- If $\mu(n+1) < 1$ and primal and dual feasibility have been attained, then $\sigma = \mu(n+1)/\phi(n)$, where $\phi(n)$ is defined as (see [12]):

$$\phi(n) = \begin{cases} n^2 & \text{if } n \leq 5000, \\ n^{3/2} & \text{if } n > 5000. \end{cases}$$

- Otherwise compute σ as follows: compute the step-length α based on the predictor directions:

$$\alpha = -\frac{\theta}{\min(\bar{X}^{-1}d_x, d_\tau/\bar{\tau}, \bar{S}^{-1}d_s, d_\kappa/\bar{\kappa}, -\theta)}$$

(in our implementation we choose $\theta = 0.99995$). The duality gap resulting from a predictor-step is given by (using Lemma 3.2):

$$\hat{g} = (\bar{x} + \alpha d_x)^T(\bar{s} + \alpha d_s) + (\bar{\tau} + \alpha d_\tau)(\bar{\kappa} + \alpha d_\kappa) = (1-\alpha)\mu(n+1).$$

Finally, we compute the centering parameter σ as:

$$\sigma = \left(\frac{\hat{g}}{\mu(n+1)}\right)^2 \frac{\hat{g}}{n} = (1-\alpha)^3 \mu \frac{n+1}{n}.$$

Before concluding this section, we like to mention that choosing homogeneous self-dual embedding as a means to solve stochastic programming is not accidental. There are at least three main reasons that are in favor of predictor-corrector plus homogeneous self-dual embedding over other interior point methods. First, this method is known to be theoretically and practically very efficient. Many excellent codes for linear programming are based on this method. It typically requires only a small number of iteration steps to reach a high precision solution. Second, it requires no knowledge of an initial feasible solution. For large size stochastic programs, it is impractical to assume that a primal and dual feasible interior solution is available. Third, it provides Farkas type certificates in case either the primal or the dual problem is infeasible. From modeling point of view, this is extremely important: if a certain stochastic programming model is infeasible, then one should know the cause of the infeasibility.

4 Decomposing the direction-finding problem

In this section we shall investigate whether a direct implementation of the homogeneous self-dual algorithm can be applied to solve a two-stage stochastic linear program. The key is to decompose the direction finding subproblem (S). In a way, this can be seen as a special treatment of the sparsity structure of the constraint matrix as a whole. However, the steps to be developed in this section to find the search directions are specialized for the two stage stochastic linear programming. This has two consequences. First, the method is not applicable for general sparse problems. Second, no general sparsity-exploiting method can do better than the complete decomposition to be developed in this section.

We observe that the system (S) can be explicitly written as follows, when the constraint matrix of a two-stage stochastic program is used:

$$(L)\begin{cases}
 & Ad_x & & & -bd_\tau & & = & \eta r_p \\
 & B_k d_x & +W_k d_{y_k} & & -h_k d_\tau & & = & \eta r_{p_k}, \\
 & & & & & & & k=1,...,K \\
-A^T d_u & & & -\sum_{k=1}^K B_k^T d_{v_k} & +cd_\tau & -d_s & = & -\eta r_d \\
 & & & -W_k^T d_{v_k} & +\pi_k c_k d_\tau & -d_{z_k} & = & -\eta r_{d_k}, \\
 & & & & & & & k=1,...,K \\
 & Sd_x & & & & +Xd_s & = & \gamma\mu e - Xs \\
 & & Z_k d_{y_k} & & & +Y_k d_{z_k} & = & \gamma\mu e - Z_k y_k, \\
 & & & & & & & k=1,...,K \\
 & & & & \kappa d_\tau & +\tau d_\kappa & = & \gamma\mu - \tau\kappa \\
b^T d_u & -c^T d_x & -\sum_{k=1}^K \pi_k c_k^T d_{y_k} & +\sum_{k=1}^K h_k^T d_{v_k} & & -d_\kappa & = & \eta r_g.
\end{cases}$$

From the fourth and the sixth equations of (L) we obtain

$$M_k^{-1} W_k^T d_{v_k} - \pi_k M_k^{-1} c_k d_\tau - d_{y_k} = \eta M_k^{-1} r_{d_k} + Z_k^{-1}(Z_k y_k - \gamma\mu e) \tag{4.1}$$

where $M_k = Y_k^{-1} Z_k$. Multiplying this equation by W_k on both sides we further obtain

$$(W_k M_k^{-1} W_k^T) d_{v_k} - W_k d_{y_k} - \pi_k W_k M_k^{-1} c_k d_\tau = \eta W_k M_k^{-1} r_{d_k} + W_k Z_k^{-1}(Z_k y_k - \gamma\mu e).$$

Using this equation and the second equation of (L) we get

$$\begin{aligned} d_{v_k} &= (W_k M_k^{-1} W_k^T)^{-1}[-B_k d_x + (h_k + \pi_k W_k M_k^{-1} c_k) d_\tau + \eta r_{p_k}) \\ &\quad + \eta W_k M_k^{-1} r_{d_k} + W_k Z_k^{-1}(Z_k y_k - \gamma\mu e)]. \end{aligned} \tag{4.2}$$

To simplify the notation we define

$$M_0 = X^{-1} S + \sum_{k=1}^K B_k^T (W_k M_k^{-1} W_k^T)^{-1} B_k \tag{4.3}$$

$$\bar{c} = c - \sum_{k=1}^K B_k^T (W_k M_k^{-1} W_k^T)^{-1}[h_k + \pi_k W_k M_k^{-1} c_k] \tag{4.4}$$

and

$$t_0 = X^{-1}(\gamma\mu e - Xs) + \sum_{k=1}^K B_k^T (W_k M_k^{-1} W_k^T)^{-1}[\eta r_{p_k} + W_k M_k^{-1} \bar{t}_k] - \eta r_d, \tag{4.5}$$

where $\bar{t}_k = \eta r_{d_k} - Y_k^{-1}(\gamma\mu e - Z_k y_k)$. Substituting (4.2) into the third equation in (L) yields

$$-A^T d_u + M_0 d_x + \bar{c} d_\tau = t_0. \tag{4.6}$$

Now we substitute (4.6) into the first equation in (L). This gives

$$-AM_0^{-1}A^T d_u + \eta r_p + (b + AM_0^{-1}\bar{c})d_\tau = AM_0^{-1}t_0$$

and so

$$d_u = qd_\tau + \alpha \tag{4.7}$$

with

$$\begin{aligned} q &= (AM_0^{-1}A^T)^{-1}(b + AM_0^{-1}\bar{c}), \\ \alpha &= (AM_0^{-1}A^T)^{-1}(\eta r_p - AM_0^{-1}t_0). \end{aligned}$$

Eliminating d_u from (4.6) and (4.7) we get

$$d_x = pd_\tau + \beta, \tag{4.8}$$

where

$$\begin{aligned} p &= M_0^{-1}(A^T q - \bar{c}) \\ \beta &= M_0^{-1}(A^T\alpha + t_0). \end{aligned}$$

Now, we may express d_{v_k} in terms of d_τ, based on (4.2) and (4.8), as follows:

$$\begin{aligned} d_{v_k} &= (W_k M_k^{-1} W_k^T)^{-1}[(h_k + \pi_k W_k M_k^{-1} c_k)d\tau + \eta r_{p_k} - B_k\beta - B_k p d_\tau \\ &\quad + \eta W_k M_k^{-1} r_{d_k} + W_k Z_k^{-1}(Z_k y_k - \gamma\mu e)], \end{aligned}$$

and so

$$d_{v_k} = q_k d_\tau + \alpha_k \tag{4.9}$$

where

$$\begin{aligned} q_k &= (W_k M_k^{-1} W_k^T)^{-1}(h_k + \pi_k W_k M_k^{-1} c_k - B_k p) \\ \alpha_k &= (W_k M_k^{-1} W_k^T)^{-1}[\eta r_{p_k} - W_k M_k^{-1}\bar{t}_k - B_k\beta]. \end{aligned}$$

Consequently, using (4.1), we have,

$$d_{y_k} = p_k d_\tau + \beta_k, \tag{4.10}$$

where

$$\begin{aligned} p_k &= M_k^{-1}(W_k^T q_k - \pi_k c_k) \\ \beta_k &= M_k^{-1}(W_k^T\alpha_k - \bar{t}_k). \end{aligned}$$

Finally, from the seventh equation of (L) we obtain:

$$d_\kappa = \frac{\gamma\mu - \tau\kappa}{\tau} - \frac{\kappa}{\tau}d_\tau.$$

Having established the relationship between d_τ and all other variables, we now substitute (4.7), (4.8), (4.9) and (4.10) into the following identity which is obtained from the last two equations of (L):

$$b^T d_u - c^T d_x - \sum_{k=1}^{K} \pi_k c_k^T d_{y_k} + \sum_{k=1}^{K} h_k^T d_{v_k} + (\kappa/\tau) d_\tau - (\gamma\mu - \tau\kappa)/\tau = \eta r_g. \tag{4.11}$$

This finally yields

$$d_\tau = (F_1 + F_2)/(E_1 + E_2) \tag{4.12}$$

where

$$\begin{aligned} E_1 &= b^T q - c^T p + \kappa/\tau \\ F_1 &= c^T \beta - b^T \alpha + (\gamma\mu - \tau\kappa)/\tau + \eta r_g \\ E_2 &= \sum_{k=1}^{K} h_k^T q_k - \sum_{k=1}^{K} \pi_k c_k^T p_k \\ F_2 &= \sum_{k=1}^{K} \pi_k c_k^T \beta_k - \sum_{k=1}^{K} h_k^T \alpha_k. \end{aligned}$$

For convenience, we state our main result in a proposition.

Proposition 1 *The first-stage primal and dual directions can be decomposed as follows:*

$$\begin{array}{ll} d_x = p d_\tau + \beta, & d_u = q d_\tau + \alpha, \\ p = M_0^{-1}(A^T q - \bar{c}_0), & q = (A M_0^{-1} A^T)^{-1}(b + A M_0^{-1} \bar{c}), \\ \beta = M_0^{-1}(A^T \alpha + t_0), & \alpha = (A M_0^{-1} A^T)^{-1}(\eta r_p - A M_0^{-1} t_0). \end{array}$$

The second-stage primal and dual directions are decomposed as follows: for each scenario $k = 1, \ldots, K$ we have

$$\begin{array}{ll} d_{y_k} = p_k d_\tau + \beta_k, & d_{v_k} = q_k d_\tau + \alpha_k, \\ p_k = M_k^{-1}(W_k^T q_k - \pi_k c_k), & q_k = (W_k M_k^{-1} W_k^T)^{-1}(h_k + \pi_k W_k M_k^{-1} c_k - B_k p), \\ \beta_k = M_k^{-1}(W_k^T v_k - \bar{t}_k), & \alpha_k = (W_k M_k^{-1} W_k^T)^{-1}[\eta r_{p_k} - W_k M_k^{-1} \bar{t}_k - B_k \beta]. \end{array}$$

Using the expression (4.12) for d_τ, all the other variables can easily be solved by formulae (4.7), (4.8), (4.9) and (4.10). Therefore, to solve the search directions we only need to compute matrices M and Q, vectors t_0 and t_k for all $k = 1, ..., K$ and finally the quantities E_1, E_2, F_1 and F_2. In each of these computations, however, only low dimensional matrix operations are involved. This decomposition technique enables us to efficiently compute the search direction at each iteration of the homogeneous self-dual algorithm.

5 Numerical Results for Random Problems

In this section we consider the performance of our decomposition algorithm on a set of randomly generated feasible test-problems. We compare the increase of solution times as the number of

Table 1: Speed-up of decomposition approach over direct approach

Problem	Size	D.E.	Decomposition
sprand25	80 210	14	13 (0.33)
sprand50	155 410	16	15 (0.90)
sprand75	230 610	18	15 (3.3)
sprand100	305 810	18	17 (4.1)
sprand125	380 1010	20	18 (9.3)
sprand150	455 1210	18	16 (11.5)
sprand175	530 1410	21	20 (15.1)
sprand200	605 1610	24	19 (25.3)

The table shows the number of iterations and speed-ups of the decomposition algorithm and solving the deterministic equivalent directly using the homogeneous self-dual method with predictor-corrector scheme. The test-problems are randomly generated such that a feasible solution exists.

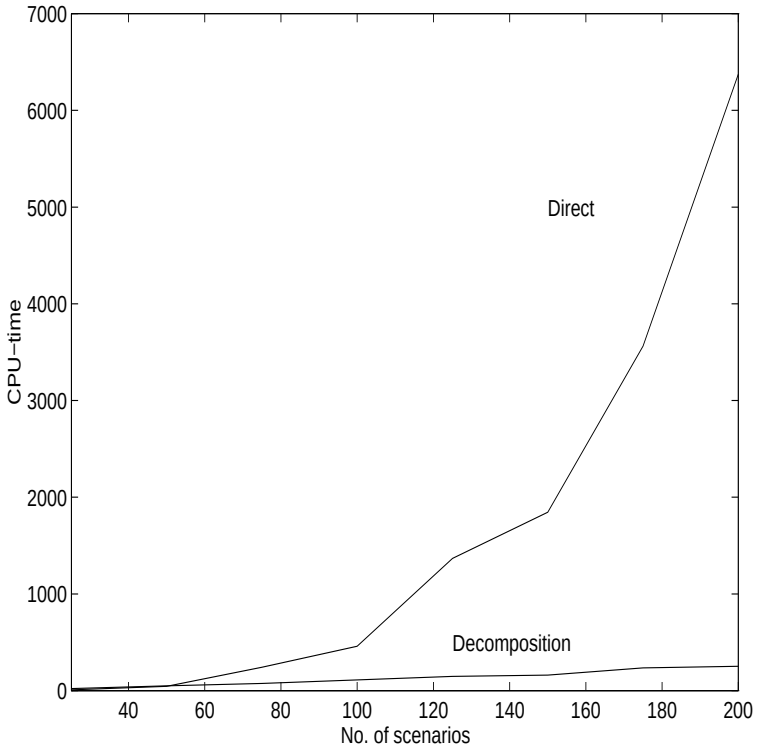
scenarios increases for both our decomposition approach and a similar implementation of our algorithm, but without decomposition. In Table 1 we show the number of iterations and the speed-ups of the decomposition algorithm over solving the deterministic equivalent directly. Only for a small number of scenarios, the direct approach performs better; however as the number of scenarios increases the decomposition algorithm is clearly superior to the direct solver. In Figure 1 we have plotted the computational times (in CPU seconds) for both the direct solver and the decomposition algorithm. This figure clearly illustrates that the decomposition algorithm performs superior. The computational times for the direct solver appear to increase quadratically with the number of scenarios, whereas the computational times for the decomposition algorithm increase only linearly with the number of scenarios. Note that also the number of iterations differ (even considerably for larger models). In principle the number of iterations of both approaches should be comparable. However, for large models the numerical linear algebra operations (e.g. Cholesky decomposition) become more involved for the deterministic equivalent, whereas the size of the sub-problems in the decomposition scheme remains constant. This accounts for more stability in the decomposition scheme. We also compared the results of our decomposition method with an implementation of the predictor-corrector interior-point method (without the homogeneous self-dual technique). The results for the latter algorithm are significantly worse. We plan to make comparisons with other decomposition algorithms in the future.

6 Guaranteed return portfolio selection

6.1 Two-stage guaranteed return portfolio model

Although the results in the previous section indicate that our decomposition method is very powerful, we only considered some simple random test-problems. We are interested in seeing how well our algorithm performs for a real world model. In this section we consider a specific

Figure 1: Number of Scenarios versus Computational Times



This figure shows the computational speed-up of our decomposition scheme (see Section 4) over solving the deterministic equivalent directly by a predictor-corrector method with the homogeneous self-dual technique. We plot CPU-time versus the number of scenarios for a set of feasible random test problems. We made a preliminary implementation of our algorithm in Matlab 5.0, Mathworks Inc. The experiments were done on PC-Pentium 100 with 64 MB Memory.

two-stage stochastic programming problem arising from an application in finance. A single-stage analog of this model was discusses in Dert and Oldenkamp [7].

We consider the following two-period problem. An investor can invest in a money-market account, a stock index, and European (exchange listed) options on this index with different maturities. We denote the stock index by S. Current time is denoted by t_0, and the expiration dates of the options by t_1 and t_2 with $t_0 < t_1 < t_2$. At t_0 the investor forms a portfolio consisting of some amount of money invested in the stock index, an investment in a zero-coupon bond maturing at t_2 and a set of options on the stock index. At time t_1 he may revise his portfolio, depending on the value of the index at t_1, i.e. he can change some of the existing positions in the options and/or buy new options starting from t_1 and maturing at t_2). The investor's goal is to guarantee that the value of the portfolio is always above a given level depending on the index at t_2, and that the expected value of the portfolio is maximized at the horizon of the investment.

Assume that the level of the stock index is S_0 at time t_0, S_1 at time t_1, and S_2 at time t_2. Moreover, there are n European puts and calls struck at K_i^j with $i = 1, 2, ..., n$, respectively, where $j = 1, 2$ denotes the expiration of the options t_j. Let $Q^p_{t_i t_j}(S) \in \mathbb{R}^n$ denote the n-dimensional vector which l-th component represents the price of buying a put option at time t_i maturing at t_j with strike price K_l, while the stock index at t_i is S. Similarly, denote $Q^c_{t_i t_j}(S) \in \mathbb{R}^n$ to be the n-dimensional vector which l-th component represent the price of buying a call option at time t_i maturing at t_j with strike price K_l while the stock index at t_i is S. The risk-free interest rate from t_0 to t_1 is denoted by r_1, the risk-free interest rate from t_0 to t_2 is denoted by r_2, and the forward rate from t_1 to t_2 is denoted by f_2. Now, let $x^p_{t_i t_j} \in \mathbb{R}^n$ denote the amount of put options purchased at time t_i maturing at t_j, and $x^c_{t_i t_j} \in \mathbb{R}^n$ be the amount of call options purchased at time t_i maturing at t_j. Let x_0^s be the amount invested in the stock index, and x_0^f be the amount invested at t_0 in the money-market account. Similarly, let x_1^s be the amount invested in the stock index and x_1^f be the amount invested in the money-market account at t_1. The decision variables $x^p_{t_0 t_j}$ and $x^c_{t_0 t_j}$ with $j = 1, 2$, and x_0^s and x_0^f denote the first-stage variables. The decision variables $x^p_{t_1 t_2}$ and $x^c_{t_1 t_2}$, and x_1^s and x_1^f denote the second-stage variables. Suppose that the initial budget for the investment is B.

Clearly, the following initial budget equation should hold:

$$B = x_0^s S_0 + x_0^f + \sum_{j=1}^{2} \langle x^p_{t_0 t_j}, Q^p_{t_0 t_j}(S_0) \rangle + \sum_{j=1}^{2} \langle x^c_{t_0 t_j}, Q^c_{t_0 t_j}(S_0) \rangle. \tag{6.13}$$

At t_1 the value of the portfolio is given by:

$$\begin{aligned} V(t_1, S_1; x^s, x^f, x^p, x^c) = x_0^s S_1 + x_0^f \exp(r_1(t_1 - t_0)) + \langle (\mathcal{K}_1 - S_1 e)^+, x^p_{t_0 t_1} \rangle + \\ + \langle (S_1 e - \mathcal{K}_1)^+, x^c_{t_0 t_1} \rangle + \langle Q^p_{t_1 t_2}(S_1), x^p_{t_0 t_2} \rangle + \langle Q^c_{t_1 t_2}(S_1), x^c_{t_0 t_2} \rangle \end{aligned} \tag{6.14}$$

where $\mathcal{K}_1 = (K_1^1, ..., K_n^1)^T$, and for given $y \in \mathbb{R}^n$, y^+ denotes the vector

$$(\max\{y_1, 0\}, ..., \max\{y_n, 0\})^T.$$

The second-stage recourse problem is as follows. First, there is an intermediate budget constraint:

$$V(t_1, S_1; x^s, x^f, x^p, x^c) = x_1^s S_1 + x_1^f + \langle Q^p_{t_1 t_2}(S_1), x^p_{t_1 t_2} \rangle + \langle Q^c_{t_1 t_2}(S_1), x^c_{t_1 t_2} \rangle. \tag{6.15}$$

Second, the value of the portfolio at the horizon is given by:

$$\begin{aligned} V(t_2, S_2; x^s, x^f, x^p, x^c) &= x_1^s S_2 + x_1^f \exp(f_2(t_2 - t_1)) + \\ &\quad + \langle (\mathcal{K}_2 - S_2 e)^+, x^p_{t_1 t_2} \rangle + \langle (S_2 e - \mathcal{K}_2)^+, x^c_{t_1 t_2} \rangle. \end{aligned} \tag{6.16}$$

We require the value of the portfolio at the horizon never to be less than $c_0 S_2 + c_1$ with $c_0 \geq 0$ and $c_1 > 0$. Using the piecewise linearity of $V(t_2, S_2; x^p, x^c)$, this yields:

$$V(t_2, K_i^2; x^s, x^f, x^p, x^c) \geq c_0 K_i^2 + c_1 \quad \text{for } i = 1, ..., n \tag{6.17}$$

and

$$V(t_2, 0; x^s, x^f, x^p, x^c) \geq c_1$$

and

$$V'_{S_2}(t_2, S_2; x^s, x^f, x^p, x^c) \mid_{S_2 = K_n^{2+}} \geq c_0.$$

These constraints are all linear in terms of x^s, x^f, x^p and x^c.

Finally, we require the probability that the portfolio value will be above a given threshold value $c_2 > 0$ to be at least λ $(0 < \lambda < 1)$. This, again by piecewise linearity, can be modeled by selecting a given I $(1 \leq I \leq n)$, and adding the following constraint:

$$V(t_2, K_i; x^s, x^f, x^p, x^c) \geq c_2 \quad \text{for } i = I, I+1, ..., n. \tag{6.18}$$

Similar constraints can be added to the model at t_1.

The expected value of the portfolio at t_2 is given by:

$$\begin{aligned} E[V(t_2, S_2; x^s, x^f, x^p, x^c)] &= x_1^s E[S_2] + x_1^f \exp(r(t_2 - t_1)) + \\ &\quad + \langle E[(\mathcal{K}_2 - S_2 e)^+], x^p_{t_1 t_2} \rangle + \langle E[(S_2 e - \mathcal{K}_2)^+], x^c_{t_1 t_2} \rangle. \end{aligned} \tag{6.19}$$

The optioned portfolio selection problem is now well defined as a two-stage stochastic linear program:

$$\begin{aligned} \max \quad & w_1 E[V(t_1, S_1; x^s, x^f, x^p, x^c)] + w_2 E[V(t_2, S_2; x^s, x^f, x^p, x^c)] \\ \text{s.t.} \quad & (6.13), (6.15), (6.17) \text{ and } (6.18) \end{aligned}$$

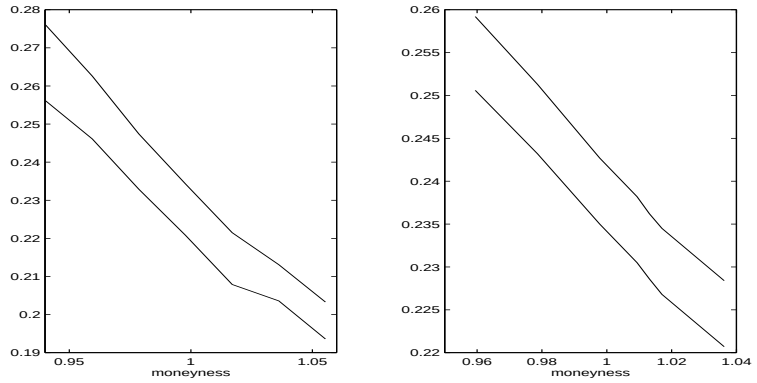
where w_1 and w_2 $(w_2 > w_1)$ are weights for the first and second stage expected values. In the numerical experiments we perform in the next section we allows for bid-ask spreads in the model and consider guaranteed constraints at both t_1 and t_2. In the next section we apply the techniques developed in Sections 3 and 4 to solve this problem.

6.2 Numerical results for two-stage guaranteed portfolio selection

In this section we present computational results for the model discussed in Section 5 based on market prices. We consider options on the Standard & Poor's 500 index. The initial date, i.e. today, is March 17, 1999, the investment horizon is June 18, 1999. The investor initially owns 1 share of the S&P500 (amounts to \$1302.84 at March 17) and he can revise his portfolio at April 16, 1999. The investor can buy and short options at March 17 with expiration at April 16 and expiration at June 18. Future option prices are based on today's implied volatility function (a more general approach where volatility is allowed to be a function of the value of the index as well can be found in Oldenkamp [15]). Today's implied volatility functions for expiration in April and June are plotted in Figure 2. To avoid arbitrage opportunities, due to a mismatch of put-call parity, we only consider call options in our analysis. The market prices of the options are displayed in Table 2. For liquidity reasons we do not use all the call options available in the market; rather we incorporate those options with moneyness between 0.94 and 1.06 only.

We generate scenarios for the first period based on a lognormal distribution with mean (annualized) 10% and volatility (annualized) 22.38% using a stratified sampling approach. Scenarios

Figure 2: Implied volatility



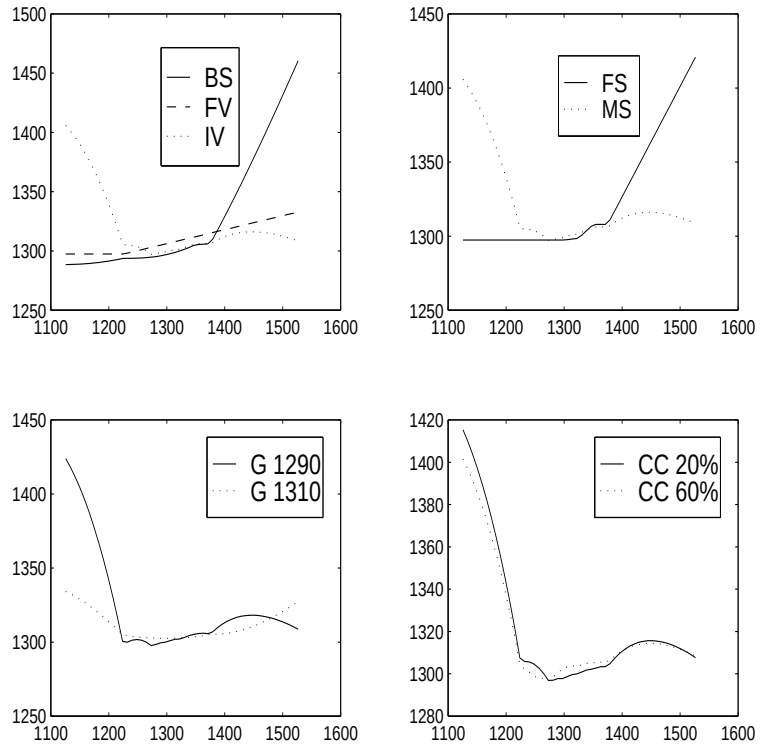
This figure shows the implied volatility for call options on the S&P500 with expiration on April 16, 1999 and June 18, 1999 based on bid and ask prices. The first plot shows implied volatility for options that expire on April 16. The second plot shows implied volatility for options that expire on June 18.

for the second period are based on the at-the-money volatility implied by today's options with expiration in June corresponding to the index level prevailing at the intermediate date. We incorporate a bid-ask spread on the index of 0.3% for both periods. The parameters of the model are summarized in Table 3. We refer to this model as the base model.

We incorporate a -5% guarantee (annualized) for each period, and impose chance constraints such that with probability 40% and 50% the investor obtains more than the risk-free investment in the first and second period respectively. A closely related model is considered by Oldenkamp [15]. For more details and a more extensive analysis we refer to his Ph.D. thesis. The solution to the base model is presented in Tables 4 and 5. From Table 5 we conclude that in many scenarios the second stage decisions have a similar structure. This indicates that we might bundle certain scenarios to capture the uncertainty. Pruning and expanding the set of scenarios in order to capture uncertainty adequately is an interesting and important topic, however we will not treat this question here.

To provide more insight in the driving forces of the guaranteed return model we consider the optimal pay-off functions at the first expiration for different instances of the model. In Figure 3 we summarize four different experiments. In the first exhibit we plot the pay-off functions for Black-Scholes and market prices. From this exhibit it is clear that the results for Black-Scholes prices and market prices is quite different. One explanation for this difference can be found by considering the second exhibit. The second exhibit illustrates the impact of different assumptions about the spread. We compare a fixed proportional spread (as used for the Black-Scholes prices) with a fixed dollar value spread (in market prices). For the model with market prices there seems to be a higher demand for far in-the-money call options than for the model with a fixed proportional spread. Looking at the solution more carefully, the investor purchases 1.68 shares of the most in-the-money call option ($K = 1225$) with shortest maturity and shorts

Figure 3: Optimal pay-off functions at first expiration (April '99)

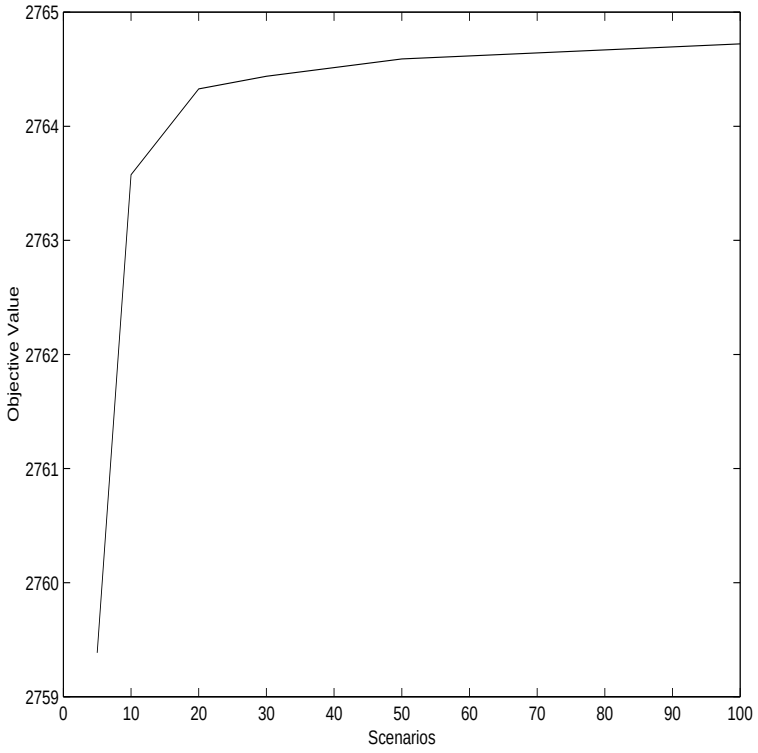


This figure shows the optimal pay-off function at the first expiration for different instances of our model. The first plot shows the optimal pay-off functions based on a) Black-Scholes (BS) prices using the estimated implied volatility function (see Figure 2), b) Black-Scholes prices with fixed volatility (FV), and c) market prices. The second plot shows the optimal pay-off functions with market prices using a fixed absolute spread (MS) and a fixed proportional spread (FS). In the third plot we display the optimal pay-off functions for different guarantee levels. The fourth plot shows the pay-off functions for different probabilities in the chance constraint.

3.64 shares of the most in-the-money call option ($K = 1250$) with expiration in June 1999. In case of a fixed proportional spread the investor only purchases out-of-the-money calls. The last two exhibits show the impact of the guarantee level and the chance constraint on the optimal pay-off function. The impact of changing the probabilities for chance constraint seems to be rather limited. Changing the guarantee level itself, however, might alter the solution more noticeably.

Since stochastic programming is concerned with discretizing the underlying random variables by means of scenarios we consider the convergence of the optimal objective as the number of scenarios increases. The number of scenarios for the second period is kept fixed (at 100 scenarios). The result is shown in Figure 4. We do not aim to provide a detailed analysis here, we merely illustrate that one should be careful in picking the number of scenarios, in order to

Figure 4: Convergence of the objective value



This figure shows the convergence of the optimal objective value as the number of scenarios increases; the number of scenarios for the second period is kept fixed at 100 scenarios.

derive stable and reliable results.

7 Summary and conclusions

In this paper we have proposed a new decomposition method for two-stage stochastic linear programming. Our algorithm is based on completely decomposing the direction-finding problem into small subproblems. We use a predictor-corrector scheme in combination with the homogeneous self-dual technique to solve the decomposed problem. We reported numerical results showing the impressive speed-up of our decomposition algorithm as compared to solving the deterministic equivalent directly. The computational times for the direct solver appear to increase at least quadratically with the number of scenarios, whereas the computational times for the decomposition method seem to increase only linearly with the number of scenarios. We have also shown that the decomposition scheme is more stable compared to solving the deterministic equivalent. As a real-world application we studied a portfolio selection problem using options.

We believe that the method proposed in this paper is very promising for several reasons. First, our algorithm requires no feasible starting point (which is a big issue in many other solution methods). Second, our algorithm is capable of detecting infeasibility and linking this infeasibility directly to a certain set of scenarios (due to the decomposition of the search-directions). Third, our algorithm provides useful information (regarding the decomposed search-directions) to perform sensitivity analysis. Fourth, our algorithm allows for stochastic recourse matrices (opposed to fixed recourse as for many decomposition algorithms). Fifth, due to the decomposition structure of the method a more efficient use of memory is possible. Finally, as a general merit of interior point methods, the number of iterations required to solve stochastic linear programs is typically low and insensitive to the dimension of the program.

References

[1] E.D. Andersen and K.D. Andersen, *The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm*, High Performance Optimization Techniques, pp. 197-232, eds. J.B.G. Frenk, K. Roos, T. Terlaky and S. Zhang, Kluwer Academic Publishers, 1999.

[2] K.A. Ariyawansa and P.L. Jiang, Polynomial cutting plane algorithms for two-stage stochastic linear programs based on ellipsoids, volumetric centers and analytic centers, *Working Paper*, Department of Pure and Applied Mathematics, Washington State University, Pullman, USA, 1996.

[3] J.R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer, New York, 1997.

[4] O. Bahn, O. du Merle, J.-L. Goffin and J.P. Vial, *A cutting plane method from analytic centers for stochastic programming*, Mathematical Programming 69, 45-73, 1995.

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

[6] I.C. Choi and D. Goldfarb, *Exploiting special structure in a primal-dual path-following algorithm*, Mathematical Programming 58, 33-52, 1993.

[7] C.L. Dert and K.P.B. Oldenkamp, *Optimal guaranteed return portfolios and the casino effect*, Technical Report, Erasmus University Rotterdam, to appear in *Operations Research*, 1998.

[8] A.J. Goldman and A.W. Tucker, Polyhedral convex cones, in H.W. Kuhn and A.W. Tucker eds., *Linear Inequalities and Related Systems*, Princeton University Press, New Jersey, pp. 19-40, 1956.

[9] J.L. Higle and S. Sen, Stochastic Decomposition: a statistical method for large-scale stochastic linear programming, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[10] P. Kall and S.W. Wallace, Stochastic Programming, John Wiley and Sons, Chichester, 1994.

[11] Z.-Q. Luo, J.F. Sturm and S. Zhang, *Duality and self-duality for conic convex programming*, Technical Report 9620/A, Erasmus University Rotterdam, 1996.

[12] I.J. Lustig, R.E. Marsten and D.F. Shanno, *On Implementing Mehrotra's Predictor-Corrector Interior-Point Method for Linear Programming*, SIAM Journal of Optimization 2, 435-449, 1992.

[13] I.J. Lustig, J.M. Mulvey and T.J. Carpenter, *The formulation of stochastic programs for interior point methods*, Operations Research 39, 757-770, 1991.

[14] J.M. Mulvey and W.T. Ziemba, *World Wide Asset and Liability Modeling*, Cambridge University Press, 1998.

[15] K.P.B. Oldenkamp, *Derivatives in Portfolio Management*, Ph.D. thesis, Erasmus University Rotterdam, Thesis Publishers Amsterdam, The Netherlands, 1999.

[16] J.F. Sturm and S. Zhang, *On a wide region of centers and primal-dual interior point algorithm for linear programming*, Mathematics of Operations Research 22, 408-431, 1997.

[17] T. Terlaky ed., *Interior Point Methods of Mathematical Programming*, Kluwer Academic Publishers, Dordrecht, 1996.

[18] R. Van Slyke and R.J.-B. Wets, *L-shaped linear programs with application to optimal control and stochastic programming*, SIAM Journal on Applied Mathematics 17, 638-663, 1969.

[19] X. Xu, P.F. Hung and Y. Ye, *A simplified homogeneous self-dual linear programming algorithm and its implementation*, Annals of Operations Research 62, 151-171, 1996.

[20] Y. Ye, M.J. Todd and S. Mizuno, *An $\mathcal{O}(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm*, Mathematics of Operations Research 19, 53-67, 1994.

[21] G.Y. Zhao, Log-barrier decomposition methods for solving stochastic programs, *Working Paper*, National University of Singapore, 1997.

Table 2: Option Prices

Moneyness	*Expiration*	*BS Price*	*Bid*	*Mid*	*Ask*	*Implied*
0.9402	April	91.33	91.25	92.25	93.25	0.2578
0.9594	April	71.28	71.125	72.125	73.125	0.248
0.9786	April	52.76	52.5	53.5	54.5	0.2353
0.9978	April	36.76	36.375	37.375	38.375	0.2238
1.0170	April	23.51	23	24	25	0.2117
1.0362	April	14.28	14	14.625	15.25	0.206
1.0554	April	7.53	7.25	7.75	8.25	0.1966
0.9594	June	101	103.125	104.125	105.125	0.2549
0.9786	June	84.38	86.25	87.25	88.25	0.2473
0.9978	June	68.9	70.5	71.5	72.5	0.2389
1.0093	June	60.45	61.875	62.875	63.875	0.2344
1.0132	June	57.63	59	60	61	0.2324
1.0170	June	54.94	56.25	57.25	58.25	0.2307
1.0362	June	43.12	44.125	45.125	46.125	0.2246

The table shows market prices of S&P500 call options with maturity April 16, 1999 and June 18, 1999, Black-Scholes prices and implied volatilities.

Table 3: Parameters and Data

Date	April 99	June 99
Scenarios	50	100
Guarantee	\$1297.36 (-5%)	\$1285.93 (-5%)
Chance Constraint	\$1307.88 (4.81%)	\$1318.976 (4.95%)
Probability	40%	50%
weight	1	1.1
riskfree rate	4.70%	4.83%
dividend	\$1.02	\$4.14

The table shows the parameters, riskfree rate and dividends for April 16, 1999 and June 18, 1999, respectively.

Table 4: First-Stage Solution

Asset	*Expiration*	Investment
exp. value		1311.41
index		0.13
risk-free		1292.36
K=1225	April	1.68
K=1275	April	0.66
K=1300	April	0.21
K=1325	April	0.31
K=1350	April	-0.01
K=1375	April	0.41
K=1250	June	-3.64

The table shows the first-stage solution.

Table 5: Second-Stage Solution

Scenario	Index	risk-free	1250	1275	1315	1350
1	1125.62	1275.25	0	0.83	-0.83	88.67
2	1133.8	1275.25	0	0.83	-0.83	69.44
3	1141.98	1275.25	0	0.83	-0.83	54.47
4	1150.16	1275.25	0	0.83	-0.83	42.62
5	1158.34	1275.25	0	0.83	-0.83	33.05
6	1166.52	1275.25	0	0.83	-0.83	25.51
7	1174.7	1275.25	0	0.83	-0.83	19.51
8	1182.88	1275.25	0	0.83	-0.83	14.66
9	1191.06	1275.25	0	0.83	-0.83	10.75
10	1199.24	1275.25	0	0.83	-0.83	7.62
11	1207.42	1275.25	0	0.83	-0.83	5.08
12	1215.6	1275.25	0	0.83	-0.83	3.04
13	1223.78	1275.25	0	0.83	-0.83	1.38
14	1231.96	1275.25	0	0.83	-0.83	1.02
15	1240.14	1275.25	0	0.83	-0.83	0.83
16	1248.32	1275.25	0	0.83	-0.83	0.63
17	1256.5	1275.25	0.51	0	-0.51	0.38
18	1264.68	1275.25	0.51	0	-0.51	0.18
19	1272.86	1275.25	0.34	0.28	-0.62	0
20	1281.04	1275.25	0	0.83	-0.83	0
21	1289.22	1275.25	0.25	0.42	-0.67	0
22	1297.4	1275.25	0	0.83	-0.83	0
23	1305.58	1275.25	0.21	0.48	-0.69	0
24	1313.76	1275.25	0.4	0.17	-0.58	0
25	1321.94	1275.25	0	0.83	-0.83	0

Scenario	Index	risk-free	1250	1315	1350
26	1330.12	1275.25	0.51	-0.51	0.01
27	1338.3	1275.25	0.51	-0.51	0.03
28	1346.48	1275.25	0.51	-0.51	0.04
29	1354.66	1275.25	0.51	-0.51	0.04
30	1362.84	1275.25	0.51	-0.51	0.03
31	1371.02	1275.25	0.51	-0.51	0.02
32	1379.2	1275.25	0.51	-0.51	0.03
33	1387.38	1275.25	0.51	-0.51	0.05
34	1395.56	1275.25	0.51	-0.51	0.06
35	1403.74	1275.25	0.51	-0.51	0.07
36	1411.92	1275.25	0.51	-0.51	0.08
37	1420.1	1275.25	0.51	-0.51	0.08
38	1428.28	1275.25	0.51	-0.51	0.08
39	1436.46	1275.25	0.51	-0.51	0.08
40	1444.64	1275.25	0.51	-0.51	0.07
41	1452.82	1275.25	0.51	-0.51	0.07
42	1461	1275.25	0.51	-0.51	0.06
43	1469.18	1275.25	0.51	-0.51	0.05
44	1477.36	1275.25	0.51	-0.51	0.05
45	1485.54	1275.25	0.51	-0.51	0.04
46	1493.72	1275.25	0.51	-0.51	0.03
47	1501.9	1275.25	0.51	-0.51	0.02
48	1510.08	1275.25	0.51	-0.51	0.02
49	1518.26	1275.25	-0.03	0	0
50	1526.44	1275.25	0.51	-0.51	0

The table shows the second-stage solution.