Multi-Stage Stochastic Programming

Sirong Luo

Faculty of Statistics and Data Science
Shanghai University of Finance and Economics

Table of Contents

Table of Contents

Multi-stage stochastic optimization

Multi-stage stochastic optimization can be seen as a generalization of the generic class of stochastic optimization model discussed in Chapter 10. Let $0, 1, \ldots, T$ index a set of stages where decisions are to be made. Assume that between two consecutive stages $t-1$ and t some random outcome ω_t is revealed. At each stage $t = 0, 1, \ldots, T$ we make a set of non-anticipatory decisions $\mathbf{x}_t$ that can only depend on the random information revealed up until that stage. Schematically, the process can be seen as follows:

$$\begin{array}{ccccccccccc} \text{decision} & & \text{random} & & \text{decision} & & \text{random} & & & & \text{random} \\ \mathbf{x}_0 & \rightsquigarrow & \text{draw } \omega_1 & \rightsquigarrow & \mathbf{x}_1 & \rightsquigarrow & \text{draw } \omega_2 & \rightsquigarrow \cdots \rightsquigarrow & & \rightsquigarrow & \text{draw } \omega_T \rightsquigarrow \\ \text{decision} & & & & & & & & & & \\ \mathbf{x}_T & & & & & & & & & & \end{array}$$

A multi-stage stochastic minimization problem is the following kind of multi-fold version of the two-stage stochastic model (10.2) discussed in Chapter 10:

$$\begin{aligned} \min_{\mathbf{x}_0} & g_0(\mathbf{x}_0) + \mathbb{E}\left[Q_1(\mathbf{x}_0, \omega_1)\right] \\ & \mathbf{x}_0 \in \mathcal{X}_0, \end{aligned} \tag{16.1}$$

where the recourse term $Q_1(\mathbf{x}_0, \omega_1)$ similarly depends on the decisions to be made at later stages:

$$Q_1(\mathbf{x}_0, \omega_1) := \min_{\mathbf{x}_1} \quad g_1(\mathbf{x}_1, \omega_1) + \mathbb{E}[Q_2(\mathbf{x}_1, \omega_2)]$$
$$\mathbf{x}_1 \in \mathcal{X}_1(\mathbf{x}_0, \omega_1)$$

with

$$Q_t(\mathbf{x}_{t-1}, \omega_t) := \min_{\mathbf{x}_t} g_t(\mathbf{x}_t, \omega_t) + \mathbb{E}[Q_{t+1}(\mathbf{x}_t, \omega_{t+1})]$$
$$\mathbf{x}_t \in \mathcal{X}_t(\mathbf{x}_{t-1}, \omega_t)$$

for $t = 2, \ldots, T-1$, and the last-stage recourse term $Q_T(\mathbf{x}_{T-1}, \omega_T)$ is of the form

$$Q_T(\mathbf{x}_{T-1}, \omega_T) := \min_{\mathbf{x}_T} \quad g_T(\mathbf{x}_T, \omega_T)$$
$$\mathbf{x}_T \in \mathcal{X}_T(\mathbf{x}_{T-1}, \omega_T).$$

The multi-stage optimization problem (16.1) can also be written as

$$\min_{\mathbf{x}_0 \in \mathcal{X}_0} g_0(\mathbf{x}_0) + \mathbb{E}\left[\min_{\mathbf{x}_1 \in \mathcal{X}_1(\mathbf{x}_0, \omega_1)} g_1(\mathbf{x}_1, \omega_1) + \cdots + \mathbb{E}\left[\min_{\mathbf{x}_T \in \mathcal{X}_T(\mathbf{x}_{T-1}, \omega_T)} g_T(\mathbf{x}_T, \omega_T)\right]\right]$$

Consider the special case of linear multi-stage stochastic optimization, where the components are linear. More precisely, each $g_t(\mathbf{x}_t, \omega_t) = \mathbf{c}_t^\top \mathbf{x}_t$ for some vector $\mathbf{c}_t$ and each inter-temporal constraint $\mathbf{x}_t \in \mathcal{X}_t(\mathbf{x}_{t-1}, \omega_t)$ is of the form

$$\mathbf{B}_t \mathbf{x}_{t-1} + \mathbf{A}_t \mathbf{x}_t = \mathbf{b}_t, \quad \mathbf{x}_t \geq 0,$$

where $\omega_t = (\mathbf{c}_t, \mathbf{A}_t, \mathbf{B}_t, \mathbf{b}_t)$ is only revealed at stage t.

In this case we often write:

$$
\begin{aligned}
\min_{x_0,x_1,\ldots,x_T} \quad & \mathbb{E}\left[c_0^\top x_0 + c_1^\top x_1 + \cdots + c_T^\top x_T\right] \\
\text{s.t.} \quad & A_0 x_0 = b_0, \\
& B_1 x_0 + A_1 x_1 = b_1, \\
& B_2 x_1 + A_2 x_2 = b_2, \\
& \vdots \\
& B_T x_{T-1} + A_T x_T = b_T, \\
& x_T \geq 0.
\end{aligned}
$$

Example

Example 16.1 (Financial planning example) Assume an investor has initial wealth W_0 at $t = 0$. At stage t she can invest in two asset classes: bonds and stocks. The (random) gross return on bonds from time $t-1$ to t is $R_{b,t}$ and the (random) gross return on stocks from $t-1$ to t is $R_{s,t}$. Assume that the investor needs to meet liabilities L_t at times $t = 1, \ldots, T$. She wants to maximize her expected wealth at time T (after covering the liabilities). Assume no shorting is allowed.

Formulation of the financial planning example

Variables: x_t : amount of money invested in bonds at stage t, for $t = 0, \ldots, T-1$;

y_t : amount of money invested in stocks at stage t, for $t = 0, \ldots, T-1$;

W_T : wealth at time T

$$
\begin{aligned}
\max \quad & \mathbb{E}(W_T) \\
\text{s.t.} \quad & x_0 + y_0 = W_0 \\
& R_{b,t} x_{t-1} + R_{s,t} y_{t-1} = L_t + x_t + y_t, \, t = 1, \ldots, T-1 \\
& R_{b,T} x_{T-1} + R_{s,T} y_{T-1} = L_T + W_T \\
& x_t, y_t \geq 0, \, t = 0, 1, \ldots, T-1 \\
& W_T \geq 0.
\end{aligned}
$$

Notice that in this model the parameters $R_{b,t}$ and $R_{s,t}$ are unknown prior to time t.

Table of Contents

As discussed in Chapter 10 for the two-stage case, a multi-stage optimization model can be recast as a deterministic equivalent if each of the random outcomes has a discrete distribution. In this case for each stage $t = 1, 2, \ldots, T$ there is a finite set of possible values or realizations $\left\{\omega_t^1, \ldots, \omega_t^S\right\}$ for the random outcome ω_t. These sets of realizations can be described by an event tree as depicted in Figure 16.1 for a problem with three stages. In this particular tree, the random variables ω_1 and ω_2 have two- and five-valued discrete distributions respectively. The tree structure is associated with the discrete filtration generated by the discrete-time random process $\omega_{[t]} := (\omega_1, \omega_2, \ldots, \omega_t), t = 1, \ldots, T$. In particular, each possible value of ω_2 has a unique predecessor value of ω_1.

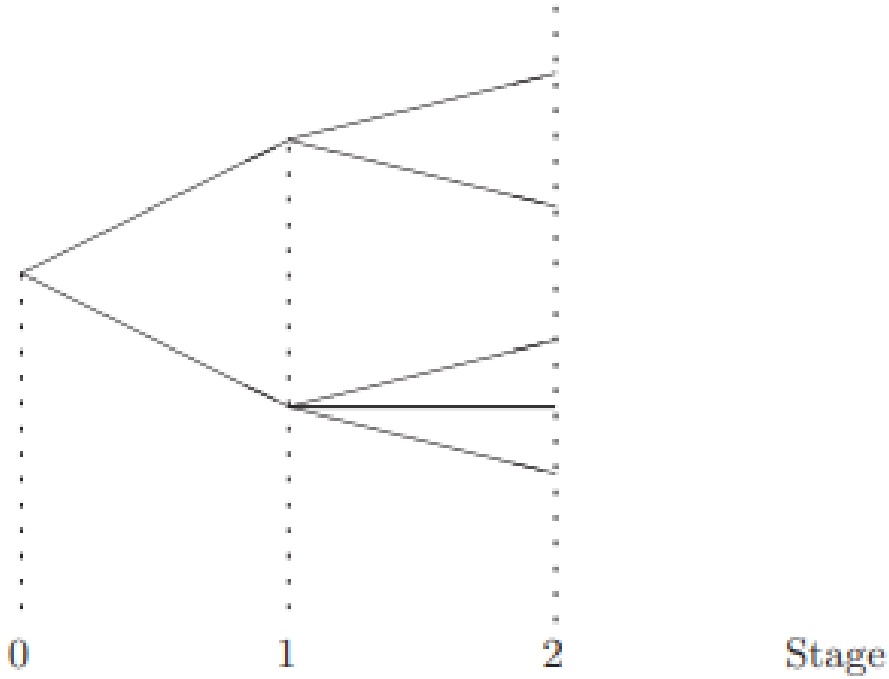


Figure: Event tree for a three-stage model

The set of scenarios described by the event tree in turn yield a deterministic equivalent of a multi-stage stochastic optimization model. We next illustrate this equivalence for the stochastic optimization model described in Example 16.1 in a particularly simple event tree. We subsequently describe the deterministic equivalent for linear multi-stage stochastic programs in more general event trees.

Example

Example 16.2 (Financial planning revisited) Consider the model described in Example 16.1. Suppose $T = 2$ and there are two equally likely outcomes (" H " and " T ") for the joint returns ($R_{b,t}, R_{s,t}$) over each period, say $(R_{b,t}(H), R_{s,t}(H)) = (1.14, 1.25)$ and $(R_{b,t}(T), R_{s,t}(T)) = (1.1, 1.06)$

$$
\begin{aligned}
\min\ & \mathbf{c}_0^\top \mathbf{x}_0 + p_1^1(\mathbf{c}_1^1)^\top \mathbf{x}_1^1 + p_1^2(\mathbf{c}_1^2)^\top \mathbf{x}_1^2 + p_2^1(\mathbf{c}_2^1)^\top \mathbf{x}_2^1 + p_2^2(\mathbf{c}_2^2)^\top \mathbf{x}_2^2 + p_2^3(\mathbf{c}_2^3)^\top \mathbf{x}_2^3 \\
& \qquad + p_2^4(\mathbf{c}_2^4)^\top \mathbf{x}_2^4 + p_2^5(\mathbf{c}_2^5)^\top \mathbf{x}_2^5 \\
\text{s.t.}\ & \mathbf{A}_0 \mathbf{x}_0 = \mathbf{b}_0 \\
& \mathbf{B}_1^1 \mathbf{x}_0 + \mathbf{A}_1^1 \mathbf{x}_1^1 = \mathbf{b}_1^1 \\
& \mathbf{B}_1^2 \mathbf{x}_0 + \mathbf{A}_1^2 \mathbf{x}_1^2 = \mathbf{b}_1^2 \\
& \mathbf{B}_2^1 \mathbf{x}_1^1 + \mathbf{A}_2^1 \mathbf{x}_2^1 = \mathbf{b}_2^1 \\
& \mathbf{B}_2^2 \mathbf{x}_1^1 + \mathbf{A}_2^2 \mathbf{x}_2^2 = \mathbf{b}_2^2 \\
& \mathbf{B}_2^3 \mathbf{x}_1^2 + \mathbf{A}_2^3 \mathbf{x}_2^3 = \mathbf{b}_2^3 \\
& \mathbf{B}_2^4 \mathbf{x}_1^2 + \mathbf{A}_2^4 \mathbf{x}_2^4 = \mathbf{b}_2^4 \\
& \mathbf{B}_2^5 \mathbf{x}_1^2 + \mathbf{A}_2^5 \mathbf{x}_2^5 = \mathbf{b}_2^5 \\
& \mathbf{x}_0,\ \mathbf{x}_1^1,\ \mathbf{x}_1^2,\ \mathbf{x}_2^1,\ \mathbf{x}_2^2,\ \mathbf{x}_2^3,\ \mathbf{x}_2^4,\ \mathbf{x}_2^5 \geq \mathbf{0}.
\end{aligned}
$$

Figure: Event tree for financial planning model

Figure 16.2 illustrates the corresponding scenario tree. In this event tree the labels " H " and " T " on the edges indicate the specific outcome between two consecutive stages. Observe that each of the four scenarios HH, HT, TH, TT in the event tree occurs with probability $1/4$.
The scenario tree in turn yields a deterministic equivalent formulation for the financial planning stochastic optimization model. In the deterministic equivalent the stage 0 decisions are made at the root of the tree and thus may not depend on any of the random outcomes. The stage 1 decisions may depend on the initial path H or T realized up to stage 1 in the event tree. Finally, the stage 2 decisions may depend on the path HH, HT, TH, or TT realized up to stage 2 in the event tree. The corresponding adaptiveness of the variables and constraints is explicitly reflected in the following deterministic equivalent formulation.

Scenario optimization model

Variables:

x_0, y_0 : money in bonds and stocks at $t = 0$;

$x_1(H), x_1(T), y_1(H), y_1(T)$: money in bonds and stocks at $t = 1$;

$W_2(\mathrm{HH}), W_2(HT), W_2(TH), W_2(TT)$: wealth at $t = 2$

$$\max \frac{1}{4}\left(W_2(HH) + W_2(HT) + W_2(TH) + W_2(TT)\right)$$

$$\text{s.t.} \quad x_0 + y_0 = W_0 \quad \text{(stage 0)}$$

$$1.14x_0 + 1.25y_0 = L_1 + x_1(H) + y_1(H) \quad \text{(stage 1, path } H\text{)}$$

$$1.1x_0 + 1.06y_0 = L_1 + x_1(T) + y_1(T) \quad \text{(stage 1, path } T\text{)}$$

$$1.14x_1(H) + 1.25y_1(H) = L_2 + W_2(HH) \quad \text{(stage 2, } HH\text{)}$$

$$1.1x_1(H) + 1.06y_1(H) = L_2 + W_2(HT) \quad \text{(stage 2, } HT\text{)}$$

$$1.14x_1(T) + 1.25y_1(T) = L_2 + W_2(TH) \quad \text{(stage 2, } TH\text{)}$$

$$1.1x_1(T) + 1.06y_1(T) = L_2 + W_2(TT) \quad \text{(stage 2, } TT\text{)}$$

$$x_0, y_0, x_1(H), x_1(T), y_1(H), y_1(T) \geq 0$$

$$W_2(HH), W_2(HT), W_2(TH), W_2(TT) \geq 0.$$

The above scenario optimization approach is quite flexible. In particular, consider a variation of the above financial planning model where the objective is $\max \mathbb{E}\left(U\left(W_T\right)\right)$ for some concave utility function $U(W)$. The corresponding deterministic equivalent has exactly the same variables and constraints as the one above and the following objective:

$$\max \frac{1}{4}\left(U\left(W_2(HH)\right)+U\left(W_2(TH)\right)+U\left(W_2(HT)\right)+U\left(W_2(TT)\right)\right).$$

Furthermore, if $U(\cdot)$ is piecewise linear, then the problem can be recast as a linear program. (See Exercise 16.1.)
Consider now the general multi-stage linear stochastic program (16.2). Suppose each random vector $\omega_t = (\mathbf{c}_t, \mathbf{A}_t, \mathbf{B}_t, \mathbf{b}_t)$ has a discrete distribution and consider their event tree representation. The description of the deterministic equivalent relies on the following notation. Let

$$\Omega_t := \left\{ \omega_t^k = \left(\mathbf{c}_t^k, \mathbf{A}_t^k, \mathbf{B}_t^k, \mathbf{b}_t^k \right) : k = 1, \ldots, S_t \right\}$$

be the set of possible realizations of the random variable ω_t for some integer $S_t \geq 1$ and for each stage $t = 1, \ldots, T$. Let $p_t^k = \mathbb{P}\left(\omega_t = \omega_t^k\right)$, with $k = 1, \ldots, S_t, t = 1, \ldots, T$. The set $\Omega_t = \left\{ \omega_t^1, \ldots, \omega_t^{S_t} \right\}$ corresponds to the nodes in layer t of the event tree, which can be conveniently denoted $(t, 1), \ldots, (t, S_t)$ as illustrated in Figure 16.3.

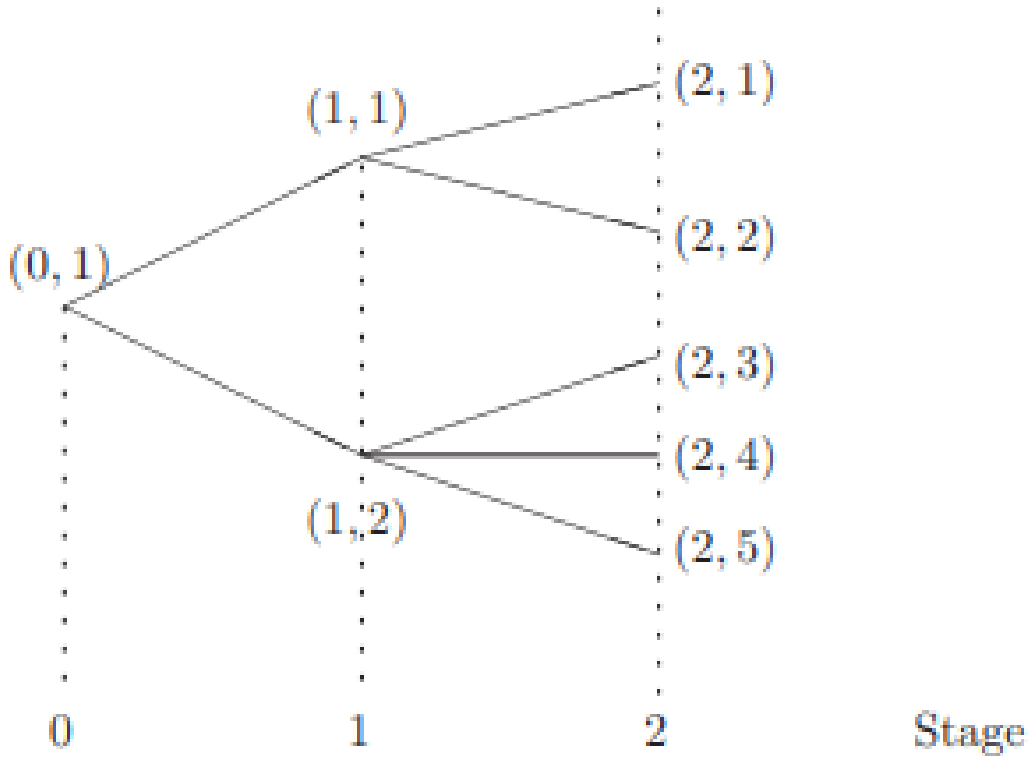


Figure: Event tree for a three-stage model with node labels

Observe that in the event tree there is always a single root node $(0,1)$ in layer 0 . Furthermore, for each $t = 1, \ldots, T-1$ each node (t,k) has a unique predecessor $(t-1,\hat{k})$ in the immediately preceding layer of the tree. For instance, in the event tree depicted in Figure 16.3 the predecessors of each of the nodes in layer 2 is a unique node in layer 1 as follows

$$\hat{1} = \hat{2} = 1, \hat{3} = \hat{4} = \hat{5} = 2.$$

Observe that the probability of a non-terminal node equals the combined probability of its direct descendants; that is, for $t = 1, \ldots, T$ and for every node $(t-1,\ell)$ we have

$$p_{t-1}^{\ell} = \sum_{(t,k):\hat{k}=\ell} p_t^k$$

Consider the multi-stage linear stochastic program (16.2) and assume the random outcomes are described via a suitable event tree. We next detail the variables, objective, and constraints of the corresponding deterministic linear program equivalent.

Variables: Stage 0 variables: $\mathbf{x}_0$. Stage t variables can be adapted to the S_t possible paths up to stage t; that is,

$$\mathbf{x}_t^k, \quad k = 1, \ldots, S_t.$$

Objective: The deterministic equivalent of the objective function

$$\min \mathbb{E}\left[\mathbf{c}_0^\top \mathbf{x}_0 + \mathbf{c}_1^\top \mathbf{x}_1 + \cdots + \mathbf{c}_T^\top \mathbf{x}_T\right] = \min \mathbb{E}\left[\mathbf{c}_0^\top \mathbf{x}_0 + \sum_{t=1}^{T} \mathbf{c}_t^\top \mathbf{x}_t\right]$$

is

$$\min \mathbf{c}_0^\top \mathbf{x}_0 + \sum_{t=1}^{T} \sum_{k=1}^{S_t} p_t^k \left(\mathbf{c}_t^k\right)^\top \mathbf{x}_t^k$$

Constraints: The deterministic equivalent of each inter-temporal constraint

$$\mathbf{B}_t \mathbf{x}_{t-1} + \mathbf{A}_t \mathbf{x}_t = \mathbf{b}_t, \quad \mathbf{x}_t \geq 0$$

is the set of constraints

$$\mathbf{B}_t^k \mathbf{x}_{t-1}^{\hat{k}} + \mathbf{A}_t^k \mathbf{x}_t^k = \mathbf{b}_t, \quad \mathbf{x}_t^k \geq 0, \text{ for } k = 1, \ldots, S_t$$

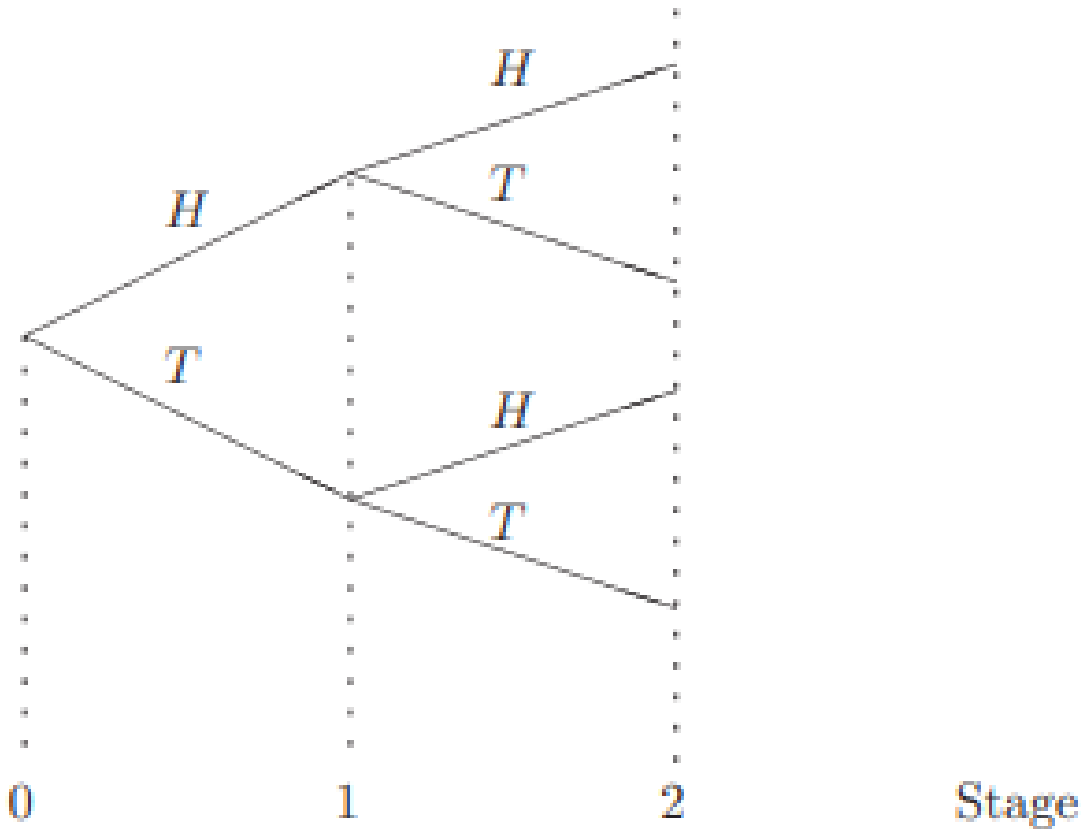
Observe that these constraints link the stage t variables $\mathbf{x}_t^k$ associated with the layer t nodes (t, k) with the variable associated with their predecessor $(t-1, \hat{k})$.

Thus the complete deterministic equivalent of (16.2) is as follows:

$$\min c_0^\top \mathbf{x}_0 + \sum_{k=1}^{S_1} p_1^k \left(\mathbf{c}_1^k\right)^\top \mathbf{x}_1^k + \cdots + \sum_{k=1}^{S_T} p_T^k \left(\mathbf{c}_T^k\right)^\top \mathbf{x}_T^k$$

$$\begin{array}{lll}
\text{s.t. } \mathbf{A}_0\mathbf{x}_0 & & = \mathbf{b}_0 \\
\mathbf{B}_1^k\mathbf{x}_0 + & \mathbf{A}_1^k\mathbf{x}_1^k & = \mathbf{b}_1^k, k = 1, \ldots, S_1 \\
& \mathbf{B}_2^k\mathbf{x}_1^k + \mathbf{A}_2^k\mathbf{x}_2^k & = \mathbf{b}_2, k = 1, \ldots, S_2 \\
& \mathbf{B}_T^k\mathbf{x}_{T-1}^{\hat{k}} + & \mathbf{A}_T^k\mathbf{x}_T^k = \mathbf{b}_T^k, k = 1, \ldots, S_T \\
\mathbf{x}_0, \mathbf{x}_1^k, & \ldots & \mathbf{x}_T^k \geq 0.
\end{array}$$

For example, if T $= 2$ and the event tree is as depicted in Figure 16.3, then the deterministic equivalent is



Observe that the constraint matrix in the above model has the following structure:

$$\begin{bmatrix}
\mathbf{A}_0 & & & & & & \\
\mathbf{B}_1^1 & \mathbf{A}_1^1 & & & & & \\
\mathbf{B}_1^2 & & \mathbf{A}_1^2 & & & & \\
 & \mathbf{B}_2^1 & & \mathbf{A}_2^1 & & & \\
 & \mathbf{B}_2^2 & & & \mathbf{A}_2^2 & & \\
 & & \mathbf{B}_2^3 & & & \mathbf{A}_2^3 & \\
 & & \mathbf{B}_2^4 & & & & \mathbf{A}_2^4 \\
 & & \mathbf{B}_2^5 & & & &
\end{bmatrix}.$$

The constraint matrix for the general deterministic equivalent (16.3) has a similar type of structure.

Table of Contents

A key aspect of multi-stage stochastic programming is the generation of scenarios so that the deterministic equivalent formulation (16.3) accurately represents the underlying stochastic optimization problem.
There are two separate issues. First, one needs to model the correlation over time among the random parameters. For a pension fund, such a model might relate wage inflation (a random parameter that influences the liability side) to interest rates and stock prices (random parameters that influence the asset side). Below we discuss a simple autoregressive model that can be used for this purpose.

A second issue is the construction of a scenario tree from these inter-temporal statistical models: A finite number of scenarios must reflect as accurately as possible the random processes modeled in the previous step, suggesting the need for a large number of scenarios. On the other hand, the linear program (16.3) can only be solved if the size of the scenario tree is reasonable, suggesting a limited number of scenarios. To reconcile these two conflicting objectives, it might be crucial to use variance reduction techniques. We address these issues in this section.

Autoregressive Model

In order to generate the random parameters underlying the stochastic program, one needs to construct an economic model reflecting the correlation between the parameters. Historical data may be available. The goal is to generate meaningful time series for constructing the scenarios. One approach is to use an autoregressive model.
Specifically, if $\mathbf{r}_t$ denotes the random vector of parameters in period t, an autoregressive model is defined by

$$\mathbf{r}_t = \mathbf{D}_0 + \mathbf{D}_1\mathbf{r}_{t-1} + \cdots + \mathbf{D}_p\mathbf{r}_{t-p} + \boldsymbol{\epsilon}_t$$

where p is the number of lags used in the regression, $\mathbf{D}_0, \mathbf{D}_1, \ldots, \mathbf{D}_p$ are timeindependent constant matrices, which are estimated through statistical methods such as maximum likelihood, and ϵ_t is a vector of i.i.d. random disturbances with mean zero.

To illustrate this, consider a problem where the vector $\mathbf{r}_t$ consists of three random parameters: s_t, b_t, and m_t are the rates of return of stocks, bonds, and the money market, respectively, in year t. An autoregressive model with $p = 1$ has the form:

$$\begin{bmatrix} s_t \\ b_t \\ m_t \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} + \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix} \begin{bmatrix} s_{t-1} \\ b_{t-1} \\ m_{t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_t^s \\ \epsilon_t^b \\ \epsilon_t^m \end{bmatrix}, \quad t = 2, \ldots, T$$

Assuming independent error terms $\epsilon_t^s, \epsilon_t^b$, and ϵ_t^m, and using historical data, one can find the parameters $d_1, d_{11}, d_{12}, d_{13}$ in the first equation,

$$s_t = d_1 + d_{11}s_{t-1} + d_{12}b_{t-1} + d_{13}m_{t-1} + \epsilon_t^s$$

using standard linear regression tools that minimize the sum of squared errors ϵ_t^s. Useful statistics, such as the standard error σ_s of the estimates s_t, can also be obtained. Similarly for b_t and m_t.

Constructing Scenario Trees

The random distributions relating the various parameters of a stochastic program must be discretized to generate a set of scenarios that is adequate for its deterministic equivalent. Too few scenarios may lead to approximation errors. On the other hand, too many scenarios will lead to an explosion in the size of 16.3 Scenario Generation 257 the scenario tree, leading to an excessive computational burden. In this section, we discuss a simple random sampling approach and two variance reduction techniques: adjusted random sampling and tree fitting. Unfortunately, scenario trees constructed by these methods could contain spurious arbitrage opportunities. We end this section with a procedure to test that this does not occur.

Random Sampling

One can generate scenarios directly from the autoregressive model introduced in the previous section:

$$\mathbf{r}_t = \mathbf{D}_0 + \mathbf{D}_1\mathbf{r}_{t-1} + \cdots + \mathbf{D}_p\mathbf{r}_{t-p} + \boldsymbol{\epsilon}_t$$

where $\epsilon_t \sim N(\mathbf{0}, \Sigma)$ are independently distributed multivariate normal distributions with mean 0 and covariance matrix Σ.
In our example with three random parameters s_t, b_t, and m_t, and independent error terms $\epsilon_t^s, \epsilon_t^b, \epsilon_t^m$, the matrix Σ is a 3×3 diagonal matrix, with diagonal entries $\sigma_s, \sigma_b, \sigma_m$. Thirty branches or so may be needed to get a reasonable approximation of the distribution of the rates of return in stage 1. For a problem with three stages, 30 branches at each stage represent 27,000 scenarios.

With more stages, the size of the linear program (16.3) explodes. Kouwenberg (2001) performed tests on scenario trees with fewer branches at each node (such as a five-stage problem with branching structure 10-6-6-4-4, meaning ten branches at the root, then six branches at each node in the next stage and so on) and he concluded that random sampling on such trees leads to unstable investment strategies. This occurs because the approximation error made by representing parameter distributions by random samples can be significant in a small scenario tree. As a result the optimal solution of (16.3) is not optimal for the actual parameter distributions. How can one construct a scenario tree that more accurately represents these distributions, without blowing up the size of the linear program (16.3)?

Adjusted Random Sampling

An easy way of improving upon random sampling is as follows. Assume that each node of the scenario tree has an even number $K = 2k$ of branches. Instead of generating 2k random samples from the autoregressive model, generate k random samples only and use the negative of their error terms to compute the values on the remaining k branches. This will fit all the odd moments of the distributions correctly. In order to fit the variance of the distributions as well, one can scale the sampled values. The sampled values are all scaled by a multiplicative factor until their variance fits that of the corresponding parameter.

Tree Fitting

How can one best approximate a continuous distribution by a discrete distribution with K values? In other words, how should one choose values v_k and their probabilities p_k, for $k = 1, \ldots, K$, in order to approximate the given distribution as accurately as possible? A natural answer is to match as many of the moments as possible. In the context of a scenario tree, the problem is somewhat more complicated since there are several correlated parameters at each node and there is interdependence between periods as well. Hoyland and Wallace (2001) propose to formulate this fitting problem as a nonlinear program. The fitting problem can be solved either at each node separately or on the overall tree.

We explain the fitting problem at a node. Let S_l be the values of the statistical properties of the distributions that one desires to fit, for $l = 1, \ldots, s$. These might be the expected values of the distributions, the correlation matrix, the skewness, and kurtosis. Let $\mathbf{v}_k$ and p_k denote the vector of values on branch k and its probability, respectively, for $k = 1, \ldots, K$. Let $f_l(\mathbf{v}, \mathbf{p})$ be the mathematical expression of property l for the discrete distribution (for example, the mean of the vectors $\mathbf{v}_k$, their correlation, skewness, and kurtosis). Each property has a positive weight w_l indicating its importance in the desired fit. Hoyland and Wallace formulate the fitting problem as

$$\begin{array}{ll} \min_{\mathbf{v},\mathbf{p}} & \sum_l w_l \left(f_l(\mathbf{v}, \mathbf{p}) - S_l\right)^2 \\ \text{s.t.} & \sum_k p_k = 1 \\ & \mathbf{p} \geq 0 \end{array} \tag{16.4}$$

One might want some statistical properties to match exactly. As an example, consider again the autoregressive model:

$$\mathbf{r}_t = \mathbf{D}_0 + \mathbf{D}_1\mathbf{r}_{t-1} + \cdots + \mathbf{D}_p\mathbf{r}_{t-p} + \boldsymbol{\epsilon}_t$$

where $\boldsymbol{\epsilon_t} \sim N(\mathbf{0}, \Sigma)$ are independently distributed multivariate normal distributions with mean 0 and covariance matrix Σ. To simplify notation, let us write $\boldsymbol{\epsilon}$ instead of $\boldsymbol{\epsilon}_t$. The random vector $\boldsymbol{\epsilon}$ has distribution $N(\mathbf{0}, \Sigma)$ and we would like to approximate this continuous distribution by a finite number of disturbance vectors ϵ^k occuring with probability p_k, for $k = 1, \ldots, K$. Let ϵ_q^k denote the q th component of vector $\boldsymbol{\epsilon}^k$. One might want to fit the mean of $\boldsymbol{\epsilon}$ exactly and its covariance matrix as well as possible. In this case, the fitting problem is

$$\begin{array}{ll} \min_{\boldsymbol{\epsilon}^1,\ldots,\boldsymbol{\epsilon}^K,\mathbf{p}} & \sum_{q=1}^{I}\sum_{r=1}^{I}\left(\sum_{k=1}^{K} p_k\epsilon_q^k\epsilon_r^k - \Sigma_{qr}\right)^2 \\ \text{s.t.} & \sum_{k=1}^{K} p_k\boldsymbol{\epsilon}^k = \mathbf{0} \\ & \sum_k p_k = 1 \\ & \mathbf{p} \geq 0 \end{array}$$

Arbitrage-Free Scenario Trees

Approximating the continuous distributions of the uncertain parameters by a finite number of scenarios in the linear program (16.3) typically creates modeling 16.4 Exercises 259 errors. In fact, if the scenarios are not chosen properly or if their number is too small, the supposed "linear programming equivalent" could be far from being equivalent to the original stochastic optimization problem. One of the most disturbing aspects of this phenomenon is the possibility of creating arbitrage opportunities when constructing the scenario tree. When this occurs, model (16.3) is flawed as it would be distorted by the arbitrage opportunities. Klaassen (2002) was the first to address this issue.

In particular, he shows how arbitrage opportunities can be detected ex post in a scenario tree. See Exercise 16.3 for details. When arbitrage opportunities exist, a simple solution is to discard the scenario tree and to construct a new one with more branches. Klaassen also discusses what constraints to add to the nonlinear program (16.4) in order to preclude arbitrage opportunities ex ante. The additional constraints are nonlinear, thus increasing the difficulty of solving (16.4).