Stochastic Programming Models for Asset Liability Management

Roy Kouwenberg[*] Stavros A. Zenios[†]

May 2, 2001

Working Paper 01–01
HERMES Center on Computational Finance & Economics
School of Economics and Management
University of Cyprus
Nicosia, CYPRUS.

`Prepared for the` *Handbook of Asset and Liability Management* `in the series` *Handbooks in Finance*`, North-Holland.`

[*]Faculty of Commerce, University of British Columbia, 2053 Main Mall Vancouver, BC, V6T 1Z2, Canada, kouwenberg@commerce.ubc.ca

[†]HERMES Center on Computational Finance & Economics, University of Cyprus, P.O. Box 20537, 1678 Nicosia, Cyprus, and Financial Institutions Center, The Wharton School, Philadelphia, USA, zenioss@ucy.ac.cy

Abstract

Stochastic programming is a powerful modelling paradigm for asset and liability management problems. It incorporates in a common framework multiple correlated sources of risk for both the asset and liability side, takes a long time horizon perspective, accommodates different levels of risk aversion and allows for dynamic portfolio rebalancing while satisfying operational or regulatory restrictions and policy requirements. This chapter introduces stochastic programming models for broad classes of asset and liability management problems, describes procedures for generating the requisite event trees, discusses the validity of model results for illustrative applications, compares stochastic programming with alternative modelling approaches, and hinges upon solution techniques and computational issues.

Contents

1 Introduction

Asset and liability management (abbreviated: ALM) problems deal with uncertainty. They deal with the planning of financial resources in the face of uncertainty about economic, capital market, actuarial, and demographic conditions. A general approach for dealing with uncertain data is to assign to the unknown parameters a probability distribution, which should then be incorporated into an appropriate mathematical programming model. Mathematical programming models for dealing with uncertainty are known as *stochastic programs*. Stochastic programming is recognized as a powerful modelling paradigm for several areas of application. Its validity for ALM problems, in particular, is enhanced by that fact that it readily incorporates in a common framework multiple correlated sources of risk for both the asset and liability side, has long time horizons, accommodates risk aversion, and allows for dynamic portfolio rebalancing while satisfying operational or regulatory restrictions and policy requirements. Thus it facilitates an integrated view of the risk management process at an enterprise-wide level, Holmer and Zenios (1995). The applicability of stochastic programming for financial planning was recognized first by Bradley and Crane (1972) and Ziemba and Vickson (1975). But it was not until the nineties that stochastic programming started gaining prominence as a decision support tools for asset and liability management. This development was motivated in part by algorithmic advances that enabled the solution of large-scale realistic models. Globalization and innovations in the financial markets are the driving force behind the development of stochastic programming models for ALM that continues unabated to this date, aided by advances in computing technology and the availability of software.

Several academic researchers and practitioners demonstrated the effectiveness of stochastic programming models in supporting financial decision making. We mention the most recent contributions: Kusy and Ziemba (1986) for bank management, Mulvey and Vladimirou (1992) for asset allocation, Zenios (1991,1995), Golub et al. (1995) and Nielsen and Zenios (1996) for fixed income portfolio management, Carino et al. (1994), Carino and Ziemba (1998), Consigli and Dempster (1998), Høyland (1998) and Mulvey, Gould and Morgan (2000) for insurance companies, Dert (1995) for pension funds, Consiglio, Cocco and Zenios (2001) for minimum guarantee products. These are some of the applications that were done jointly with commercial sponsors and were adopted in practical settings. See also the case studies in this

Handbook; additional references are given in Zenios (1993) and Ziemba and Mulvey (1998).

This chapter reviews stochastic programming models for asset and liability management. Section 2 introduces the basics of stochastic programming and formulates a canonical model for portfolio management. The key issue of generating probabilistic data for a stochastic programming ALM system is elaborated in Section 3 on scenario generation methods. The performance of a stochastic programming ALM model for pension funds is also discussed in this section, in conjunction with alternative scenario generation methods. Section 4 places stochastic programming models in the context of the traditional portfolio choice literature from financial economics, and discusses its advantages and limitations. A brief review of stochastic programming applications to ALM in several institutional settings is given in Section 5, and references are also made to models from other chapters of this handbook. Section 6 hinges upon solution techniques illustrating the size of problems that are solvable with current state-of-the-art software. Open issues are discussed in Section 7

2 Stochastic Programming

Stochastic programming models were first formulated as mathematical programs in the late 1950s, independently, by G.B. Dantzig and E.M.L. Beale. Modern textbook treatments of stochastic programming are Kall and Wallace (1994) and Birge and Louveaux (1997), and research literature is given in the chapter by Wets (1989) or the book by Censor and Zenios (1997) which focuses on solution methods. We introduce the basics of stochastic programming and then formulate a canonical model for portfolio management. The appendix gives some background on probability theory which is essential in understanding stochastic programming with continuous random variables. Readers interested in the resulting large-scale nonlinear programs defined using discrete and finite scenario sets can do without this background.

2.1 Basic Concepts in Stochastic Programming

We formulate first two special cases of stochastic programs, the *anticipative* and the *adaptive* models. We then combine the two in the most general formulation of the *recourse* model which is the one suited for financial applications. Boldface Greek characters are used to denote random vectors which belong to some probability space as defined in the appendix.

2.1.1 Anticipative models

Consider the situation where a decision x must be made in an uncertain world where the uncertainty is described by the random vector $\boldsymbol{\omega}$. The decision does not in any way depend on future observations, but prudent planning has to anticipate possible future realizations of the random vector.

In anticipative models feasibility is expressed in terms of *probabilistic* (or *chance*) *constraints*. For example, a *reliability* level α, where $0 < \alpha \leq 1$, is specified and constraints are expressed in the form

$$P\{\boldsymbol{\omega} \mid f_j(x, \boldsymbol{\omega}) = 0, \;\; j = 1, 2, \ldots, n\} \geq \alpha,$$

where x is the m-dimensional vector of decision variables and $f_j : \mathbb{R}^m \times \Omega \to \mathbb{R}, \;\; j = 1, 2, \ldots, n$. The objective function may also be of a reliability type, such as $P\{\boldsymbol{\omega} \mid f_0(x, \boldsymbol{\omega}) \leq \gamma\}$, where $f_0 : \mathbb{R}^m \times \Omega \to \mathbb{R} \cup \{+\infty\}$ and γ is a constant.

An anticipative model selects a policy that leads to some desirable characteristics of the constraint and objective functionals under the realizations of the random vector. In the example above it is desirable that the probability of a constraint violation is less than the prespecified threshold value α. The precise value of α depends on the application at hand, the cost of constraint violation, and other similar considerations.

2.1.2 Adaptive models

In an adaptive model observations related to uncertainty become available before a decision x is made, such that optimization takes place in a learning environment. It is understood that observations provide only partial information about the random variables because otherwise the model would simply wait to observe the values of the random variables, and then make a decision x by solving a deterministic mathematical program. In contrast to this situation we have the other extreme where all observations are made after the decision x has been made, and the model becomes anticipative.

Let $\mathcal{A}$ be the collection of all the relevant information that could become available by making an observation. This $\mathcal{A}$ is a subfield of the σ-field (see the appendix) of all possible events, generated from the support set Ω of the random vector $\boldsymbol{\omega}$. The decisions x depend on the events that could be observed, and x is termed $\mathcal{A}$-*adapted* or $\mathcal{A}$-*measurable*. Using the conditional expectation with respect to $\mathcal{A}$, $E[\,\cdot\mid\mathcal{A}]$, the adaptive stochastic program can be written as:

$$
\begin{array}{llll}
 & \text{Minimize} & E[f_0(x(\boldsymbol{\omega}),\boldsymbol{\omega})\mid\mathcal{A}] & \\
(1) & \text{subject to} & E[f_j(x(\boldsymbol{\omega}),\boldsymbol{\omega})\mid\mathcal{A}]=0, & j=1,2,\dots,n, \\
 & & x(\boldsymbol{\omega})\in X, & \text{almost surely.}
\end{array}
$$

The mapping $x:\Omega\to X$ is such that $x(\boldsymbol{\omega})$ is $\mathcal{A}$-measurable. This problem can be addressed by solving for every $\boldsymbol{\omega}$ the following deterministic programs:

$$
\begin{array}{llll}
(2) & \text{Minimize} & E[f_0(x,\cdot)\mid\mathcal{A}](\boldsymbol{\omega}) & \\
(3) & \text{subject to} & E[f_j(x,\cdot)\mid\mathcal{A}](\boldsymbol{\omega})=0, & j=1,2,\dots,n, \\
(4) & & x\in X. &
\end{array}
$$

The two extreme cases (i.e., complete information with $\mathcal{A}=\Sigma$, or no information at all) deserve special mention. The case of no information reduces the model to the form of the anticipative model; when there is complete information model (1) is known as the *distribution model*. The goal in this later case is to characterize the distribution of the optimal objective function value. The precise values of the objective function and the optimal policy x are determined after realizations of the random vector $\boldsymbol{\omega}$ are observed. The most interesting situations arise when partial information becomes available after some decisions have been made, and models to address such situations are discussed next.

2.1.3 Recourse models

The *recourse problem* combines the anticipative and adaptive models in a common mathematical framework. The problem seeks a policy that not only anticipates future observations but also takes into account that observations are made about uncertainty, and thus can adapt by taking *recourse* decisions. For example, a portfolio manager specifies the composition of a portfolio considering both future movements of stock prices (anticipation) and that the portfolio will be rebalanced as prices change (adaptation).

The two-stage version of this model is amenable to formulations as a large-scale deterministic nonlinear program with a special structure of the constraint matrix. To formulate the two-stage stochastic program with recourse we need two vectors for decision variables to distinguish between the anticipative policy and the adaptive policy. The following notation is used.

$x \in \mathbb{R}^{m_0}$ denotes the vector of *first-stage* decisions. These decisions are made before the random variables are observed and are anticipative.

$y \in \mathbb{R}^{m_1}$ denotes the vector of *second-stage* decisions. These decisions are made after the random variables have been observed and are adaptive. They are constrained by decisions made at the first-stage, and depend on the realization of the random vector.

We formulate the *second-stage problem* in the following manner. Once a first-stage decision x has been made, some realization of the random vector can be observed. Let $q(y, \boldsymbol{\omega})$ denote the second-stage cost function, and let $\{T(\boldsymbol{\omega}), W(\boldsymbol{\omega}), h(\boldsymbol{\omega}) \mid \boldsymbol{\omega} \in \Omega\}$ be the model parameters. Those parameters are functions of the random vector $\boldsymbol{\omega}$ and are, therefore, random parameters. T is the *technology matrix* of dimension $n_1 \times m_0$. It contains the technology coefficients that convert the first-stage decision x into resources for the second-stage problem. W is the *recourse matrix* of dimension $n_1 \times m_1$. h is the second-stage resource vector of dimension n_1.

The second-stage problem seeks a policy y that optimizes the cost of the second-stage decision for a given value of the first-stage decision x. We denote the optimal value of the second-stage problem by $\mathcal{Q}(x, \boldsymbol{\omega})$. This value depends on the random parameters and on the value of the first-stage variables x. $\mathcal{Q}(x, \boldsymbol{\omega})$ is the optimal value, for any given Ω, of the following nonlinear program

$$
\begin{aligned}
\text{Minimize} \quad & q(y, \boldsymbol{\omega}) \\
\text{subject to} \quad & W(\boldsymbol{\omega})y = h(\boldsymbol{\omega}) - T(\boldsymbol{\omega})x, \\
& \qquad y \in \mathbb{R}^{m_1}_{+}.
\end{aligned} \tag{5}
$$

If this second-stage problem is infeasible then we set $\mathcal{Q}(x, \boldsymbol{\omega}) \doteq +\infty$. The model (5) is an *adaptation* model in which y is the *recourse* decision and $\mathcal{Q}(x, \boldsymbol{\omega})$ is the *recourse cost function.*

The two-stage stochastic program with recourse is an optimization problem in the first-stage variables x, which optimizes the sum of the cost of the first-stage decisions, $f(x)$, and the expected cost of the second-stage decisions. It is written as follows.

$$
\begin{aligned}
&\text{Minimize} && f(x) + E[\mathcal{Q}(x, \boldsymbol{\omega})] \\
&\text{subject to} && Ax = b, \\
& && x \in \mathbb{R}_+^{m_0},
\end{aligned}
\tag{6}
$$

where A is an $n_0 \times m_0$ matrix of constraint coefficients, and b is an n_0-vector denoting available resources at the first stage.

Combining (5) and (6) we obtain the following model:

$$
\begin{aligned}
&\text{Minimize} && f(x) + E[\text{Min}_{y \in \mathbb{R}_+^{m_1}} \{q(y, \boldsymbol{\omega}) \mid T(\boldsymbol{\omega})x + W(\boldsymbol{\omega})y = h(\boldsymbol{\omega})\}] \\
&\text{subject to} && Ax = b, \\
& && x \in \mathbb{R}_+^{m_0}.
\end{aligned}
\tag{7}
$$

("Min" denotes the minimal function value.)

Let $K_1 \doteq \{x \in \mathbb{R}_+^{m_0} \mid Ax = b\}$, denote the feasible set for the first-stage problem. Let also $K_2 \doteq \{x \in \mathbb{R}^{m_0} \mid E[\mathcal{Q}(x, \boldsymbol{\omega})] < +\infty\}$ denote the set of *induced constraints*. This is the set of first-stage decisions x for which the second-stage problem is feasible. Problem (6) is said to have *complete recourse* if $K_2 = \mathbb{R}^{m_0}$, that is, if the second-stage problem is feasible for any value of x. The problem has *relatively complete recourse* if $K_1 \subseteq K_2$, that is, if the second-stage problem is feasible for any value of the first-stage variables that satisfies the first-stage constraints. *Simple recourse* refers to the case when the resource matrix $W(\boldsymbol{\omega}) = I$ and the recourse constraints take the simple form $Iy_+ - Iy_- = h(\boldsymbol{\omega}) - T(\boldsymbol{\omega})x$, where I is the identity matrix, and the recourse vector y is written as $y \doteq y_+ - y_-$ with $y_+ \geq 0,\ y_- \geq 0$.

2.1.4 Deterministic equivalent formulation

We consider now the case where the random vector $\boldsymbol{\omega}$ has a discrete and finite distribution, with support $\Omega = \{\omega^1, \omega^2, \ldots, \omega^N\}$. In this case the set Ω is called a *scenario set*. Denote by p^l the probability of realization of the lth scenario ω^l. That is, for every $l = 1, 2, \ldots, N$,

$$\begin{aligned} p^l &\doteq \mathrm{Prob}(\boldsymbol{\omega} = \omega^l) \\ &= \mathrm{Prob}\left\{(q(y,\boldsymbol{\omega}), W(\boldsymbol{\omega}), h(\boldsymbol{\omega}), T(\boldsymbol{\omega})) \;=\; \Big(q(y,\omega^l), W(\omega^l), h(\omega^l), T(\omega^l)\Big)\right\}. \end{aligned}$$

It is assumed that $p^l > 0$ for all $\omega^l \in \Omega$, and that $\sum_{l=1}^N p^l = 1$.

The expected value of the second-stage optimization problem can be expressed as

$$E[\mathcal{Q}(x,\boldsymbol{\omega})] = \sum_{l=1}^{N} p^l \mathcal{Q}(x,\omega^l). \tag{8}$$

For each realization of the random vector $\omega^l \in \Omega$ a different second-stage decision is made, which is denoted by y^l. The resulting second-stage problems can then be written as:

$$\begin{aligned} &\text{Minimize} && q(y^l,\omega^l) \\ &\text{subject to} && W(\omega^l)y^l = h(\omega^l) - T(\omega^l)x, \\ &&& y^l \in \mathbb{R}_+^{m_1}. \end{aligned} \tag{9}$$

Combining now (8) and (9) we reformulate the stochastic nonlinear program (7) as the following *large-scale deterministic equivalent* nonlinear program:

$$\begin{aligned} &\text{Minimize} && f(x) + \sum_{l=1}^{N} p^l q(y^l,\omega^l) && (10) \\ &\text{subject to} && Ax = b, && (11) \\ &&& T(\omega^l)x + W(\omega^l)y^l = h(\omega^l) \qquad \text{for all } \omega^l \in \Omega, && (12) \\ &&& x \in \mathbb{R}_+^{m_0}, && (13) \\ &&& y^l \in \mathbb{R}_+^{m_1}. && (14) \end{aligned}$$

The constraints (11)–(14) for this deterministic equivalent program can be combined into a matrix equation with block-angular structure:

$$
\begin{pmatrix}
A & & & & \\
T(\omega^1) & W(\omega^1) & & & \\
T(\omega^2) & & W(\omega^2) & & \\
\vdots & & & \ddots & \\
T(\omega^N) & & & & W(\omega^N)
\end{pmatrix}
\begin{pmatrix} x \\ y^1 \\ y^2 \\ \vdots \\ y^N \end{pmatrix}
=
\begin{pmatrix} b \\ h(\omega^1) \\ h(\omega^2) \\ \vdots \\ h(\omega^N) \end{pmatrix}. \tag{15}
$$

2.1.5 Multistage models

The recourse problem is not restricted to the two-stage formulation. It is possible that observations are made at T different stages and are captured in the information sets $\{\mathcal{A}_t\}_{t=1}^T$ with $\mathcal{A}_1 \subset \mathcal{A}_2 \cdots \subset \mathcal{A}_T$. Stages correspond to time instances when some information is revealed and a decision can be made. (Note that T is a time index, while $T(\boldsymbol{\omega})$ are matrices.)

A multistage stochastic program with recourse will have a recourse problem at stage τ conditioned on the information provided by $\mathcal{A}_\tau$, which includes all information provided by the information sets $\mathcal{A}_t$, for $t = 1, 2, \ldots, \tau$. The program also anticipates the information in $\mathcal{A}_t$, for $t = \tau + 1, \ldots, T$.

Let the random vector $\boldsymbol{\omega}$ have support $\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_T$, which is the product set of all individual support sets Ω_t, $t = 1, 2, \ldots, T$. $\boldsymbol{\omega}$ is written componentwise as $\boldsymbol{\omega} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_T)$. Denote the first-stage variable vector by y_0. For each stage $t = 1, 2, \ldots, T$, define the recourse variable vector $y_t \in \mathbb{R}^{m_t}$, the random cost function $q_t(y_t, \boldsymbol{\omega}_t)$, and the random parameters $\{T_t(\boldsymbol{\omega}_t), W_t(\boldsymbol{\omega}_t), h_t(\boldsymbol{\omega}_t) \mid \boldsymbol{\omega}_t \in \Omega_t\}$.

The multistage program, which extends the two-stage model (7), is formulated as the following nested optimization problem

$$
\begin{aligned}
\text{Minimize} \quad & f(y_0) + E\left[\underset{y_1 \in \mathbb{R}_+^{m_1}}{\text{Min}} \ q_1(y_1, \boldsymbol{\omega}_1) + \cdots E\left[\underset{y_T \in \mathbb{R}_+^{m_T}}{\text{Min}} \ q_T(y_T, \boldsymbol{\omega}_T) \right] \cdots \right] \\
\text{subject to} \quad & T_1(\boldsymbol{\omega}_1) y_0 + W_1(\boldsymbol{\omega}_1) y_1 = h_1(\boldsymbol{\omega}_1), \\
& \vdots \\
& T_T(\boldsymbol{\omega}_T) y_{T-1} + W_T(\boldsymbol{\omega}_T) y_T = h_K(\boldsymbol{\omega}_T), \\
& y_0 \in \mathbb{R}_+^{m_0}.
\end{aligned} \tag{16}
$$

For the case of discrete and finitely distributed probability distributions it is again possible to formulate the multistage model into a deterministic equivalent large-scale nonlinear program.

2.2 Stochastic Programming Model for Portfolio Management

Portfolio management problems can be viewed as multiperiod dynamic decision problems where transactions take place at discrete time points. At each point in time the manager has to assess the prevailing market conditions (such as prices and interest rates) and the composition of the existing portfolio. The manager also has to assess the potential future fluctuations in interest rates, prices, and cashflows. This information is incorporated into a sequence of actions of buying or selling securities, and short-term borrowing or lending. Thus, at the next point in time the portfolio manager has a seasoned portfolio and, faced with a new set of possible future movements, must incorporate the new information so that transactions can be executed.

The model specifies a sequence of investment decisions at discrete time points. Decisions are made at the beginning of each time period. The portfolio manager starts with a given portfolio and a set of scenarios about future states of the economy which she incorporates into an investment decision. The precise composition of the portfolio depends on transactions at the previous decision point and on the scenario realized in the interim. Another set of investment decisions are made that incorporate both the current status of the portfolio and new information about future scenarios.

We develop a three-stage model, with decisions made at time instances $t_0, t_1,$

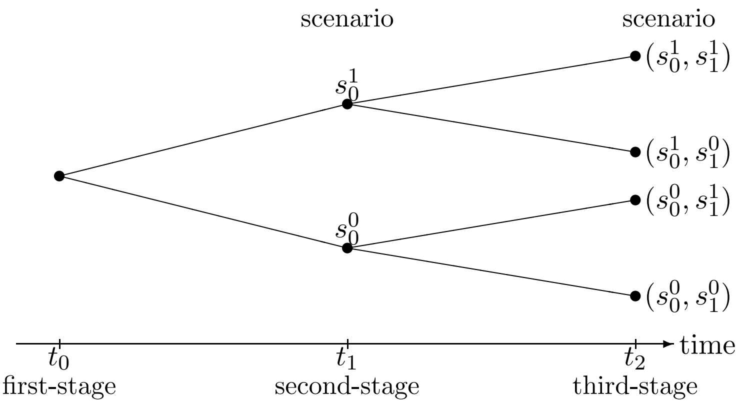


Figure 1: Scenarios and the flow of information through time on a binomial event tree.

and t_2. Extension to a multistage model is straightforward. Scenarios unfold between t_0 and t_1, and then again between t_1 and t_2. A simple three-stage problem is illustrated in the *event tree* of Figure 1. An event tree shows the flow of information across time. In the example of this figure it is assumed that scenarios evolve on a binomial tree. At instance t_0 two alternative states of nature are anticipated and by instance t_1 this uncertainty is resolved. Denote these states by s_0^0 and s_0^1. At t_1 two more states are possible, s_1^0 and s_1^1. A complete path is denoted by a pair of states and such a pair is a scenario. In this example there are four scenarios from t_0 to t_2 denoted by the pairs $(s_0^0, s_1^0), (s_0^0, s_1^1), (s_0^1, s_1^0)$, and (s_0^1, s_1^1). In the context of a multistage formulation introduced earlier the states s_0^0 and s_0^1 are indices of scenarios from the set Ω_1, and the states s_1^0 and s_1^1 indices of scenarios from Ω_2. The scenarios of the stochastic program are the pairs drawn from $\Omega = \Omega_1 \times \Omega_2$.

The stochastic programming model will determine an optimal decision for each state of the event tree, given the information available at that point. As there are multiple succeeding states the optimal decisions will not exploit hindsight, but they should anticipate future events.

2.2.1 Notation

The model is developed using variables to represent the buying and selling of securities, investments in the riskless asset and holdings of securities in the portfolio. Investment decisions are in dollars of face value. Some models in the literature—especially those dealing with strategic asset allocation—define decision variables in percentages of total wealth, which is usually normalized to 1 unit of the risk free asset. We define first the parameters of the model.

$S_0,\ S_1$: the index sets of states anticipated at t_0 and t_1 respectively. We use s_0 and s_1 to denote states from S_0 and S_1, respectively. Scenarios are denoted by pairs of the form (s_0, s_1), and with each scenario we associate a probability $p(s_0, s_1)$.

I : the index set of available securities or asset classes. The cardinality of I (i.e., number of available investment opportunities) is m.

c_0 : the dollar amount of riskless asset available at t_0.

$b_0 \in \mathbb{R}^m$: a vector whose components denote the composition of the initial portfolio.

$P_0^a,\ P_0^b \in \mathbb{R}^m$: vectors of ask and bid prices respectively, at t_0. These prices are known with certainty. In order to buy an instrument the buyer has to pay the price asked by traders, and in order to sell it the owner must be willing to accept the price at which traders are bidding.

$P_1^a(s_0),\ P_1^b(s_0) \in \mathbb{R}^m$, for all $s_0 \in S_0$: vectors of ask and bid prices, respectively, realized at t_1. These prices depend on the state s_0.

$P_2^a(s_0, s_1),\ P_2^b(s_0, s_1) \in \mathbb{R}^m$, for all $s_0 \in S_0$ and all $s_1 \in S_1$: vectors of ask and bid prices, respectively, realized at t_2. These prices depend on the scenario (s_0, s_1).

$\alpha_0(s_0),\ \alpha_1(s_0, s_1) \in \mathbb{R}^m$, for all $s_0 \in S_0$ and all $s_1 \in S_1$: vectors of *amortization factors* during the time intervals $[t_0, t_1)$ and $[t_1, t_2)$ respectively. The amortization factors indicate the fraction of outstanding face value of the securities at the end of the interval compared to the outstanding face value at the beginning of the interval. These factors capture the effects of any embedded options, such as prepayments and calls, or the effect of lapse behavior. For example, a corporate security that is called during the interval has an amortization factor of 0, and an uncalled bond has an amortization factor of 1. A mortgage security that experiences a 10 percent prepayment and that pays, through scheduled payments, an additional 5 percent of the outstanding loan has an amortization factor of 0.85. These factors depend on the scenarios.

$k_0(s_0)$, $k_1(s_0, s_1) \in \mathbb{R}^m$, for all $s_0 \in S_0$, and all $s_1 \in S_1$: vectors of *cash accrual factors* during the intervals $[t_0, t_1)$ and $[t_1, t_2)$ respectively. These factors indicate cash generated during the interval, per unit face value of the security, due to scheduled payments and exercise of the embedded options, accounting for accrued interest. For example, a corporate security that is called at the beginning of a one-year interval, in a 10 percent interest rate environment, will have a cash accrual factor of 1.10. These factors depend on the scenarios.

$\rho_0(s_0)$, $\rho_1(s_0, s_1)$: short-term riskless reinvestment rates during the intervals $[t_0, t_1)$ and $[t_1, t_2)$ respectively. These rates depend on the scenarios.

$L_1(s_0)$, $L_2(s_0, s_1)$: liability payments at t_1 and t_2 respectively. Liabilities may depend on the scenarios as discussed in section 3.1.

Now let us define decision variables. We have four distinct decisions at each point in time: how much of each security to buy, sell, or hold in the portfolio, and how much to invest in the riskless asset. All variables are constrained to be nonnegative.

First-stage variables at t_0:

$x_0 \in \mathbb{R}^m$: the components of the vector denote the face value of each security bought.

$y_0 \in \mathbb{R}^m$: denotes, componentwise, the face value of each security sold.

$z_0 \in \mathbb{R}^m$: denotes, componentwise, the face value of each security held in the portfolio .

v_0^+ : the dollar amount invested in the riskless asset.

Second-stage variables at t_1 for each state $s_0 \in S_0$:

$x_1(s_0) \in \mathbb{R}^m$: denotes the vector of the face values of each security bought.

$y_1(s_0) \in \mathbb{R}^m$: denotes the vector of the face values of each security sold.

$z_1(s_0) \in \mathbb{R}^m$: denotes the vector of the face values of each security held in the portfolio.

$v_1^+(s_0)$: the dollar amount invested in the riskless asset.

Third-stage variables at t_2 for each scenario (s_0, s_1) such that $s_0 \in S_0$ and $s_1 \in S_1$:

$x_2(s_0, s_1) \in \mathbb{R}^m$: denotes the vector of the face values of each security bought.

$y_2(s_0, s_1) \in \mathbb{R}^m$: denotes the vector of the face values of each security sold.

$z_2(s_0, s_1) \in \mathbb{R}^m$: denotes the vector of the face values of each security held in the portfolio.

$v_2^+(s_0, s_1)$: the dollar amount invested in the riskless asset.

2.2.2 Model formulation

There are two basic constraints in stochastic programming models for portfolio optimization. One expresses cashflow accounting for the riskless asset, and the other is an inventory balance equation for each asset class or each security at all time periods.

First-stage constraints:

At the first stage (i.e., at time t_0) all prices are known with certainty. The *cashflow accounting equation* specifies that the original endowment in the riskless asset, plus any proceeds from liquidating part of the existing portfolio, equal the amount invested in the purchase of new securities plus the amount invested in the riskless asset, i.e.,

$$(17) \quad c_0 + \sum_{i=1}^{m} P_{0i}^b y_{0i} = \sum_{i=1}^{m} P_{0i}^a x_{0i} + v_0^+.$$

For each asset class in the portfolio we have an *inventory balance constraint*:

$$(18) \quad b_{0i} + x_{0i} = y_{0i} + z_{0i} \text{ for all } i \in I.$$

Second-stage constraints:

Decisions made at the second stage (i.e., at time t_1) depend on the state s_0 realized during the interval $[t_0, t_1)$. Hence, we have one constraint for each state. These decisions also depend on the investment decisions made at the first stage, i.e., at t_0.

Cashflow accounting ensures that the amount invested in the purchase of new securities and the riskless asset is equal to the income generated by the existing portfolio during the holding period, plus any cash generated from sales, less the liability payments. There is one constraint for each state:

$$\begin{aligned} &(1+\rho_0(s_0))v_0^+ + \sum_{i=1}^m k_{0i}(s_0)z_{0i} + \sum_{i=1}^m P_{1i}^b(s_0)y_{1i}(s_0) \\ (19)\qquad &= v_1^+(s_0) + \sum_{i=1}^m P_{1i}^a(s_0)x_{1i}(s_0) + L_1(s_0), \text{ for all } s_0 \in S_0. \end{aligned}$$

This constraint allows investment in the riskless asset (variable v_1^+) but not borrowing. Borrowing can be incorporated in this equation by introducing a new variable v^-. Borrowing will contribute to the cash inflow (left hand side of the equation above) but borrowing from previous time periods must be paid back, with proper interest, at subsequent periods. This will increase the cash outflows (right hand side of the equation above). The cashflow accounting equation with borrowing and reinvestment at each state $s_0 \in S_0$ is written as follows:

$$\begin{aligned} &(1+\rho_0(s_0))v_0^+ + \sum_{i=1}^m k_{0i}(s_0)z_{0i} + \sum_{i=1}^m P_{1i}^b(s_0)y_{1i}(s_0) + v_1^-(s_0) \\ (20)\qquad &= v_1^+(s_0) + \sum_{i=1}^m P_{1i}^a(s_0)x_{1i}(s_0) + L_1(s_0) + v_0^-(1+\rho_0(s_0)+\delta) \end{aligned}$$

where δ is the spread between borrowing and lending rates.

Inventory balance equations constrain the amount of each security sold or remaining in the portfolio to be equal to the outstanding amount of face value at the end of the first period, plus any amount purchased at the beginning of the second stage. There is one constraint for each security and for each state:

$$(21)\quad \alpha_{0i}(s_0)z_{0i} + x_{1i}(s_0) = y_{1i}(s_0) + z_{1i}(s_0), \text{ for all } i \in I,\ s_0 \in S_0.$$

Third-stage constraints:

Decisions made at the third stage (i.e., at time t_2) depend on the scenario (s_0, s_1) realized during the period $[t_1, t_2)$ and on the decisions made at t_1. The constraints are similar to those of the second stage. The cashflow accounting equation, without borrowing, is

$$
\begin{aligned}
(1+\rho_1(s_0,s_1))v_1^+(s_0) + \sum_{i=1}^{m} k_{1i}(s_0,s_1)z_{1i}(s_0) + \sum_{i=1}^{m} P_{2i}^b(s_0,s_1)y_{2i}(s_0,s_1) \\
= v_2^+(s_0,s_1) + \sum_{i=1}^{m} P_{2i}^a(s_0,s_1)x_{2i}(s_0,s_1) + L_2(s_0,s_1), \qquad (22)
\end{aligned}
$$

for all scenarios (s_0, s_1) such that $s_0 \in S_0$ and $s_1 \in S_1$.

The inventory balance equation is:

$$
\alpha_{1i}(s_0,s_1)z_{1i}(s_0) + x_{2i}(s_0,s_1) = y_{2i}(s_0,s_1) + z_{2i}(s_0,s_1), \qquad (23)
$$

for all $i \in I$, and all scenarios (s_0, s_1) such that $s_0 \in S_0$ and $s_1 \in S_1$.

Other conditions:

At each stage of the stochastic program we have formulated two sets of constraints: cashflow accounting and the inventory balance. Depending on the application at hand other conditions may need to be modelled as constraints. The general setup with the variables as defined here is usually adequate for formulating additional constraints. We discuss several examples of conditions that appear in practice. Some applications require multiple cash accounts. For instance, international portfolio management requires different handling of cash in different currencies when exchange rates are hedged (Consiglio and Zenios 2001). Deposits from different product lines may be held in separate accounts when regulators apply different rules for different sources. This is the case for Japanese saving type insurance policies that are treated differently than conventional policies (Carino et al. 1994). Other conditions may include limits on the position in a given asset class. For instance, the allowable exposure of Italian insurers to corporate bonds or international Government bonds is limited by regulators, see the chapter by Consiglio, Cocco and Zenios in this volume. Investments in “tokkin” funds by Japanese insurers may not exceed seven percent of the total assets (Carino et al. 1994). Taxes must be computed distinguishing income return from price return, and this requirement can be formulated using the sales variables (y) to model income return, and the inventory variables (z) to model price return. Leverage restrictions may be imposed by regulators requiring the calculation of the ratio of debt to equity in funding liabilities. These, and several other conditions, may be imposed to the basic constraints formulated above.

Objective function:

The objective function maximizes the expected utility of terminal wealth. In order to measure terminal wealth all securities in the portfolio are marked-to-market. This approach is in agreement with U.S. Federal Accounting Standards Board (FASB) regulations that require reporting portfolio market and book values. The composition of the portfolio and its market value depend on the scenarios (s_0, s_1). The objective of the portfolio optimization model is

$$\text{Maximize} \sum_{(s_0,s_1)\in S_0\times S_1} p(s_0, s_1)\mathcal{U}\left(W(s_0, s_1)\right),$$

where $p(s_0, s_1)$ is the probability associated with scenario (s_0, s_1); $W(s_0, s_1)$ denotes terminal wealth; and $\mathcal{U}$ denotes the utility function. Terminal wealth is given by

$$(24) \quad W(s_0, s_1) \doteq v_2^+(s_0, s_1) + \sum_{i=1}^{m} P_{2i}^b(s_0, s_1) z_{2i}(s_0, s_1).$$

This is not by any means a standard objective function, although it is the one in agreement with the literature on discrete multi-period models (Mossin 1968, Samuelson 1969, Hakkansson 1970). Other choices may be more appropriate for some applications. For instance, Dert (1995) minimizes the expected cost of funding a defined benefits pension fund. Carino et al. (1994) consider a multicriteria objective function that maximizes terminal wealth while minimizing expected shortfalls. Consiglio, Cocco and Zenios (2001) consider the maximization of return-on-equity to shareholders as a proxy for shareholder value. For indexed funds the objective function is a measure of deviation of portfolio returns from the target index. Quite often only downside deviations are minimized. The case studies in this Handbook by Ziemba, Mulvey and Thomas, and Høyland and Wallace discuss different objective functions as well.

In general creating an objective function for investors over long time horizons is a poorly understood task. First, temporal considerations trading short return versus long-term goals must be estimated. Second, uncertainty over extended time periods complicates the decision making process by creating potential regret. Reconciling the choice of an objective function with accepted theories on investor preferences and utility functions is an important step of the modelling process.

3 Scenario Generation

An important issue for successful applications of stochastic programming models is the construction of *event trees* with asset and liability returns. As a first step, a return generating process for the assets and relevant economic factors has to specified. This task can be quite complicated, as many economic factors can affect the assets and liabilities of a large firm, pension fund or financial institution. Secondly, the liability values have to be estimated with appropriate rules taking into account actuarial risks, pension or social security fund provisions, and other relevant factors for the institution's line of business. We describe some of the simulation systems that have been proposed in the literature to handle the complicated task of scenario generation for ALM (Zenios 1991 and 1995, Mulvey and Zenios 1994, Mulvey 1996, Boender 1997, Carino et al. 1994 and 1998, amongst others).

In order to solve a multi-stage stochastic programming model for ALM, the return distributions underlying the generation process have to be discretized with a small number of nodes in the event tree. Otherwise the computational effort for solving the model would explode. Clearly, a small number of nodes in the event tree for describing the return distribution might lead to approximation error. An important question is the extent to which the approximation error in the event tree will bias the optimal solutions of the model. Moreover, event trees for ALM models with options and interest rate dependent securities require special attention to preclude arbitrage opportunities. We discuss research on these important issues, including Carino et al. (1994), Shtilman and Zenios (1993), Klaassen (1997, 1998), Pflug and Swietanowski (1998), Kouwenberg (1998), Høyland and Wallace (1999) and Gondzio, Kouwenberg and Vorst (1999).

3.1 Scenarios for the Liabilities

Any asset liability management model requires a projection of the future value of the liabilities. The liabilities typically represent the discounted expected value of the future obligatory payments by the financial institution or firm. Examples include liabilities resulting from bank deposits, pension fund or social security liabilities due to future benefit payments, and liabilities resulting from the sale of insurance contracts. Each firm and financial institution typically has its own unique set of liabilities. Hence we can not provide a general recipe for calculating the value of the liabilities. For pension funds and insurance companies actuarial methods can be very important, while other financial institutions might require financial economic valuation models (Embrechts 2000).

The liabilities of pension funds and insurance companies usually consist of a large number of individual contracts, and the development of the total liability value is influenced by multiple sources of uncertainty. As this setting frustrates mathematical analysis, simulation is an important approach for ALM applications with a complex liability structure. A simulation model must be able to capture the complex interactions between the state of the economy, the financial markets, security prices and the value of the liabilities. Figure 2 illustrates a hierarchy of simulation models that capture these interactions. User intervention is an important part of the process as some effects can not be captured by simulation models. These are sometimes called "Gorbachev effects" in reference to changes brought about by events that could not be anticipated in any simulation model.

The development of sophisticated multi-period simulation models for asset liability management is already reported in the early eighties by Goldstein and Markowitz (1982), Kingsland (1982) and Winklevoss (1982). Simulation models try to replicate the composition and development of the liability structure as closely as possible in order to increase the accuracy of liability estimates. Macroeconomic variables and actuarial predictions drive the liability side, whereas the economic variables drive the financial markets and determine security prices on the asset side.

We discuss briefly two examples to illustrate the issues. A simulation system for Dutch pension funds developed by Boender (1997) first simulates the future status of a large group of fund participants, according to assumed mortality rates, retirement rates, job termination rates and career promo-

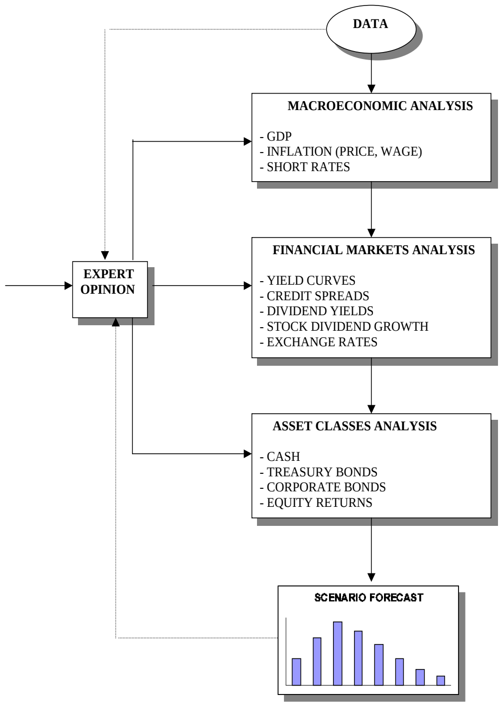


Figure 2: Hierarchy of simulation models for the generation of scenarios for asset and liability management.

tion probabilities. As a second step, values of the future wage growth are simulated, as this is an important factor for determining the pension payments in the long run. Combining the simulated status of the individual participants with the simulated wage growth, the future value of the liabilities in the scenarios is calculated as a discounted expected value of the pension payments. Established actuarial rules apply for the valuation once the economic and financial scenarios have been generated. Regulatory authorities check the solvency of pension funds by comparing the liabilities to the asset value of the fund, and hence accurate projections of the future liabilities are important for financial planning.

A different approach is taken by financial institutions, such as banks and money management firms, dealing primarily with assets and liabilities that are mostly influenced by changes in interest rates. The same is true with some products offered by insurers. The nature of their business gives rise to another important class of asset liability management problems. For instance, the bank ALM problem described by Kusy and Ziemba (1986) has fixed term deposits as the main liability. Other examples include government agencies such as Fannie Mae and Freddie Mac, that fund the purchase of mortgages by issuing debt (Holmer 1994, Zenios 1995), fixed-income money managers (Golub et al. 1995), insurance companies that offer combined insurance and investment products (Asay, Bouyoucos and Marciano 1993 and the chapter by Consiglio, Cocco and Zenios in this volume). The liabilities of defined contribution pension plans in the US are also mainly affected by interest rate changes, as fixed payments in the future are discounted with the current market interest rate. For this class of ALM problems scenario generation methods focus on the simulation of risk free rates and other key financial primitives, such as credit spreads, liquidity premia, prepayment or lapse. Defined benefits funds and social security, on the other hand, fall under the class of models of the previous paragraph as the benefits depend, through some regulatory formula, on economic indicators such as inflation or wage growth.

Methods for generating scenarios for asset liability management problems that mainly involve interest rate dependent securities are described in the above references. First, a lattice for the short term interest rate is constructed using a model such as the one suggested by Black, Derman and Toy (1990) or Hull and White (1990). An important property is that the prices of treasury bonds computed with this lattice are consistent with the initial yield curve. Second, the prices of other relevant interest rate dependent securities (such as mortgage backed securities, single-premium deferred annuities, callable bonds) and the value of the liabilities are added to the lattice by applying financial economic valuation rules and simulations for other factors such as prepayments or lapse. Finally, consistent scenarios of interest rate movements, fixed income prices and liability values can be constructed by sampling paths from the lattice. This methodology generates price scenarios under the risk-neutral probability measure. For short horizons the risk neutral and the objective measure are indistinguishable and lattice-based scenario generation is valid. For long horizons the methodology breaks down, except for some problems involving index replication, and a risk premium must be properly estimated and incorporated in the valuation stage.

An important distinction must be made in this class of models between state-dependent and path-dependent instruments. In the former case the price of an instrument is uniquely determined at each state of the lattice. In the later case the prices depend on the path that leads to a given state. While the number of states is a polynomial function of the number of steps, the number of paths grows exponentially. A 360-step lattice of monthly steps over 30-years has $360^2/2$ states but 2^{360} paths. High-performance computations may be needed for the simulation of path-dependent securities (Cagan, Carriero and Zenios 1993).

3.2 Scenarios for Economic Factors and Asset Returns

Asset liability management applications typically require simulation systems that integrate the asset prices with the liability values. This integration is crucial as the assets and liabilities are often affected by the same underlying economic factors. For example, in pension fund simulations wage growth and inflation are crucial factors for the value of the liabilities, and these factors are also associated with the long run returns on stocks and bonds (Boender 1997). In fixed income ALM applications for money management the short term interest rate is driving the returns on both assets and liabilities (Zenios 1995). We echo the first chapter of this handbook that the integration of assets and liabilities is crucial for successful ALM applications at an enterprise-wide level. The integration starts with the consistent simulation of future scenarios for both sides of the balance sheet. As the liabilities are often unique and different in each ALM application, we will from now on concentrate on generating scenarios for economic factors and asset returns. Values of the liabilities may be added to the economic scenarios with a consistent method following actuarial practices or standard financial valuation tools (Embrechts 2000).

Perhaps the most complete instantiation of the framework illustrated in Figure 2 is the scenario generation system developed by the company Towers Perrin for pension management problems (Mulvey 1996). The economic forecasting system consists of a linked set of modules that generate scenarios for different economic factors and asset returns. At the highest level of the system, the Treasury yield curve is modelled by a two-factor model based on Brennan and Schwartz (1982). Other models could have been used here as well, perhaps accounting for market shocks. Based on the scenarios for the short and consol rates, other modules generate forecasts of the price inflation, bond returns and the dividend yield on stocks. After the return on a major stock index (e.g., the S&P 500) has been generated conditional on the dividend yield, the return on corporate bonds and small cap stocks are derived at the lowest level of the system. The cascade design of the Towers Perrin scenario generation system limits the number of coefficients that have to be estimated with the available data and leads to consistent forecasts for the returns on a large number of assets.

Other models generating asset returns are described in Brennan, Schwartz and Lagnado (1997) for strategic asset allocation, Carino et al. (1994, 1998) for an insurance company, Consiglio, Cocco and Zenios (this volume) for minimum guarantee products, amongst others. These approaches do not model in detail the economic conditions. Given the interest-rate dependence of the liabilities in these studies, and the strategic decisions they address, this omission was not significant. However, a more general model is needed for defined benefits pension funds, social security funds, long term insurance products and so on.

Once the model for generating scenarios has been specified, the coefficients have to be calibrated in order to produce plausible values for the returns. For example, the Towers Perrin system consists of a number of diffusions for the key economic factors such as the interest rate and the dividend yield. The coefficients of these diffusions have to be estimated: one can apply a pragmatic approach that matches historical summary statistics and expert opinions (Høyland and Wallace 1999, Mulvey, Gould and Morgan 2000) or traditional econometric methods for discrete-time models (Green 1990, Hamilton 1994) and for continuous time-models (Duffie and Singleton

1993, Hansen and Scheinkman 1995). ALM applications with fixed income securities, such as mortgage assets, are often based on interest rate lattice models. These require the calibration of a lattice that perfectly matches the current yield curve of treasury bills and bonds. Proper calibration ensures that the coefficients of a model are consistent with historical data or current prices (Black, Derman and Toy 1990, Hull and White 1990). As scenarios are projections of the future, the users of ALM models can of course adjust the estimated coefficients in order to incorporate their own views about the economy and the asset markets (Koskosides and Duarte 1997). Sometimes stress scenarios are incorporated in response to requirements by the supervisory authorities or to satisfy corporate safeguards.

3.3 Methods for Generating Scenarios

In this subsection we describe three specific methods for generating asset return scenarios with more detail: *(i)* bootstrapping historical data, *(ii)* statistical modelling with the Value-at-Risk approach, and *(iii)* modelling economic factors and asset returns with vector autoregressive models.

3.3.1 Bootstrapping historical data

The simplest approach for generating scenarios using only the available data without any mathematical modelling is to bootstrap a set of historical records. Each scenario is a sample of returns of the assets obtained by sampling returns that were observed in the past. Dates from the available historical records are selected randomly and for each date in the sample we read the returns of all asset classes or risk factors during the month prior to that date. This are scenarios of *monthly* returns. If we want to generate scenarios of returns for a long horizon—say 1 year—we sample 12 monthly returns from different points in time. The compounded return of the sampled series is the 1-year return. Note that with this approach the correlations among asset classes are preserved.

3.3.2 Statistical models from the Value-at-Risk literature

Time series analysis of historical data can be used to estimate volatilities and correlation matrices among asset classes of interest. Riskmetrics (1996) has become an industry standard in this respect. These correlation matrices are used to measure risk exposure of a position through the Value-at-Risk (VaR) methodology.

Denote the random variables by the K-dimensional random vector $\boldsymbol{\omega}$. The dimension of $\boldsymbol{\omega}$ is equal to the number of risk factors we want to model. Assuming that the random variables are jointly normally distributed we can define their probability density function of $\boldsymbol{\omega}$ by

$$(25)\quad f(\boldsymbol{\omega}) = (2\pi)^{-p/2} \mid Q \mid^{-1/2} \exp\left[-\frac{1}{2}(\boldsymbol{\omega}-\bar{\boldsymbol{\omega}})' Q^{-1} (\boldsymbol{\omega}-\bar{\boldsymbol{\omega}})\right],$$

where $\bar{\boldsymbol{\omega}}$ is the expected value of $\boldsymbol{\omega}$ and Q is the covariance matrix and they can be calculated from historical data. (It is typically the case in financial time series to assume that the logarithms of the changes of the random variables have the above probability density function, so that the variables themselves follow a lognormal distribution.)

Once the parameters of the multivariate normal distribution are estimated we can use it in Monte Carlo simulations, using either the standard Cholesky factorization approach (see, e.g., RiskMetrics, 1996, ch. 7) or scenario generation procedures based on principal component analysis discussed in Jamshidian and Zhu (1997).

The simulation can be applied repeatedly at different states of an event tree. However, we may want to condition the generated random values on the values obtained by some of the random variables. For instance, users may have views on some of the variables, or a more detailed model may be used in the simulation hierarchy to estimate some of the variables. This information can be incorporated when sampling the multivariate distribution.

The conditional sampling of multivariate normal variables proceeds as follows. Variable $\boldsymbol{\omega}$ is partitioned into two subvectors $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$, where ω_1 is the vector of dimension K_1 of random variables for which some additional information is available and $\boldsymbol{\omega}_2$ is the vector of dimension $K_2 = K - K_1$ of the remaining variables. The expected value vector and covariance matrix are partitioned similarly as

$$(26)\quad \bar{\boldsymbol{\omega}} = \begin{bmatrix} \bar{\boldsymbol{\omega}}_1 \\ \bar{\boldsymbol{\omega}}_2 \end{bmatrix} \text{ and } Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}.$$

The marginal probability density function of $\boldsymbol{\omega}_2$ given $\boldsymbol{\omega}_1 = \boldsymbol{\omega}_1^*$ is given by $f(\boldsymbol{\omega}_2 \mid \boldsymbol{\omega}_1 = \boldsymbol{\omega}_1^*) =$

$$(27)\quad (2\pi)^{-p_2/2} \mid Q_{22.1} \mid^{-1/2} \exp\left[-\frac{1}{2}(\boldsymbol{\omega}_2 - \bar{\boldsymbol{\omega}}_{2.1})' Q_{22.1}^{-1}(\boldsymbol{\omega}_2 - \bar{\boldsymbol{\omega}}_{2.1})\right],$$

where the conditional expected value and covariance matrix are given by

$$(28)\quad \bar{\boldsymbol{\omega}}_{2.1}(\boldsymbol{\omega}_1^*) = (\bar{\boldsymbol{\omega}}_2 - Q_{21}Q_{11}^{-1}\mu_1) + Q_{21}Q_{11}^{-1}\boldsymbol{\omega}_1^*,$$

and

$$(29)\quad Q_{22.1} = Q_{22} - Q_{21}Q_{11}^{-1}Q_{12},$$

respectively. Scenarios of $\boldsymbol{\omega}_2$ for period t conditioned on values of $\boldsymbol{\omega}_1$ given by ω_1^* can be generated from the multivariate normal variables from (27) through the expression

$$\boldsymbol{\omega}_{2i}^t = \boldsymbol{\omega}_{2i}^0 \exp\left[\sigma_i \sqrt{t} \boldsymbol{\omega}_{2i}\right],$$

where $\boldsymbol{\omega}_{2i}^0$ is today's value and σ_i is the single-period volatility of the ith component of the random variable ω_2.

Consiglio and Zenios (2001) use the Riskmetrics methodology *in conjunction* with discrete lattice models to generate joint scenarios of term-structure and exchange rates. Interest rate differentials among two countries are key determinants of the exchange rate between the currencies. Hence, exchange rate scenarios are conditioned on the interest rates of the two currencies, the base currency and the foreign currency. The standard assumption applies that the logarithms of the ratios of exchange rates at period t to period $t-1$, and the logarithms of the ratios of spot interest rates at period t to period

$t-1$ follow a multivariate normal distribution. Daily and weekly rates do not follow normal distributions but there is lack of empirical evidence against normality for monthly data such as those used by Consiglio and Zenios.

Figure 3 illustrates the conditional probabilities for exchange rate scenarios. On the same figure we plot the exchange rate that was realized *ex post* on the date for which the scenarios were estimated. Note that the same exchange rate value may be obtained for various different scenarios of interest rates and samples drawn from (27); the figure plots several points with the same exchange rate value but different conditional probabilities.

3.3.3 Modelling economic factors and asset returns for a pension fund

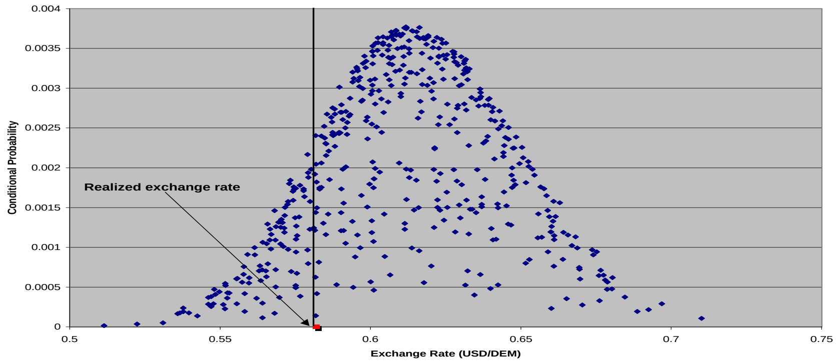
In order to illustrate the calibration of a system of equations for economic factor values and asset returns, we consider an ALM simulation system for Dutch pension funds as an example (see Boender 1997). As the scope of ALM systems for Dutch pension funds is often limited to long term strategic decisions, the investment model only considers a small set of broad asset classes: deposits, bonds, real estate and stocks. Apart from the returns on these assets, each scenario should contain information about future wage growth in order to calculate the future values of the pension liabilities.

In order to generate asset returns and the wage growth rate a vector autoregressive model is applied by Boender (1997):

$$(30) \quad R_t = c + Vh_{t-1} + \epsilon_t, \ \epsilon_t \sim N(0, Q), \ t = 1, 2, \ldots, T,$$
$$(31) \quad R_{it} = ln(1 + r_{it}), \ i = 1, 2, \ldots, m, \ t = 1, 2, \ldots, T,$$

where m is the number of asset time series, r_{it} is the discrete rate of change of variable i in year t, R_t is an m-dimensional vector of continuously compounded rates, c is the m-dimensional vector of coefficients, V is an $m \times m$ matrix of coefficients, ϵ_t is the m-dimensional vector of error terms and Q is the $m \times m$ covariance matrix. In order to estimate the coefficients of the model we will yearly data on the asset returns and the general wage increase in the Netherlands from 1956 to 1994. Table 1 displays the descriptive statistics of the data and Table 2 displays the correlation matrix.

The specification of the vector autoregressive model should be chosen care-



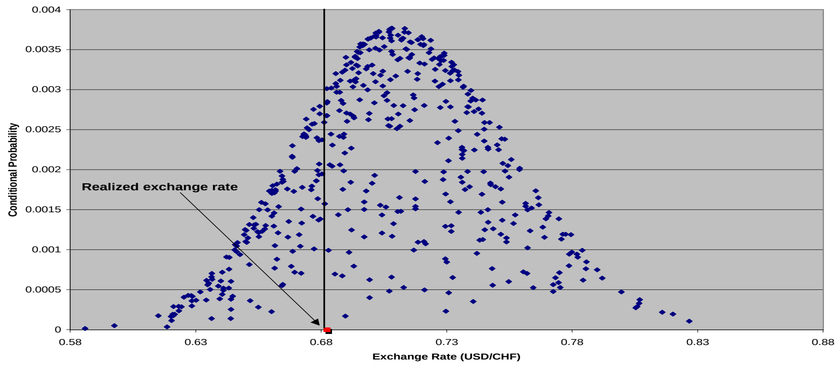


Figure 3: Exchange rate scenarios and their conditional probabilities for the DEM and CHF against the USD.

Table 1: Statistics of time series for asset returns, 1956-1994.[1]

	mean	st.dev.	skewness	kurtosis
wages	0.061	0.044	0.434	2.169
deposits	0.055	0.025	0.286	2.430
bonds	0.061	0.063	0.247	3.131
real estate	0.081	0.112	-0.492	7.027
stocks	0.102	0.170	0.096	2.492

[1] Wages is the rate of change of the Dutch general wage level. The time series for deposits is based on the average of the 3-month borrowing rate for government agencies. In each year a premium of 0.5% has been subtracted, because the pension fund will have to lend cash to commercial banks. The asset class of bonds represents the total return of a roll-over investment in long term Dutch government bonds. Real estate consists of total returns of the property fund Rodamco. Stocks is the total return of the internationally diversified mutual fund Robeco. All timeseries were provided by Ortec Consultants.

Table 2: Correlations of asset classes, annually 1956-1994.

	wages	deposits	bonds	real estate	stocks
wages	1				
deposits	-0.059	1			
bonds	-0.127	0.259	1		
real estate	0.162	-0.053	0.360	1	
stocks	-0.296	-0.157	0.379	0.326	1

Table 3: Coefficients of the vector autoregressive model for asset returns in the Dutch markets.[1]

$ln(1 + wages_t)$	$=$	0.018 (2.058)	$+$	$0.693\ ln(1 + wages_{t-1}) + \epsilon_{1t}$ (5.789)	$\sigma_1 = 0.030$
$ln(1 + deposits_t)$	$=$	0.020 (2.865)	$+$	$0.644\ ln(1 + deposits_{t-1}) + \epsilon_{2t}$ (5.448)	$\sigma_2 = 0.017$
$ln(1 + bonds_t)$	$=$	0.058 (6.241)	$+$	ϵ_{3t}	$\sigma_3 = 0.060$
$ln(1 + realestate_t)$	$=$	0.072 (4.146)	$+$	ϵ_{4t}	$\sigma_4 = 0.112$
$ln(1 + stocks_t)$	$=$	0.086 (3.454)	$+$	ϵ_{5t}	$\sigma_5 = 0.159$

[1] Estimated with iterative weighted least squares using annual data for the period 1956-1994, t-statistics in parenthesis. σ_i denotes the mean square error (standard deviation of the residuals) for each asset return equation $i = 1, 2, \ldots, 5$.

Table 4: Residual correlations of vector autoregressive model.

	wages	deposits	bonds	real estate	stocks
wages	1				
deposits	0.227	1			
bonds	-0.152	-0.268	1		
real estate	-0.008	-0.179	0.343	1	
stocks	-0.389	-0.516	0.383	0.331	1

fully. Although some inter-temporal relationships between the returns might be weakly significant based on historical data, that does not imply that these relationships are also useful for generating scenarios for a financial optimization model with a long time horizon. To avoid any problems with unstable and spurious predictability of returns, we do not use lagged variables for explaining the returns of bonds, real estate, and stocks in the vector autoregressive model (Table 3). The timeseries of the return on deposits and the increase of the wage level on the other hand are known to have some memory, so we model them by a first order autoregressive process.

There are many ways to estimate vector autoregressive models, see, e.g., Judge et al. (1988). Table 3 shows the coefficients estimated by the method of iterative weighted least squares, using the econometric software Eviews; Table 4 displays the estimated correlation matrix of the residuals. We would like to point out that the average return on bonds is rather low in our sample of data, due to two outliers in 1956 and 1957. In order to generate plausible future bond returns, we choose to increase the coefficient of the bond returns in the vector autoregressive model by 1%. For the purpose of generating scenarios we can adjust the coefficients based on subjective expectations, as historical data is not necessarily representative for the future. User intervention may be required as illustrated in Figure 2 and discussed in Koskosides and Duarte (1997).

Finally, scenarios of asset returns for a financial planning model can be constructed by sampling from the error distribution of the vector autoregressive model, given the estimated (and possibly user-adjusted) coefficients of Table 3. After the vector autoregressive model has been used to generate scenarios of asset returns and wage growth, the liability values can be added to each scenario in a consistent manner by applying appropriate actuarial rules or financial valuation principles (Embrechts 2000).

3.4 Constructing Event Trees

A stochastic programming model is based upon an event tree for the key random variables (Figure 1). Each node of the event tree has multiple successors, in order to model the process of information being revealed progressively through time. The stochastic programming approach will determine an optimal decision for each node of the event tree, given the information available at that point. As there are multiple succeeding nodes the optimal decisions will be determined without exploiting hindsight. If a stochastic programming model is formulated then the optimal policy will be tailor-made to fit the condition of the state of financial institution and the economy in each node, while anticipating the optimal adjustment of the policy later on as the tree evolves and more information is revealed.

A key issue for the successful application of stochastic programming in ALM is the construction of event trees with asset and liability returns from the scenarios generated by the processes discussed in Sections 3.1 and 3.2. The underlying return distributions have to be discretized with a small number of nodes in the event tree, otherwise the computational effort for solving a multi-stage stochastic programming model can easily explode. Clearly, a small number of nodes describing the return distribution at every stage of the event tree might lead to some approximation error. An important question is to which extent the approximation error in the event tree will bias the optimal solutions of the model.

In this section we will consider three different methods to construct event trees for stochastic programming models: (*i*) random sampling, (*ii*) adjusted random sampling, and (*iii*) tree fitting. In order to compare these methods we will apply them to construct trees with asset and liability returns for the estimated vector autoregressive model of Table 3. We will also demonstrate the importance of constructing proper event trees by solving the multi-stage stochastic programming model for a Dutch pension fund using these event trees as input.

The stochastic programming model for pension funds of Kouwenberg (1998) minimizes the average contribution rate by the pension plan sponsor, while taking into account the risk of deficits and the state of the fund at the planning horizon (see also Dert 1995). Risk aversion is modelled with a quadratic penalty on deficits at the end of the horizon. In particular the objective function is the second downside moment of the ratio of assets to liabilities. The decision variables of the model are the contribution rate and the strategic asset mix, consisting of bonds, stocks and real estate (see Table 3). The equations of the model consist of budget equations, restrictions on the yearly change of the contribution rate and restrictions on the investment strategy (e.g., no short selling). Transaction costs on trading are also included in the model. For more details of the model we refer to Kouwenberg (1998).

The next section applies the three methods for constructing event trees from the scenarios generated by the vector autoregressive model. However these methods are applicable to any scenario generation model such as the Towers Perrin model, the lattice-based models, or the models discussed later in Section 3.6

3.4.1 Random sampling

As a first method for constructing event trees we consider random sampling from the error distribution of the vector autoregressive model. Given the estimated coefficients and the estimated covariance matrix of the vector autoregressive model, we can draw one random vector of yearly returns for bonds, real estate, stocks, deposits and wage growth. If we would like to construct an event tree with ten nodes after one year (we assume that the duration of each stage is one year), we can simply repeat this procedure ten times, sampling independent vectors of returns for each node. The nodes at stage two in the event tree can also be sampled randomly, however the conditional distribution from stage one to stage two depends on the outcomes at the first stage. For example, wage growth follows an autoregressive process, so the expected wage growth from year one to year two depends on the realized wage growth rate in the previous period. The sampled wage growth data at stage one will be input to the right hand side of the wage equation (Table 3) when sampling error terms at stage two.

We create an entire event tree for the stochastic program by applying random sampling recursively, from stage to stage, while adjusting the conditional expectations of wage growth and deposits in each node based on previous outcomes. Applying random sampling to construct an event tree consisting of five (yearly) stages and a branching structure of 1-10-6-6-4-4 we obtain a tree with $10 \times 6 \times 6 \times 4 \times 4 = 5760$ nodes at time period 5. We solved the stochastic programming model of Kouwenberg (1998) with this randomly sampled event tree as input. Borrowing and lending are not allowed in the model, in order to limit the number of decision variables. The asset weights are restricted to be nonnegative to prevent short selling. Transaction costs of 1% are imposed both on buying and on selling of assets.

Table 5 contains information about the optimal solution of the stochastic programming model. The initial asset mix—first row of the table—is the part of the solution that could be implemented. In this case the model recommends that the pension fund invests 100% of its endowment in stocks.

Table 5: Solution of Stochastic Programming Model: Random Sampling[1]

Initial asset mix	Bonds	Real	Stocks
	0.00	0.00	1.00
Portfolio turnover	Bonds	Real	Stocks
	0.31	0.37	0.42
Objective Value	0.2294		
Mean Contribution Rate	0.0126		
Downside Risk	0.0302		

[1] The turnover behavior of the optimal asset policy through time is measured by the average absolute change of the asset weights. Risk is measured by the second downside moment of the funding ratio, while the costs of funding the liabilities are measured by the average contribution rate. The objective function is to minimize the costs, while adding a penalty for downside risk.

The mean absolute change of the asset weights at subsequent stages is 37%. This indicates major changes in the asset mix through time; on average 1/3rd of the portfolio is turned over at each period. The average contribution rate for the asset mix recommended by this model equals 1.26%.

Given the large turnover of the optimal investment strategy, we should regard the solution with some suspicion. The sparse branching structure of the event tree is in part the culprit. At each time period there are no more than ten states to represent the underlying conditional distribution of five time series. Moreover, these states are sampled randomly. As a result, the mean and covariance matrix will most likely be specified incorrectly in most nodes of the tree. As a result the optimizer chooses an investment strategy that has the best return and downside risk characteristics, but based on an erroneous approximation of the return distribution.

The random sampling procedure for constructing a sparse multi-period event tree apparently leads to unstable investment strategies. An obvious way to deal with this problem is to increase the number of nodes in the randomly sampled event tree, in order to reduce the approximation error relative to the vector autoregressive model. However, the stochastic program might become computationally intractable if we increase the number of nodes at each stage, due to the exponential growth rate of the tree. Alternatively, the switching of asset weights might be bounded by adding constraints to the model or enforcing robustness through the choice of an objective function (Mulvey, Vanderbei and Zenios 1995). Although we might get a more stable solution in this case, the underlying problem remains the same: the optimal decisions are based on an erroneous representation of the return distributions in the event tree.

3.4.2 Adjusted random sampling

An adjusted random sampling technique for constructing event trees can resolve some of the problems of the simple random sampling method. First, assuming an even number of nodes, we apply antithetic sampling in order to fit every odd moment of the underlying distribution. For example, if there are ten succeeding nodes at each stage then we sample five vectors of error terms from the vector autoregressive model. The error terms for the five remaining nodes are identical but with opposite signs. As a result we match every odd moment of the underlying error distributions (note that the errors have a mean of zero). Second, we rescale the sampled values in order to fit the variance. This can be achieved by multiplying the set of sampled returns for each particular asset class by an amount proportional to their distance from the mean, as in Carino et al. (1994). In this way the sampled errors are shifted away from their mean value, thus changing the variance until the target value is achieved. The adjusted values for the error terms are substituted in the estimated equations of the vector autoregressive model to generate a set of nodes for the event tree.

We now solve the model for the Dutch pension funds using the event tree generated with the adjusted random sampling method. Table 6 displays the solution. The portfolio composition in this case is 57% in stocks and 43% in bonds, and the average turnover has decreased to 12.3%. Compared to the results obtained with standard random sampling (Table 5) the average contribution rate has risen from 1.26% to 5.83%, and both downside risk and the objective value of the model have worsened.

It seems to be more difficult to make investment profits when using an event tree constructed with adjusted random sampling, compared to plain random sampling. This is a strong signal that the profits in Table 5 were actually spurious, based on a flawed description of the underlying return distributions. A second signal is the reduction of asset mix switching due to the adjustment of the random samples. The heavy trading activity in

Table 6: Solution of Stochastic Programming Model: Adjusted Random Sampling

Initial asset mix	Bonds	Real	Stocks
	0.00	0.43	0.57
Turnover	Bonds	Real	Stocks
	0.16	0.09	0.12
Objective Value	0.5013		
Mean Contribution Rate	0.0583		
Downside Risk	0.0375		

the optimal solution of Table 5 probably results from random errors in the mean returns, to which portfolio optimization models are very sensitive (see Chopra and Ziemba 1993). Using adjusted random sampling to match the mean and the variance, we substantially reduce useless trading. The additional computational effort for adjusting the random samples is negligible.

3.4.3 Fitting the mean and the covariance matrix

A third method for constructing event trees is to estimate returns that match the first few moments of the underlying return distributions. This can be achieved by solving a non-linear optimization model following Høyland and Wallace (1999). The decision variables in the optimization model are the returns and the probabilities of the event tree, while the objective function and the constraints enforce the desired statistical properties. The probabilities and returns in all nodes of the event tree can be estimated simultaneously. However, with this approach it might take longer to construct a desirable event tree than to solve the stochastic programming model for ALM itself. The tree fitting problem can be simplified by applying the method at each stage recursively as suggested in Kouwenberg (1998). This requires the assumption that the return distributions are not path dependent. This assumption is valid for long term asset and liability management with broad asset classes, but fails when modelling money management problems with path-dependent securities.

To illustrate the concepts, we write down the tree fitting equations to estimate a set of perturbations that will fit the mean and the residual covariance matrix of the vector autoregressive process. The probabilities are assumed uniform in order to ease comparison with random sampling. Let $i = 1, 2, \ldots, m$, denote the random time series that are modelled by the vector autoregressive process. In our example these are the returns on stocks, bonds, deposits, real estate and the wage growth rate. Suppose that a total of M succeeding nodes at stage $t+1$ are available to describe the conditional distribution of these random variables in a particular node at stage t. We define the perturbation ϵ_{ti}^l as the realization in node l for the ith element of the vector ϵ_t.

A tree fitting model that matches the mean of zero and the estimated covariance of vector autoregressive model (30)–(31) estimates the perturbations by solving equations (32) and (33). Equation (32) specifies that the average of the perturbations should be zero, while equation (33) states that they should have a covariance matrix equal to Q:

$$\frac{1}{M}\sum_{l=1}^{M}\epsilon_{ti}^{l} = 0, \text{ for all } i = 1, 2, \ldots, m, \tag{32}$$

$$\frac{1}{(M-1)}\sum_{l=1}^{M}\epsilon_{ti}^{l}\epsilon_{tj}^{l} = Q_{ij}, \text{ for all } i = 1, 2, \ldots, m, \; j = 1, 2, \ldots, m. \tag{33}$$

Obtaining a solution of the non-linear system (32)–(33) can be difficult, specially when higher order moments like skewness and kurtosis are also included as additional restrictions. Instead of solving a system of nonlinear equations we may solve instead a non-linear optimization model that penalizes deviations from the desired moments in the objective function. Good starting points for this optimization can be obtained using the adjusted random sampling method of the previous subsection, which is computationally very efficient. After solving the non-linear fitting model, we can substitute the optimal set of perturbations in the estimated equations of the vector autoregressive model to generate conditional return distributions. By applying this procedure recursively, from node to node and from stage to stage, we generate an event tree that fits the time varying conditional expectation and the covariance matrix of the underlying return distributions.

We have applied this method to construct a fitted 1-10-6-6-4-4 event tree. For the later stages, with four succeeding nodes each, we fitted only the

Table 7: Solution of Stochastic Programming Model: Fitted Tree

Initial asset mix	Bonds	Real	Stocks
	0.37	0.30	0.33
Turnover	Bonds	Real	Stocks
	0.08	0.05	0.06
Objective Value	0.5213		
Mean Contribution Rate	0.0624		
Downside Risk	0.0346		

mean and covariance matrix. For the earlier stages with six or ten succeeding nodes we also matched skewness and kurtosis. More nodes are used in the earlier stages, while fewer nodes in the later stages introduce less approximation error into the optimal solution of the stochastic program. Table 7 displays the optimal solution of the stochastic programming model based upon the fitted event tree. The initial asset mix now consists of 33% stocks, 30% real estate and 37% bonds. Average turnover of the investment strategy has been reduced to 6%, but the average contribution rate increased to 6.24%. Note that the optimal solutions based on adjusted random sampling and tree fitting are still quite different.

It is still difficult to conclude much about the quality of the optimal solutions, as the ALM model was solved just three times using three particular event trees. Testing of stochastic programming models requires extensive out-of-sample simulations (Dupacova, Bertocchi and Morrigia 1998) and rolling horizon dynamic games (Golub et al. 1995). A rolling horizon simulation was used by Kouwenberg (1998) to investigate the issue of scenario generation comparing the three different methods for constructing event trees. The results of repeated experiments for the 5-period pension fund model based on 1-10-6-6-4-4 event trees confirm that random sampling leads to poor results. Adjusted random sampling significantly improves the performance of the model. Tree fitting is only slightly better for this particular application. With event trees based on the fitting method the stochastic programming approach was shown to outperform a benchmark static model and a benchmark model with an optimized decision rule.

Finally, for completeness we would like to mention some other promising methods for constructing event trees from the stochastic programming literature. Mulvey and Zenios (1994) and Zenios (1995) discusses simulation techniques to generate scenarios of returns for fixed-income portfolio models, based on a underlying fine-grained interest rate lattice. Lattice models for scenario generation that integrate interest rate and credit risk are given in Jobst and Zenios (2001). Carino, Myers and Ziemba (1998) introduce a clustering algorithm that reduces a large random sample of asset and liability returns to a small number of representative nodes, while preserving important statistical properties such as the mean and the standard deviation of the larger sample. This is an idea similar to the adjusted random sampling method we discussed above. Pflug and Swietanowski (1998) derive promising theoretical results for optimal scenario generation for multiperiod financial optimization. Shtilman and Zenios (1993) derive theoretical results for the optimal sampling from lattice models. Proper spacing of the returns seems crucial for reducing approximation errors. Further theoretical and empirical research in this area is important, as the event trees used as input are crucial for the effectiveness of the stochastic programming approach to ALM.

3.5 Options, Bonds and Arbitrage

We now turn to another requirement for scenarios used in ALM systems, the persuasive no-arbitrage condition. We review first the issues in the context of stochastic programming and then discuss the construction of arbitrage-free event trees. The generation of scenarios for ALM applications with multiple bonds or derivatives requires special attention to the no-arbitrage condition. The prices of bonds with different maturities are often driven by a small number of underlying factors such as the short term interest rate, a long term yield and the credit spread. Because of the close relationship between bond prices and interest rates, the prices movements of bonds of a similar type, but with different maturities, are often strongly associated. The price changes of derivative securities on a single underlying value, e.g. a stock index, are also often closely related.

An important concept for bond and option valuation is the construction of so called replicating strategies. For example, consider a portfolio consisting of one stock with price S and a European put option on this stock with price p, exercise price X and maturity T. We assume that the stock does not pay dividends and that the riskless interest rate for a maturity of T is equal to r. It is straightforward to verify that the value of the first portfolio at time T can be replicated by a strategy that invests $exp(-rT)X$ at the riskless interest rate and holds a European call on the stock with price c, exercise price X and maturity T. Because both portfolios eventually deliver the same payoff at time T, their prices should also be equal at all times t with $0 \leq t < T$. It follows that the option prices should satisfy the following relation, called put-call parity (see Hull 1989):

$$(34) \quad p + S = exp(-r(T-t))X + c.$$

If the prices of European call and put options with equal strike price do not satisfy the put-call parity relationship, then there is an *arbitrage opportunity* available. We could make a *riskless profit* by selling the overvalued option, while replicating its payoff with a portfolio containing the cheap option. One of the fundamental assumptions in the theory of finance is that arbitrage opportunities are not available: some investors will recognize these opportunities and their trading will immediately eliminate them. The absence of arbitrage and the concept of replicating strategies are at the heart of modern derivative pricing methods.

The absence of arbitrage opportunities is an important property for event trees of asset returns that are used as input for stochastic programming models as well. If there is an arbitrage opportunity in the event tree, then the optimal solution of the stochastic programming model will exploit it. An arbitrage strategy creates profits without taking risk, and hence it will increase the objective value of nearly any financial planning model. An important question is whether the arbitrage opportunities in the event tree are also available on the actual financial market. It is prudent for long term ALM applications to generate scenarios that do not allow for arbitrage. If arbitrage does arise in practice then arbitrageurs will exploit it, quickly aligning the real-world prices with our arbitrage-free event tree.

A potential problem for stochastic programming models in ALM are arbitrage opportunities in the event tree that are due to approximation errors. Klaassen (1997) was the first to address this issue. Arbitrage opportunities might arise because the underlying return distributions are sometimes approximated poorly with a small number of nodes in the event tree. If the application only involves broad asset classes such as a stock index, a bond index and real estate index, then arbitrage opportunities are unlikely to occur unless the errors in the event tree are very big. However, applications that involve options, multiple bonds or other interest rate derivative securities can be quite vulnerable to these problems. For example, the prices of European call and put options with equal strike price should satisfy put-call parity in each node of the event tree. If this relationship is violated because of a small approximation error, then the event tree contains an arbitrage opportunity and hence a source of spurious profits for the stochastic programming model.

3.5.1 Arbitrage-free event trees

Before we introduce scenario generation methods for arbitrage-free event trees we first introduce the no-arbitrage theorem from financial economics. Formally, arbitrage opportunities do not exist if there are no portfolios having a negative price that provide a non-negative payoff. Suppose that m primitive securities that are traded in a one period model. The initial asset prices are denoted by P_{0i} and the final asset prices and payoffs are, respectively, P^l_{1i} and F^l_{1i}, for $i = 1, 2, \ldots, m$ and $l \in \Omega$, where Ω is a scenario set denoting states of the economy, i.e., nodes on our event tree. Harrison and Kreps (1979) prove that the following conditions are both necessary and sufficient for the absence of arbitrage opportunities[1]:

Theorem 1 There are no arbitrage opportunities if and only if there exists a strictly positive probability measure $p^l > 0$, such that

$$(35)\quad \sum_{l\in\Omega} p^l \frac{P^l_{1i} + F^l_{1i}}{P_{0i}} = \sum_{l\in\Omega} p^l \frac{P^l_{11} + F^l_{11}}{P_{10}}, \quad \text{for all } i = 1, 2, \ldots, m.$$

If we take the base security $i = 1$ to be the one-period riskless bond, providing a continuously compounded return of r, then condition (35) reduces to:

$$(36)\quad P_{0i} \;=\; e^{-r} \sum_{l\in\Omega} p^l (P^l_{1i} + F^l_{1i}), \quad \text{for all } i = 2, 3, \ldots, m.$$

It is clear that given a set of m primitive assets, at least m states are needed in the set Ω to satisfy the no arbitrage condition if we would also like to

[1] We assume that all initial asset prices are strictly positive. Moreover, at least one primitive asset should have a set of strictly positive prices at time 1.

avoid linearly dependent asset returns. For the event tree of a stochastic programming model this means that we need at least m succeeding states in every node of the event tree in order to represent the conditional return distribution of the assets $i = 1, 2, \ldots, m$ from one period to the next. Otherwise we either introduce a money machine or the asset returns become linearly dependent.

In practice round-off errors might cause the system (36) to be infeasible, even if arbitrage profits are economically insignificant due to transaction costs. Naik (1995) provides a version of the no-arbitrage conditions that takes bid-ask spreads into account and is more easy to apply:

Theorem 2 Suppose that assets are bought at time zero at the ask price P^a_{0i} and sold at the bid price P^b_{0i}, with $P^a_{0i} > P^b_{0i}$. Security $i = 1$ is a riskless bond, with rate r. There are no arbitrage opportunities if and only if there exists a strictly positive probability measure $p^l > 0$, such that

$$(37) \quad P^b_{0i} \; \leq \; e^{-r} \sum_{l \in \Omega} p^l (P^l_{1i} + F^l_{1i}) \leq P^a_{0i}, \;\; \text{for all } i = 2, 3, \ldots, m.$$

Once an event tree with asset prices has been constructed, we can check for arbitrage opportunities by solving equation (36) or (37) in each node of the event tree. If we can find a set of strictly positive probabilities p^l for each node then the event tree does not contain arbitrage opportunities. In case the system is infeasible, it might be unwise to solve a stochastic programming model with this event tree as input, as this might lead to unbounded and biased solutions (Klaassen 1997).

It is possible to enforce the absence of arbitrage opportunities while constructing an event tree for asset returns. The first approach is to add the no-arbitrage condition (37) as a constraint to the tree fitting model (32)–(33). The tree fitting model has risk neutral probabilities p^l as additional decision variables, which are required to be strictly positive. If we can not find a feasible solution for the tree fitting problem then arbitrage opportunities are inevitable. In this case we could eliminate arbitrage by reducing the number of moment matching constraints in the tree fitting model or by increasing the number of nodes in the event tree. This entails a tradeoff between model accuracy and computational complexity.

A second approach is to start with a very fine-grained event tree of asset prices without arbitrage opportunities and then to reduce it to a smaller tree, while preserving the property of no-arbitrage. Klaassen (1998) studies a bond portfolio management problem, where the bond prices are calculated with a one-factor model for the term-structure of the interest rates. Bond prices are first generated with a binomial lattice for the one-factor interest rate model: the lattice is consistent with the initial bond prices on the market and contains no arbitrage opportunities. However, the lattice is not suited as input for a stochastic programming model, as it consists of many small time-steps in order to calculate prices of bonds and other interest rate dependent securities accurately. The problem arises as each possible path on the recombining interest rate lattice from the initial period to the planning horizon becomes a unique scenario in the non-recombining event tree for the stochastic programming model. Aggregation methods are essential to reduce the recombining lattice to a much smaller event tree with less trading dates, while preserving the property of no-arbitrage.

Klaassen (1998) proposes an aggregation method that starts with a full-blown non-recombining event tree, consisting of all possible interest rate scenarios. Recursively, a combination of nodes at a particular time period can be replaced by one agregated node, while preserving the no-arbitrage property. If a node has only one particular successor remaining at the next time, then the intermediate period can be eliminated. Another method for reducing a fine-grained lattice of security prices to a sparse event tree without arbitrage is discussed in Gondzio, Kouwenberg and Vorst (1999). They apply their method to an option hedging problem with two sources of uncertainty: the stock price and stochastic volatility. First a three-dimensional fine-grained grid of time versus stock price and volatilty is constructed to calculate option prices. Second, the points on the grid are partitioned into groups at a small number of trading dates, corresponding to the decision stages in the stochastic programming model. Each groups of points on the grid is represented by a single aggregated node in the event tree of the stochastic programming model. If the prices in each aggregated node are calculated as a conditional expectation under the risk neutral measure of the prices in the corresponding partition on the grid, then the aggregated event tree will not contain arbitrage opportunities.

Although the absence of arbitrage opportunities is important for financial stochastic programs with derivative securities, one should keep in mind that it is only a *minimal requirement* for the event tree. The fact that the stochastic program can not generate riskless profits from arbitrage opportunities does not imply that the event tree is also a good approximation of the underlying return process. We still have to take care that the conditional return distributions of the assets and the liabilities are represented properly in each node of the event tree. In order to avoid computational problems that arise if the tree becomes too big, one could reduce the number of stages of the stochastic program. In this way more nodes are available to describe the return distributions accurately. It is also important to include more nodes for the earlier stages, while larger errors in the later stages will have a small effect on the first-stage decisions which are the decisions implemented today by the decision makers. End effects of stochastic programming models for ALM applications are studied by Carino and Ziemba (1998) and Carino, Myers and Ziemba (1998).

3.6 Additional Methods for Scenario Generation

*[*** Stavros: this section is repeating information from subsection 3.3. Therefore I suggest to delete the entire section 3.5. ? ***]*

Scenario generation methods for stochastic programming models for ALM are not restricted to the systems described above. Neither is the framework in Figure 2 the only way to approach scenario generation methods. Indeed modern finance, with its interest in Value-at-Risk as the standard for risk measurement (Jorion 1996), provides a variety of methods for scenario generation. Any methodology used for estimating Value-at-Risk can be used to build event trees for stochastic programs. One would need to extend the simulations to handle multiple time-periods and the conditional dependence of one period on its predecessors. Jamshidian and Zhu (1997) and Jobst and Zenios (2001) discuss two approaches in this direction, the former dealing with market and currency risk and the later with market and credit risk.

The standard Value-at-Risk approaches are usually backward looking, relying on historical data for calibrating the simulations. Dembo et al. (2000) propose a forward looking simulation framework that is both multi-step and dynamic, and integrates disparate sources and measures of risk and reward. This framework (called by the authors Mark-to-Future) is also suitable for stochastic programming ALM applications.

Finally, we mention a simple approach that has been adopted by several authors and appears to work well in practice. Scenarios of asset returns can be generated by bootstrapping a set of monthly records from historical data. Each sample is a scenario of returns of the assets. For each date in the sample we read the returns of all assets during the month prior to that date, and this is one scenario of returns. With this approach the correlations among asset classes are preserved (see for example Jorion 1996). The Prometeia model described by Consiglio, Cocco and Zenios in this handbook use bootstrapping for scenario generation.

4 Comparison of Stochastic Programming with Other Methods

The literature on models for ALM is vast and dates back to the seminal contribution of Markowitz (1952). But it was not until the 1980s that the use of formal mathematical models to support financial decision making rose to wide-spread prominence in practice (Zenios 1993). Globalization and innovations in the financial markets are the driving force behind this development that continues unabated to this date, aided by advances in computing technology and the availability of software. Four alternative modelling approaches have emerged as suitable frameworks for representing ALM problems (Ziemba and Mulvey 1998). These approaches are briefly discussed in this section and compared with stochastic programming. See also the chapters by Markowitz and by Brennan and Xia in this volume for alternative ALM methodologies.

4.1 Mean-Variance Models and Downside Risk

The mean-variance framework of Markowitz (1952) is widely considered as the starting point for modern research about optimal investment. In the mean-variance framework the optimal portfolio of an investor is derived by minimizing the variance of the portfolio return, subject to a given mean return. Markowitz demonstrated that investors can reduce risk by forming well-diversified portfolios of individual stocks supporting the popular advise "Don't put all your eggs in one basket". Moreover, Markowitz (1952) makes clear that we have to pay the price of increasing risk (variance) in order to obtain a higher expected return.

The original version of the mean-variance model ignores the liability side of the investor's balance sheet. Sharpe and Tint (1990) propose an extended mean-variance model for the surplus, defined as the asset value minus the liabilities. In a surplus management model the investor minimizes the variance of the surplus return, for a given level of mean surplus return. The main result is that the optimal investment policy depends on the covariance of the asset returns with the liability return. The investor is willing to accept a lower expected return on assets that provide a higher covariance with the liability return. In order to quantify this effect, Sharpe and Tint (1990) introduce the *liability hedging credit* of an asset.

Liability hedging credits are at the core of investment strategies for asset liability management. If the investor is concerned about the net value of his balance sheet (i.e., the surplus), then he has to reckon with the correlation between asset and liability returns. Although this basic insight provided by the surplus management model of Sharpe and Tint is important, the model is of limited use in practice. There are two major drawbacks associated with mean-variance models in the context of asset and liability management. Variance is not always a good risk measure for investors, and a single-period model might be inappropriate for multi-period investment problems with long horizons. We first turn our attention to alternative risk measures and then focus on the literature about multi-period investment models.

Intuitively it seems rather odd that variance as a risk measure penalizes positive returns and negative returns equivalently. Of course, it is only important to make a distinction between negative and positive returns if the distribution of the portfolio returns is asymmetric. However, most stock return distributions are skewed (Fama 1965). Moreover, if we add derivatives such as call and put options to the investment opportunity set then the portfolio return distribution might also become asymmetric. In order to deal with these problems, downside-risk measures have been introduced as a substitute for variance (Bawa and Lindenberg 1977 and Fishburn 1977).

Downside-risk measures only penalize returns below a given threshold level, specified by the investor. Popular measures include: shortfall probability, expected shortfall, semivariance, one-sided mean absolute deviation (see Bawa and Lindenberg 1977, Worzel, Vassiadou-Zeniou and Zenios 1994 and Rockafellar and Uryasev 2000). In the normative ALM literature the downside-risk concept is highly successful and seems to have replaced variance as a risk measure (Harlow 1991, Sortino and van der Meer 1991, Boender 1997, Artzner et al. 1999). Investors that try to meet their liabilities usually apply a threshold of 1 for the funding ratio, i.e., the ratio of assets to liabilities. A mean-semivariance model for the surplus or the funding ratio can be solved very efficiently when formulated in an equivalent mean-absolute deviation model. This has been demonstrated by Konno and Yamazaki (1991) and Konno and Kobayashi (1997).

4.2 Discrete-Time Multi-Period Models

Apart from the criticism of the risk measure in the mean-variance model, a second concern is that the model only considers a single period, without opportunities to change the investment strategy inter-temporally. Most asset liability management problems cover a long period of time until the planning horizon, with multiple opportunities to change the investment portfolio. For example, the planning horizon for most insurance products extends beyond a decade, for pension funds it is more than 30 years, and for social security plans it may go up to 50. It is clearly inappropriate to model several decades into the future in one single period without allowing for trading to adjust the investment portfolio.

Several authors have studied models that overcome the single-period restriction of the mean-variance model. Mossin (1968), Samuelson (1969) and Hakansson (1970) study the multi-period consumption-investment problem for investors maximizing the expectation of a power utility function of wealth at the end of their horizon, under the assumption of intertemporally independent distributed asset returns. A power utility function has the property of constant relative risk aversion and as a result the portfolio weights are equal in each period, regardless of the individual's age and wealth. The investment policies are called myopic as the investor behaves identically to a single-period investor. Mossin (1968) showed that a sequence of myopic models are optimal for a multi-period investor under the assumptions of intertemporal independence of returns, no transaction costs and no cash infusion or withdrawals. Hakansson (1969,1971) generalizes Mossin's results for the logarithmic utility functions to add stochastic wage income, serially correlated asset returns and uncertainty about the individual's lifetime to the basic setup. Mulvey and Zenios (1994) discuss how the assumptions on which this line of literature is based fail in the modern capital markets, especially when dealing with fixed-income and derivative securities.

A discrete-time retirement saving model with labor income as a benchmark for investments and contribution payments was introduced by Berkelaar and

Kouwenberg (1999). The asset liability management problem is considered from the point of view of a plan sponsor who has established an investment fund for the future retirement of one employee. The goal of the plan sponsor is to minimize his contribution payments, while trying to maximize the value of the investment fund at the retirement date, relative to labor income of the employee. Labor income is used as a benchmark for investments in order to ensure that the employee can continue his consumption habits after retirement. Due to the additional complexity of the retirement saving problem and the use of a penalty function that considers downside risk, Berkelaar and Kouwenberg (1999) can not derive closed-form solutions. Instead, they implement the dynamic programming algorithm numerically, on a three-dimensional grid for the state variables: time, wage growth and asset value. The goal of the paper is to investigate the effects of downside-risk measurement and contribution payments on the optimal investment strategy. The main conclusion is that both of these features of the model lead to increased risk taking at low levels of wealth (gambling effects, see also Dert and Oldenkamp 2000).

Dynamic programming for ALM can lead to interesting insights, as the optimal policies are derived in feed-back form. However, the curse of dimensionality limits the type of ALM models that can be solved. Models with more than three or four state variables are bound to run into serious computational problems, and it is very difficult to handle transaction costs. An alternative approach is to specify a *decision rule* for changing the investment strategy in advance, and optimize the parameters of the given rule. For instance, Maranas et al. (1997) use this approach for a multi-period fixed mix model, which optimizes a set of fixed asset weights to maximize the objective function of a multi-period ALM model. The ALM simulation system of Boender (1997) optimizes the parameters of decision rules that update the strategic asset mix and the contribution rate as a function of the current funding ratio.

Models with decision rules, such as fixed mix, have several advantages. First, they can handle transaction costs and operational, regulatory or corporate restrictions on the investment policy. Moreover the optimized rule can be easily interpreted and understood by decision makers. A major problem is that the model is non-unimodal. Multiple local solutions may exist, and global optimization algorithms have to be applied. Recent progress on global optimization nonwithstanding these algorithms can often handle only a small number of decision variables (i.e., coefficients of the decision rules). Finally, we are never sure that a given decision rule is actually optimal for the problem at hand. For example, we do not always know in advance that a fixed mix rule is optimal for a particular ALM problem. It might as well be that another dynamic investment rule is more efficient: for example, we could adjust the asset mix as a function of the ratio of assets to liabilities. If we apply the latter rule, there is still an imense number of different functional forms that we could choose, as the optimal relationship between the asset weights and the funding ratio can be non-linear. A disadvantage is therefore that we might have to try a large number of different specifications for the decision rule, before we are confident that we have found a rule with relatively good performance.

4.3 Continuous-Time Models

Continuous-time models play an important role in modern finance (Merton 1990). As a companion paper of Samuelson (1969), Merton (1969) formulates the consumption-investment problem in a continuous-time framework, where the timestep between consecutive trading dates decreases to zero in the limit. The asset prices are assumed to follow geometric Brownian motions, which corresponds to a log-normal return distribution. For the class of power utility functions he concludes that myopic investment policies are optimal, analogous to the findings of Samuelson (1969). In a more general consumption-investment setup Merton (1971) confirms the findings of Hakansson (1969,1971). An interesting result of Merton (1969,1971) is *fund-separation*. The optimal portfolio of every investor can be separated into a small number of mutual funds: the riskless asset, the growth-optimal portfolio of risky assets, and a hedge-portfolio for each external source of risk that affects the asset returns or the utility of the investor.

The models of Merton are of limited practical value for institutional asset and liability management. The assumptions about the utility function and asset prices are restrictive. More importantly in the context of ALM for large institutions this model ignores transaction costs and does not incorporate trading restrictions that may be imposed by regulators or dictated by corporate policy. The financial economics literature has lately paid more attention to previously ignored "details" that are very relevant for practitioners, resorting to numerical techniques if necessary. For example, Brennan, Schwartz and Lagnado (1997) numerically investigate the impact of return predictability on optimal portfolio choice in a continuous-time investment model based on Merton (1969). The optimal portfolio weights reported by Brennan, Schwartz and Lagnado (1997) tend to fluctuate drastically through time, resembling "yoyo-strategies", due to the absence of transaction costs in the model and the lack of uncertainty about the model parameters.

Balduzzi and Lynch (1999, 2000) show that transaction costs can indeed stabilize the optimal policy of an optimal control model with return predictability. In practice transaction costs might not be the only concern, but also uncertainty about the actual value of model parameters such as the mean asset return. Brennan (1998), Barberis (2000) and Xia (2000) study continuous-time investment models under parameter uncertainty, with dynamic learning about parameter values (Brennan 1998), with return predictability (Barberis 2000) and with both learning and predictability (Xia 2000). Another recent development is that optimal portfolio and consumption problems with return predictability can be solved in closed-form, as demonstrated by Kim and Omberg (1996) and Liu (1999). See also the chapter by Brennan and Xia in this volume.

An advantage of the continuous-time framework is that optimal decision rules can be derived for some basic models: Cairns and Parker (1997) and Rudolf and Ziemba (1998) derive optimal decision rules in closed form for small ALM problems. Moreover, the impact of transaction costs, trading limits, return predictability, parameter uncertainty and market incompleteness can be analyzed quite accurately in models that focus on one or two of these issues in isolation. However, general models that incorporate all of these issues simultaneously have not been solved yet. Similarly, practical constraints reflecting regulatory restrictions, operational requirements or corporate policy have not been incorporated. Moreover, an attempt to solve such a general model is very likely to run into computational problems due to the curse of dimensionality.

4.4 Stochastic Programming

Continuous-time models and discrete-time models solved with dynamic programming and optimal control can provide good qualitative insights about fundamental issues in investments and ALM, as the optimal decision rules are in feed-back form. However, their practical use as a tool for decision making is limited by the many simplifying assumptions that are needed to derive the solutions in a reasonable amount of time. The stochastic programming approach for ALM discussed in this chapter can be considered as a practical multi-period extension of the normative investment approach of Markowitz (1952). The advantage of stochastic programming models for multi-period investment and ALM problems is that important practical issues such as transaction costs, multiple state variables, market incompleteness, taxes and trading limits, regulatory restrictions and corporate policy requirements can be handled *simultaneously* within the framework.

Of course this flexibility comes at a price and stochastic programming also has a drawback. The computational effort explodes as the number of decision stages in a multi-stage stochastic programming model increases. While implementing a stochastic programming model for ALM, we are therefore often forced to make a trade off between the number of decision stages in the model and the number of nodes in the event tree that are used to approximate the underlying returns distributions. While setting up stochastic programming models it is important to keep in mind that a normative model does not necessarily have to include every possible decision moment up to the planning horizon. Capturing the first few opportunities accurately can be good enough to make an informed decision right now. End effects created by limiting the number of stages can be mitigated using the techniques discussed in Carino and Ziemba (1998) and Carino, Myers and Ziemba (1998).

It is not an exaggeration to claim that stochastic programming can deal simultaneously with all important aspects of an ALM system. However, even if the model would be solvable, too many details would confuse instead of support the decision maker. Like the alternative methodologies discussed in this section, stochastic programming applications to ALM have a strong element of art. Stochastic programming seems to enjoy several advantages over the alternatives but it is not without shortcomings.

5 Applications of Stochastic Programming to ALM

In recent years the number of publications about stochastic programming for asset liability management has risen drastically, probably inspired by the radical increase of efficiency and accessibility of computer systems. Ziemba and Mulvey (1998) categorize the models in three generations: (*i*) model origins, that deal with the early mathematical formulations, (*ii*) early models, that deal with real-world applications but developed and tested in a limited setting, mostly by academic researchers, and (*iii*) modern models, that deal with a variety of institutional problems, developed as large-scale applications and tested extensively, usually in collaboration with institutional asset and liability managers. From the early models we mention the bond portfolio management model of Bradley and Crane (1971), the bank ALM model of Kusy and Ziemba (1986), the fixed income model of Zenios (1991), the asset allocation model of Mulvey and Vladimirou (1992) and the stochastic dedication model of Hiller and Eckstein (1993).

Under modern models Ziemba and Mulvey list around forty references. We mention a sample that covers a broad range of applications, selecting publications where the commercial component of the model was substantial. The insurance ALM model of Carino and co-authors (1994, 1998) has been used extensively by the Frank Rusell company in consulting ALM managers in insurance and pension fund. Their work with the The Yasuda Fire and Marine Insurance Company (Japan) was a finalist at the Franz Edelman Competition for Management Science Achievements. Similar acclaim was achieved by the Towers Perrin–Tillinghast model of Mulvey, Gould and Morgan (2000). Stochastic programming models for Dutch pension funds were developed by Dert (1995), a general ALM model for insurers by Consigli and Dempster (1998) and an application to the Norwegian insurance industry by Høyland and Wallace (this volume). Models for money management with mortgage backed securities were developed by Golub et al. (1995) and for insurance products by Nielsen and Zenios (1996). A multiperiod model, but without portfolio rebalancing decisions, for insurance products with minimum guarantee is discussed in the chapter by Consiglio, Cocco and Zenios in this volume.

The relevance of a normative approach can only be judged on the basis of actual performance. Carino, Myers and Ziemba (1998) report good performance of the Frank Rusell ALM model applied to Yasuda Insurance, with yearly savings of up to $79 million compared to a simple constant mix strategy. Mulvey, Gould and Morgan (2000) report an estimate of $450 to $1,000 million savings in opportunity costs using the Towers Perrin–Tillinghast model to plan the US West pension plan. Golub et al. (1995), Kouwenberg (1998) and Fleten, Høyland and Wallace (1998) apply simulation in order to test the performance of stochastic programming models, relative to simple one-period models. The results indicate that dynamic stochastic programming models can outperform simple mean-variance models and multi-period models with fixed mix decision rules. Significant savings on transaction costs achieved with stochastic programming models were also demonstrated by Mulvey (1993). The effectiveness of stochastic programming models for tracking broad market indices has been demonstrated by Consiglio and Zenios (2001), Zenios et al. (1998) and Worzel, Vassiadou-Zeniou and Zenios (1994).

Golub et al. (1995) apply both out-of-sample simulations and dynamic, rolling horizon games, to compare the performance of stochastic programming models against single-period myopic models. This is, to the best of or knowledge, the first study to compare these two classes of models using out-of-sample simulations and dynamic games. In particular they tested a mean-absolute deviation model against stochastic programming for money management problems with mortgage-backed securities. The objective of the money manager was to achieve a target return in excess of the 3-year Government benchmark by investing in a diversified portfolio of mortgage backed securities and government bonds. Uncertainty in interest rates and mortgage prepayments are the main risk factors of this setting. In both out-of-sample simulations and dynamic games a two-stage stochastic programming model consistently outperformed the single period model. The stochastic programming model was also found to be more robust than mean-absolute deviation with respect to out-of-sample changes in volatility. Golub et al. also tested the performance of the model against the popular fixed-income portfolio immunization technique (Reddington, 1952, Christensen and Fabozzi, 1987, Zenios, 1993). Their findings here confirmed the conclusions of Mulvey and Zenios (1994) that modern fixed income securities are best managed by capturing correlations—in a single-period setting in this study—instead of using duration matching in an immunization framework.

The work of Golub et al. was extended by Zenios et al. (1998) to multistage models. Their empirical investigations were once more carried out using out-of-sample simulations and rolling horizon dynamic games, but this time comparing single period with two-stage and three-stage models. The three-stage model outperformed the two-stage model for 12 out of the 15 repetitions of the dynamic game, over a three year period. Summary statistics are reported in Table 8.

This model was also tested, *ex poste* in tracking the Salmon Brothers index of mortgage-backed securities in the US. Backtesting was carried out over the three year period January 1989 to December 1991, in monthly steps. During this period the index realized an annualized return of 14.05%, the

Table 8: Zenios et al. (1998): Dynamic games with stochastic programming.

Statistic	Single-period model	Two-stage stoch. prog.
Min. return	8.11	7.74
Max. return	11.61	9.78
Mean return	8.60	8.83
Std. dev.	0.86	0.59

[1] The table shows the results of dynamic rolling horizon games for money management with mortgage backed securities. The target is to exceed the benchmark return of the 3-year Treasury and achieve highest expected return during the holding period. A three-year horizon is considered and the experiments were repeated 15 times with a three-year rolling horizon and portfolio rebalancing every six months.

single-period period model (mean-absolute deviation) 14.18% and the two-stage stochastic program 15.10%. (Returns of the model portfolios account for transaction costs.) Figure 4 illustrates the performance of both models.

Gondzio, Kouwenberg and Vorst (1999) apply a stochastic programming model to an option hedging problem in an incomplete market with stochastic volatility and transaction costs. They test the performance of the stochastic programming model and other hedging strategies with simulations. The test problem assumes that an investor has sold a call option on the stock index with a maturity of one year, for a price of \$8.41. This liability is unique: it can not be traded on an organized market. The aim of the investor is to minimize the expected negative hedging errors, which occur if the value of the hedging portfolio drops below the value of the liability. In order to hedge his liability the investor can trade stocks, borrow and lend money, and trade liquid short-term call and put options on the index with a maturity of 3 months. All trades lead to transactions costs due to differences in bid and ask prices: 0.25% for trading the index (futures), 2.5% for trading short-term options and there is a spread of 0.5% between borrowing and lending.

The stochastic programming model for the hedging problem consists of three trading dates, covering the first three weeks of the investor's problem. The model does not necessarily have to incorporate every trading date up to the maturity of the liability, as it can be solved again each period (assuming end effects are modelled properly, see also Carino e.a. 1994, 1998). Gondzio, Kouwenberg and Vorst (1999) compare the optimal strategy of the stochastic programming model to a delta hedging strategy and a delta-vega hedging

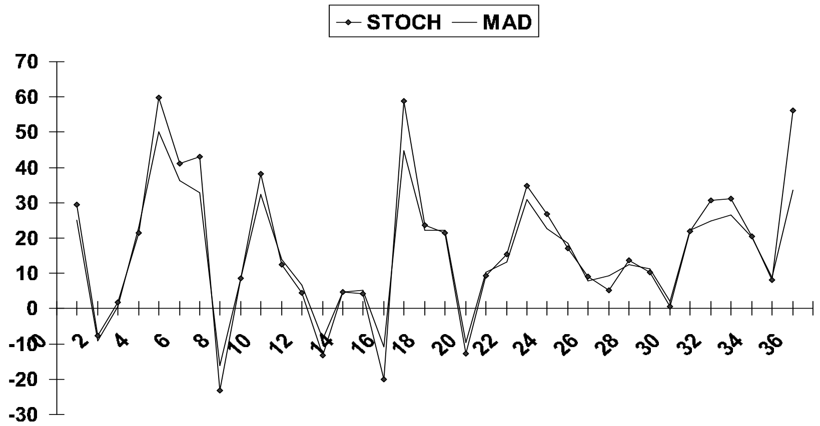


Figure 4: Tracking error of a singe-period and a stochastic programming model in tracking the Salomon Brothers index of mortgage backed securities during the 36-month period January 1989 to December 1991.

strategy, which are decision rules that are traditionally applied (see Hull 1989). The delta hedging strategy eliminates exposure to small movements of the stock price, but it ignores the effects of stochastic volatility. As the value of the liability is very sensitive to changes in the volatility of the stock index, this strategy can lead to large hedging errors. The delta-vega hedging strategy eliminates this volatility exposure by taking a position in exchange-traded short-term options. Note that both strategies require frequent adjustments to keep the hedge up to date and this could lead to considerable transaction costs.

Gondzio, Kouwenberg and Vorst (1999) sample a total of 100,000 simulation paths for the stock index and its volatilty to compare these two hedging strategies with the optimal strategy of a stochastic programming model. Panel A of Table 9 shows the simulation results of the delta hedging strategy, panel B shows the performance of the delta-vega hedging strategy and panel C presents the stochastic programming approach. The results clearly show that the delta hedging strategy (panel A) performs very badly: its average negative hedging error is approximately 100 times worse than the hedging error of the stochastic programming model. This poor performance comes as no surprise, given that a pure delta hedging strategy ignores volatility movements. The delta-vega hedging strategy in panel B performs much better, because it leads to lower hedging errors and it entails less trading. The stochastic programming approach in panel C improves the results even further by reducing transaction costs. The optimal strategy of the stochastic program is on average quite close to a delta-vega hedging strategy, but with additional slack in order to avoid needless trading costs. In practice portfolio managers and traders might aditionally face limits on the amounts they can borrow and sell short. Gondzio, Kouwenberg and Vorst (1999) show that these restrictions can be easily incorporated in a stochastic programming model, which is another advantage of the approach.

6 Solution Methods and Computations

Stochastic programming models grow in size very quickly with the number of stages and the number of scenarios at each stage. Some of the problems solved in the literature have equivalent deterministic formulations with hundreds of thousands of variables and constraints. For instance, the model used in previous sections to test different event trees has an equivalent linear programming formulation with 24614 constraints, 32100 variables and 96586 (0.012%) non-zeros in the constraint matrix. Due to the exponential growth of the number of nodes in the event tree, the number of variables and constraints is huge. The deterministic equivalent linear program of the stochastic programming model is solved using an interior point algorithm that exploits the sparse block-angular structure (Kouwenberg 1998). The fixed income models and the asset allocation models can also be represented as network flow problems and can be solved using special purpose network optimization algorithms (Mulvey and Vladimirou, 1992, Nielsen and Zenios, 1996). Table 10 summarizes the characteristics of some of the models we have cited.

Problem of this size need some specialized solution algorithms. The state-of-the-art in optimization software allows the solution of large-scale problems although the computer resources required may be substantial. In general one should expect several hours of computer time on a dedicated high-performance workstation. Almost real-time solutions have been reported in the literature for these problems, but such performance was invariably

Table 9: Gondzio, Kouwenberg and Vorst (1999): Simulation Results

A. Delta Hedging: Hedging Error 1.609%

Date	T-costs	Turn-over	Delta Gap	Vega Gap
1	0.1418	688 %	0	100 %
2	0.0154	83 %	0	100 %
3	0.0153	84 %	0	100 %

B. Delta-Vega Hedging: Hedging Error 0.075%

Date	T-costs	Turn-over	Delta Gap	Vega Gap
1	0.1284	292 %	0	0
2	0.0125	27 %	0	0
3	0.0273	48 %	0	0

C. Stochastic Program: Hedging Error 0.017%

Date	T-costs	Turn-over	Delta Gap	Vega Gap
1	0.1354	247 %	0.24 %	0.70 %
2	0.0070	24 %	0.87 %	5.59 %
3	0.0034	12 %	1.51 %	10.35 %

[1] The table shows the results of simulations with different hedging strategies. Panel A represents delta-hedging with stock trading only. Panel B represents delta-vega-hedging with a short-term at-the-money option and the stock. Panel C represents the optimal trading strategy of the SOH model, involving the five available traded short-term options and the stock (see Gondzio, Kouwenberg and Vorst 1999 for details). The hedging error denotes the average negative hedging error after three weerks, denoted as a percentage of the intial value of the liability ($8.41). Next, for the three trading date are shown: the average transaction costs in cents, the average portfolio turn-over, the average absolute delta gap and the average absolute vega gap. The portfolio turn-over is defined as the sum of the absolute changes of the money invested in each asset divided by two times the value of the portfolio. The absolute delta (vega) gap is the absolute difference between the delta (vega) of the hedge portfolio and the delta (vega) of the 1-year call option.

Application	Stages	Scenarios	Deterministic equivalent	Source
Fixed income	3	16,392	492,036 × 1,377,676	Nielsen and Zenios (1996)
Fixed income	5	6,461	51,082 × 154,178	”
Insurance	5	1,024	123,425 × 174,480	Carino et al. (1994,1998)
Insurance	10	2688	352,114 × 671,961	Consigli and Demptser (1998)
Pension Funds	5	5760	24,614 × 32,100	Kouwenberg (1998)
Pension Funds	6	4,826,809	12,469,250 × 24,938,502	Gondzio and Kouwenberg (1999)

Table 10: Large scale ALM applications of stochastic programming.

made possible with the use of parallel computers.

The special block-angular structure of the constraint matrix of stochastic programs has prompted the development of specialized algorithms. Modern implementations of the simplex method, such as IBM's OSL or CPLEX by Ilog, incorporate many theoretical results of research on this problem, making commercially available two excellent versions of this algorithm for the solution of stochastic programming problems. Similarly, interior point methods have been specialized for block angular structures and both OSL and CPLEX implement this feature as well.

Finally we mention the development of special-purpose decomposition algorithms for breaking up the deterministic equivalent formulation into smaller problems. These can be solved either serially or in parallel. In any event they are much smaller than the original problem hence solution times are substantially improved. OSL supports some decomposition methods. However, most software implementations of decomposition methods are supported by academic researchers. In general such systems are very efficient and quite robust, but they are not of industrial quality.

Work on the solution of stochastic programs has also focused on the intelligent sampling and pruning of the event tree. Clearly not all events on an event tree will have an effect on the optimal solution. It is important to sample only those events that have the most impact on the solution. Importance sampling (Dantzig and Infanger 1991) and EVPI (expected value of perfect information, Dempster and Gassmann 1991) have appeared as promising avenues for restricting the tree size, and structuring problems of moderate size. For a discussion of solution techniques and an extensive list of references see Censor and Zenios (1997, Ch. 13).

7 Summary and Open Issues

There are still some interesting open issues in the area of stochastic programming and asset liability management.

Perhaps the most important issue is in expanding the applicability of stochastic programming to address enterprise-wide risk management problems. The first step in this direction is to shaping firm-wide risk analysis according to a portfolio approach. What risks arise in the operation of the business? What risks are connected to the core activities of the firm (core exposures) and what to the facilitating activities (peripheral exposures)? How do these separate risks interact on the firm-wide level and how can their aggregate influence on the business performance be analyzed? What is the contribution of these risks to the firm's (diversified) overall risk profile? Once these questions are addressed—and these are mostly pricing and simulation questions—we need to design firm-wide risk management recognizing portfolio diversification benefits and natural hedges. Stochastic programming is the ideal tool for synthesizing the firm-wide risk analysis into firm-wide risk management. Finally post-optimality analysis of the models will allow us to develop of a firm-wide performance measurement system and allowing for the decomposition of risk measures and performance measures into components attributable to the various underlying risk factors.

With regard to the generation of scenarios, an important issue is how to measure the approximation error of the returns in the event tree compared to the true underlying distribution. Once appropriate measures have been identified, one could try to develop methods for constructing event trees that minimize the approximation error (assuming the size of the event tree is fixed). A promising first step in this direction is made by Pflug and Swietanowski (1998). Postoptimality analysis (Dupacova et al., 1998) also holds great potential in this respect.

With regard to the computational side of stochastic programming, there seems to be a need for flexible and efficient model generation tools. Specialized optimization algorithms and the ever increasing computational power of computers make it feasible to solve large scale multi-stage ALM models with millions of variables and constraints on desktop computers nowadays. However, most commercial mathematical modelling languages are not capable of generating the data of these huge problems efficiently. Morever, if the modelling language does not exploit the special structure of the stochastic program, it can easily run in to memory problems that could be avoided. Model generation seems to have become the bottleneck that limits the size of multi-stage stochastic programming models applied to ALM.

Finally, with regard to designing models for actual ALM problems, more research into end-effects might be helpful. Many ALM problems in practice are long term in nature and have much more decision moments than can be captured in a single multi-stage stochastic programming model. If we only consider the first few decision moments in a stochastic programming model, it is very important to choose an objective that is consistent with the long run goals of the company and makes sure that business can also continue as usual after the planning horizon of the model. Carino et al. (1994) and Carino, Myers and Ziemba (1998) make some promising contributions in this direction.

A Basics of Probability Spaces

We give some basic definitions needed in this chapter. Additional background material can be found in Billingsley (1995) and, with emphasis on stochastic programming in Wets (1989). Boldface Greek characters denote random vectors.

Let Ω be an arbitrary space or set of points. A *σ-field* for Ω is a family Σ of subsets of Ω such that Ω itself, the complement with respect to Ω of any set in Σ, and any union of countably many sets in Σ are all in Σ. The members of Σ are called *measurable* sets or *events* . The set Ω with the σ-field Σ is called a *measurable space* and is denoted by (Ω, Σ).

Let Ω be a (linear) vector space and Σ a σ-field. A *probability measure* P on (Ω, Σ) is a real-valued function defined over the family Σ, which satisfies the following conditions: (i) $0 \leq P(A) \leq 1$ for $A \in \Sigma$; (ii) $P(\emptyset) = 0$ and $P(\Omega) = 1$; and (iii) if $\{A_k\}_{k=1}^{\infty}$ is a sequence of disjoint sets $A_k \in \Sigma$ and if $\cup_{k=1}^{\infty} A_k \in \Sigma$ then $P(\cup_{k=1}^{\infty} A_k) = \sum_{k=1}^{\infty} P(A_k)$. The triplet (Ω, Σ, P) is called a *probability space*. The *support* of (Ω, Σ, P) is the smallest subset of Ω with probability 1. If the support is a countable set then the probability measure is said to be discrete. The term *scenario* is used for the elements of Ω of a probability space with a discrete distribution.

A proposition is said to hold *almost surely* if it holds on a subset $A \subseteq \Omega$ with $P(A) = 1$. The *expected value* of a random variable $\mathcal{Q}$ on (Ω, Σ, P) is the Stieltjes integral of $\mathcal{Q}$ with respect to the measure P:

$$E[\mathcal{Q}] \doteq \int \mathcal{Q} dP = \int_{\Omega} \mathcal{Q}(\boldsymbol{\omega}) dP(\boldsymbol{\omega}).$$

Let (Ω, Σ, P) be a probability space and suppose that $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_K$ is a finite partition of the set Ω. From this partition we form a σ-field $\mathcal{A}$ which is a subfield of Σ. The *conditional expectation* of the random variable $\mathcal{Q}(\boldsymbol{\omega})$ on (Ω, Σ, P) given $\mathcal{A}$ at $\boldsymbol{\omega}$ is denoted by $E[\mathcal{Q} \mid \mathcal{A}]$ and defined as

$$E[\mathcal{Q} \mid \mathcal{A}] \doteq \frac{1}{P(\mathcal{A}_i)} \int_{\mathcal{A}_i} Q(\boldsymbol{\omega}) dP(\boldsymbol{\omega})$$

for $\boldsymbol{\omega} \in A_i$, assuming that $P(\mathcal{A}_i) > 0$.

References

[1] P. Artzner, F. Delbaen, J-M. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, 9:203–228, 1999.

[2] M.R. Asay, P.J. Bouyoucos, and A.M. Marciano. An economic approach to valuation of single premium deferred annuities. In S.A. Zenios, editor, *Financial Optimization*, pages 100–135. Cambridge University Press, 1993.

[3] P. Balduzzi and A. Lynch. Transaction costs and predictability: Some utility cost calculations. *Journal of Financial Economics*, 52:47–78, 1999.

[4] P. Balduzzi and A. Lynch. Predictability and transaction costs: The impact on rebalancing rules and behavior. *Journal of Finance*, 55:2285–2309, 2000.

[5] N. Barberis. Investing for the long run when returns are predictable. *Journal of Finance*, 55:225–264, 2000.

[6] V.S. Bawa and E.B. Lindenberg. Capital market equilibrium in a mean-lower partial moment framework. *Journal of Financial Economics*, 5:189–200, 1977.

[7] E.M.L. Beale. On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society*, 17:173–184, 1955.

[8] A. Berkelaar and R. Kouwenberg. Retirement saving with contribution payments and labor income as a benchmark for investments. Econometric institute report ei-9946/a, Erasmus University Rotterdam, Netherlands, 1999.

[9] P. Billingsley. *Probability and Measure*. John Wiley and Sons, New York, USA, 1995.

[10] J.R. Birge and F. Louveaux. *Introduction to Stochastic Programming.* Springer, Heidelberg, 1997.

[11] F. Black, E. Derman, and W. Toy. A one-factor model of interest rates and its application to treasury bond options. *Financial Analysts Journal*, pages 33–39, Jan./Feb. 1990.

[12] G.C.E. Boender. A hybrid simulation/optimisation scenario model for asset/liability management. *European Journal of Operational Research*, 99:126–135, 1997.

[13] S.P. Bradley and D.B. Crane. A dynamic model for bond portfolio management. *Management Science*, 19:139–151, Oct. 1972.

[14] M.J. Brennan. The role of learning in dynamic portfolio decisions. *European Economic Review*, 1:295–306, 1998.

[15] M.J. Brennan and E.S. Schwartz. An equilibrium model of bond pricing and a test of the market efficiency. *Journal of Financial and Quantitative Analysis*, XVII:301–329, Sept. 1982.

[16] M.J. Brennan, E.S. Schwartz, and R. Lagnado. Strategic asset allocation. *Journal of Economic Dynamics and Control*, 21:1377–1403, 1997.

[17] L.D. Cagan, N.S. Carriero, and S.A. Zenios. A computer network approach to pricing mortgage-backed securities. *Financial Analysts Journal*, pages 55–62, March/April 1993.

[18] A.J.G. Cairns and G. Parker. Stochastic pension fund modelling. *Insurance: Mathematics and Economics*, 21:43–79, 1997.

[19] D.R. Carino, T. Kent, D.H. Myers, C. Stacy, M. Sylvanus, A.L. Turner, K. Watanabe, and W.T. Ziemba. The Russell-Yasuda Kasai model: An asset/liability model for a Japanese insurance company using multistage stochastic programming. *Interfaces*, 24(1):29–49, 1994.

[20] D.R. Carino, D.H. Myers, and W.T. Ziemba. Concepts, technical issues, and uses of the Russell-Yasuda Kasai financial planning model. *Operations Research*, 46:450–462, 1998.

[21] D.R. Carino and W.T. Ziemba. Formulation of the Russell-Yasuda Kasai financial planning model. *Operations Research*, 46:433–449, 1998.

[22] Y. Censor and S.A. Zenios. *Parallel Optimization: Theory, Algorithms, and Applications*. Series on Numerical Mathematics and Scientific Computation. Oxford University Press, New York, NY, 1997.

[23] V.K. Chopra and W.T. Ziemba. The effect of errors in means, variances and covariances on optimal portfolio choice. *Journal of Portfolio Management*, pages 6–11, Winter 1993.

[24] P.E. Christensen and F.J. Fabozzi. Bond immunization: An asset liability optimization strategy. In F.J. Fabozzi and I.M. Pollack, editors, *The Handbook of Fixed Income Securities*. Dow Jones Irwin, 1987.

[25] G. Consigli and M.A.H. Dempster. The CALM stochastic programming model for dynamic asset and liability management. In W.T. Ziemba and J.M. Mulvey, editors, *Worldwide Asset and Liability Modeling*, pages 464–500. Cambridge University Press, 1998.

[26] A. Consiglio, F. Cocco, and S.A. Zenios. The value of integrative risk management for insurance products with guarantees. *Journal of Risk Finance*, pages 1–11, Spring 2001.

[27] A. Consiglio and S.A. Zenios. Integrated simulation and optimization models for tracking international fixed income indices. *Mathematical Programming, Series B*, 89:311–339, 2001.

[28] G. B. Dantzig. Linear programming under uncertainty. *Management Science*, 1:197–206, 1955.

[29] G. B. Dantzig and G. Infanger. Large-scale stochastic linear programs: importance sampling and Benders decomposition. Report sol 91–4, Department of Operations Research, Stanford University, 1991.

[30] R.S. Dembo, A.R. Aziz, D. Rosen, and M. Zerbs. Mark to Future: A framework for measuring risk and reward. Algorithmics publications, Algorithmics Inc., May 2000.

[31] M.A.H. Dempster and H.I. Gassmann. Stochastic programming: Using the expected value of perfect information to simplify the decision tree. In *Proceedings 15th IFIP Conference on System Modelling and Optimization*, pages 301–303, Zurich, 1991.

[32] C. Dert. *Asset Liability Management for Pension Funds*. PhD thesis, Erasmus University, Rotterdam, Netherlands, 1995.

[33] C. Dert and B. Oldenkamp. Optimal guaranteed return portfolios and the casino effect. *Operations Research*, 48:768–775, 2000.

[34] D. Duffie and K. Singleton. Simulated moments estimation of markov models of asset prices. *Econometrica*, 61:929–952, 1993.

[35] J. Dupacova, M. Bertocchi, and V. Morrigia. Postoptimality of scenario based financial planning models with an application to bond

portfolio management. In W.T. Ziemba and J.M. Mulvey, editors, *World Wide Asset and Liability Management*, pages 263–285, Cambridge, 1998. Cambridge University Press.

[36] P. Embrechts. Actuarial versus finacial pricing of insurance. *The Journal of Risk Finance*, 1(4):17–26, 2000.

[37] E.F. Fama. The behavior of stock-market prices. *Journal of Business*, 38:34–105, 1965.

[38] P.C. Fishburn. Mean-variance risk analysis with risk associated with below target returns. *American Economic Review*, 67:116–126, 1977.

[39] S.-E. Fleten, K. Høyland, and S.W. Wallace. The performance of stochastic dynamic and fixed mix portfolio models. *European Journal of Operational Research*, 1998. (to appear).

[40] A.B. Goldstein and B.G. Markowitz. Sofasim: A dynamic insurance model with investment structure, policy benefits and taxes. *Journal of Finance*, 37:595–604, 1982.

[41] B. Golub, M. Holmer, R. McKendall, L. Pohlman, and S.A. Zenios. Stochastic programming models for money management. *European Journal of Operational Research*, 85:282–296, 1997.

[42] J. Gondzio and R. Kouwenberg. High-performance computing for asset liability management. *Operations Research*, 1999. (to appear).

[43] J. Gondzio, R. Kouwenberg, and A.C.F. Vorst. Hedging options under transaction costs and stochastic volatility. *Journal of Economic Dynamics and Control*, 1999. (to appear).

[44] W.H. Green. *Econometric Analysis*. MacMillan, New York, 1990.

[45] N.H. Hakansson. Optimal investment and consumption strategies under risk, an uncertain lifetime and insurance. *International Economic Review*, 10:443–466, 1969.

[46] N.H. Hakansson. Optimal investment and consumption strategies under risk for a class of utility functions. *Econometrica*, 38:587–607, 1970.

[47] N.H. Hakansson. On optimal myopic portfolio policies, with and without serial correlation of yields. *Journal of Business*, 44:324–334, 1971.

[48] J.D. Hamilton. *Time Series Analysis*. Princeton University Press, Princeton NJ, 1994.

[49] L. Hansen and J. Scheinkman. Back to the future: Generating moment implications for continuous-time markov processes. *Econometrica*, 63:767–804, 1995.

[50] W. Harlow. Asset allocation in a downside risk framework. *Financial Analyst Journal*, 47:28–40, 1991.

[51] J.M. Harrison and D.M. Kreps. Martingales and arbitrage in multiperiod securities markets. *Journal of Economic Theory*, 20:381–408, 1979.

[52] R.S. Hiller and J. Eckstein. Stochastic dedication: Designing fixed income portfolios using massively parallel Benders decomposition. *Management Science*, 39(11):1422–1438, 1993.

[53] M.R. Holmer. The asset/liability management system at Fannie Mae. *Interfaces*, 24(3):3–21, 1994.

[54] M.R. Holmer and S.A. Zenios. The productivity of financial intermediation and the technology of financial product management. *Operations Research*, 43(6):970–982, 1995.

[55] K. Høyland. *Asset Liability Management for a Life Insurance Company: A stochastic programming approach*. PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway, 1998.

[56] K. Høyland and S.W. Wallace. Generating scenario trees for multi stage problems. *Management Science*, 1999. (to appear).

[57] J. Hull. *Options, Futures and other derivative securities*. Prentice-Hall, 1989.

[58] J. Hull and A. White. Pricing interest rate derivative securities. *Review of Financial Studies*, 3:573–592, 1990.

[59] F. Jamshidian and Y. Zhu. Scenario simulation: Theory and methodology. *Finance and Stochastics*, pages 43–67, 1997.

[60] N.J. Jobst and S.A. Zenios. Extending credit risk (pricing) models for simulation and valuation of portfolios of interest rate and credit risk

sensitive securities. Working paper, HERMES Center on Computational Finance & Economics, University of Cyprus, Nicosia, Cyprus, 2001.

[61] P. Jorion. *Value at Risk: The new Benchmark for Controlling Market Risk*. Irwin Professional Publishing, IL, 1996.

[62] G.G. Judge, R.C. Hill, W.E. Griffiths, U. Lütkepohl, and T.C. Lee. *The Theory and Practice of Econometrics*. John Wiley and Sons, New York, 1988.

[63] P. Kall and S.W. Wallace. *Stochastic Programming*. John Wiley & Sons, New York, 1994.

[64] T.S. Kim and E. Omberg. Dynamic nonmyopic portfolio behavior. *Review of Financial Studies*, 9:141–161, 1996.

[65] L. Kingsland. Projecting the financial condition of a pension plan using simulation analysis. *Journal of Finance*, 37:577–584, 1982.

[66] P. Klaassen. Discretized reality and spurious profits in stochastic programming models for asset/liability management. *European Journal of Operational Research*, 101:374–392, 1997.

[67] P. Klaassen. Financial asset-pricing theory and stochastic programming models for asset-liability management: A synthesis. *Management Science*, 44:31–48, 1998.

[68] H. Konno and K. Kobayashi. An integrated stock-bond portfolio optimization model. *Journal of Economic Dynamics and Control*, 21:1427–1444, 1997.

[69] H. Konno and H. Yamazaki. Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market. *Management Science*, 37(5):519–531, 1991.

[70] Y. Koskosides and A. Duarte. A scenario-based approach for active asset allocation. *The Journal of Portfolio Management*, pages 74–85, Winter 1997.

[71] R. Kouwenberg. Scenario generation and stochastic programming models for asset liability management. *European Journal of Operational Research*, 1998. (to appear).

[72] M.I. Kusy and W.T. Ziemba. A bank asset and liability management model. *Operations Research*, 34:356–376, 1986.

[73] J. Liu. Portfolio selection in stochastic environments. Working paper, Stanford University, Stanford, US, 1999.

[74] C.D. Maranas, I.P. Androulakis, C.A. Floudas, A.J. Berger, and J.M. Mulvey. Solving long-term financial planning problems via global optimization. *Journal Economic Dynamics and Control*, 21:1405–1425, 1997.

[75] H.M. Markowitz. Portfolio selection. *Journal of Finance*, 7:77–91, 1952.

[76] R.C. Merton. Lifetime portfolio selection under uncertainty: The continuous-time case. *Review of Economics and Statistics*, 51:247–257, 1969.

[77] R.C. Merton. Optimum consumption and portfolio rules in a continuous-time model. *Journal of Economic Theory*, 3:373–413, 1971.

[78] R.C. Merton. *Continuous-Time Finance*. Blackwell, Cambridge, MA, 1990.

[79] J. Mossin. Optimal multiperiod portfolio policies. *Journal of Business*, 41:215–229, 1987.

[80] J.M. Mulvey. Incorporating transaction costs in models for asset allocation. In S.A. Zenios, editor, *Financial Optimization*, pages 243–259. Cambridge University Press, 1993.

[81] J.M. Mulvey. Generating scenarios for the towers perrin investment system. *Interfaces*, 26:1–15, 1996.

[82] J.M. Mulvey, G. Gould, and C. Morgan. An asseet and liability management system for Towers Perrin–Tillinghast. *Interfaces*, 30(1):96–114, 2000.

[83] J.M. Mulvey, R.J. Vanderbei, and S.A. Zenios. Robust optimization of large scale systems. *Operations Research*, 43:264–281, 1995.

[84] J.M. Mulvey and H. Vladimirou. Stochastic network programming for financial planning problems. *Management Science*, 38:1643–1664, 1992.

[85] J.M. Mulvey and S.A. Zenios. Capturing the correlations of fixed-income instruments. *Management Science*, 40:1329–1342, 1994.

[86] V. Naik. Finite state securities market models and arbitrage. In R. Jarrow et al, editor, *Handbooks in OR&MS*, pages 31–64. Elsevier Science, 1995.

[87] S.S. Nielsen and S.A. Zenios. Solving multistage stochastic network programs on massively parallel computers. *Mathematical Programming*, 75:227–250, 1996.

[88] S.S. Nielsen and S.A. Zenios. A stochastic programming model for funding single premium deferred annuities. *Mathematical Programming*, 75:177–200, 1996.

[89] G.Ch. Pflug and A. Świetanowski. Optimal scenario tree generation for multiperiod financial optimization. Technical report AURORA TR1998–22, Vienna University, 1998.

[90] F.M. Reddington. Review of the principles of life-office valuations. *Journal of the Institute of Actuaries*, 78:286–340, 1952.

[91] R.T. Rockafellar and S. Uryasev. Optimization of conditional Value-at-Risk. *The Journal of Risk*, 2(3):21–41, 2000.

[92] M. Rudolf and W.T. Ziemba. Intertemporal surplus management. Working paper, University of British Columbia, Vancouver, Canada, 1998.

[93] P.A. Samuelson. Lifetime portfolio selection by dynamic stochastic programming. *Review of Economics and Statistics*, 51:239–246, 1969.

[94] W.F. Sharpe and L.G. Tint. Liabilities - a new approach. *Journal of Portfolio Management*, pages 5–10, Winter 1990.

[95] M. S. Shtilman and S.A. Zenios. Constructing optimal samples from a binomial lattice. *Journal of Information and Optimization Sciences*, 14:1–23, 1993.

[96] F.A. Sortino and R. van der Meer. Downside risk - capturing whats at stake in investment situations. *Journal of Portfolio Management*, 17:27–31, 1991.

[97] R.J.-B. Wets. Stochastic programming. In G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd, editors, *Handbooks in Operations Research and Management Science*, volume 1, pages 573–629. North-Holland, Amsterdam, 1989.

[98] H.E. Winklevoss. Plasm: Pension liability and asset simulation model. *Journal of Finance*, 37:585–594, 1982.

[99] K.J. Worzel, C.Vassiadou-Zeniou, and S.A. Zenios. Integrated simulation and optimization models for tracking fixed-income indices. *Operations Research*, 42(2):223–233, March-April 1994.

[100] Y. Xia. Learning about predictability: The effects of parameter uncertainty on dynamic asset allocation. *Journal of Finance*, 56:585–594, 2000.

[101] S.A. Zenios. Massively parallel computations for financial modeling under uncertainty. In J. Mesirov, editor, *Very Large Scale Computing in the 21-st Century*, pages 273–294. SIAM, Philadelphia, PA, 1991.

[102] S.A. Zenios, editor. *Financial Optimization*. Cambridge University Press, Cambridge, England, 1993.

[103] S.A. Zenios. Asset/liability management under uncertainty for fixed-income securities. *Annals of Operations Research*, 59:77–98, 1995. (Reprinted in *World Wide Asset and Liability Modeling*, W.T. Ziemba and J.M. Mulvey (editors), Cambridge University Press, 1998.).

[104] S.A. Zenios, M. Holmer, R. McKendall, and C. Vassiadou-Zeniou. Dynamic models for fixed-income portfolio management under uncertainty. *Journal of Economic Dynamics and Control*, 22:1517–1541, 1998.

[105] W.T. Ziemba and J.M. Mulvey, editors. *Worldwide asset and liability modeling*. Cambridge University Press, Cambridge, England, 1998.

[106] W.T. Ziemba and R.G. Vickson, editors. *Stochastic Optimization Models in Finance*. Academic Press, 1975.