Optimal Dynamic Portfolio Management with Stochastic Programming

John R. Birge
Northwestern University
OUTLINE

• Mean-variance versus other utility functions
• Discrete time, piecewise linear utility
• Policy structure
• Enhanced models
• Computation: abridged nested decomposition
Static Portfolio Model

Markowitz model

- Choose portfolio to minimize risk for a given return
- Find the **efficient frontier**
Markowitz Mean-Variance model

- For a given set of assets, find
  - fixed percentages to invest in each asset
  - maintain same percentage over time

- Needs
  - rebalance as returns vary
  - cash to meet obligations
Alternative Dynamic Model

- Assume possible outcomes over time
  - discretize generally
- In each period, choose mix of assets
- Can include transaction costs and taxes
- Can include liabilities over time
- Can include different measures of risk aversion
Example: Retirement Planning

- **GOAL:** Accumulate $G Y$ years from now

- **Assume:**
  - $W(0)$ - initial wealth
  - $K$ - investments
  - concave utility (piecewise linear)

**Note:** Similar to meeting a target or benchmark
Formulation with No Transactions Fees

- **SCENARIOS:** $\mathcal{？？？？}$
  - Probability, $p(\cdot)$
  - Groups, $S_t^1, \ldots, S_t^{S_t}$ at $t$

- **MULTISTAGE STOCHASTIC NLP FORM:**

$$\max \quad \mathcal{？？？？} p(\mathcal{？？？？} \mathcal{U}(W(\cdot, T)))$$

s.t. (for all $\cdot$):

- $\forall k \quad x(k, 1, \cdot) = W(o)$ (initial)
- $\forall k \quad r(k, t-1, \cdot) x(k, t-1, \cdot) - \forall k \quad x(k, t, \cdot) = 0$, all $t > 1$
- $\forall k \quad r(k, T-1, \cdot) x(k, T-1, \cdot) - W(\cdot, T) = 0$, (final)
- $x(k, t, \cdot) \geq 0$, all $k, t$

**Nonanticipativity:**

$x(k, t, ?') - x(k, t, ?) = 0$ if $?', ?' \in S_t^i$ for all $t, i, ?'$, $\mathcal{？？？？？？？？？？？？？？}$

*This says decision cannot depend on future.*
DATA and SOLUTIONS

• **ASSUME:**
  - Y=15 years
  - G=$80,000
  - T=3 (5 year intervals)
  - k=2 (stock/bonds)

• **Returns (5 year):**
  - Scenario A: r(stock) = 1.25   r(bonds)= 1.14
  - Scenario B: r(stock) = 1.06   r(bonds)= 1.12

• **Solution:**

<table>
<thead>
<tr>
<th>PERIOD</th>
<th>SCENARIO</th>
<th>STOCK</th>
<th>BONDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-8</td>
<td>41.5</td>
<td>13.5</td>
</tr>
<tr>
<td>2</td>
<td>1-4</td>
<td>65.1</td>
<td>2.17</td>
</tr>
<tr>
<td>2</td>
<td>5-8</td>
<td>36.7</td>
<td>22.4</td>
</tr>
<tr>
<td>3</td>
<td>1-2</td>
<td>83.8</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3-4</td>
<td>0</td>
<td>71.4</td>
</tr>
<tr>
<td>3</td>
<td>5-6</td>
<td>0</td>
<td>71.4</td>
</tr>
<tr>
<td>3</td>
<td>7-8</td>
<td>64.0</td>
<td>0</td>
</tr>
</tbody>
</table>
Static Markowitz Solution

Find efficient frontier:

![Graph showing efficient frontier with returns ranging from 0.115 to 0.155]
Results with Static Model

- Fixed proportion in stock and bonds in each period
- 80% stock for 15% return
- 40% stock for 14% return
- Results: no fixed proportion achieves target better than 50% of time
- Dynamic achieves target 87.5% of time
Analysis of Dynamic Model

• With discrete outcomes, p.l. utility:
  – Optimal solution has number of investments at most equal to number of branches in each period
  – Constrain the number of positive investments with the number of outcomes per period

• Impact of transaction fees and taxes
  – Additional constraints
  – Creates potential for more active investments in each period
  – Additional constraints can be imposed with linearization (representation other variance information)
  – Number of constraints can be used to limit number of investments
Other Model Gains

- Can include transaction costs
  - Fixed proportion requires transaction costs each period just to re-balance
  - can accumulate

- Can include tax considerations
  - Model size grows (lots of each asset)

- Maintain consistent utility
Current Study

- Portfolios of major indexes
- Constructed efficient frontier
- Developed decision tree form for stochastic program
- Gains in basic model for stochastic program of 3-5% over 10 periods
Solution Procedure

• Goal:
  – Take advantage of the problem structure
  – Reduce solutions of similar problems

• Approach:
  – Nested decomposition
  – Include sampling of large tree
General Multistage Stochastic Program

\[
\begin{align*}
\min & \quad c_1 x_1 + Q_2 x_1 \\
\text{s.t.} & \quad W_1 x_1 + h_1 \\
& \quad x_1 \leq 0 \\
& \quad Q_t x_{t+1,a} \text{ prob } Q_{t,k} x_{t+1,a} x_{t+k} \\
& \quad Q_{t,k} x_{t+1,a} x_{t+k} \text{ is a piecewise linear, convex function of } x_{t-1,a(k)} \\
& \quad Q_{N+1}(x_N) = 0, \text{ for all } x_N, \\
& \quad Q_{t,k}(x_{t-1,a(k)}) \text{ is a piecewise linear, convex function of } x_{t-1,a(k)}
\end{align*}
\]
Nested Decomposition

- In each subproblem, replace expected recourse function $Q_{t,k}(x_{t-1,a(k)})$ with unrestricted variable $\hat{?}_{t,k}$
  - **Forward Pass:**
    - Starting at the root node and proceeding forward through the scenario tree, solve each node subproblem
      $$\hat{Q}_{t,k} \min_{\hat{?}_{t,k}, x_{t,k}, \xi_t} \quad \begin{align*}
      \text{s.t.} & \quad c_t \hat{?}_{t,k} x_{t,k} + h_t \hat{?}_{t,k} x_{t,k} + T_{t+1} \hat{?}_{t,k} x_{t+1,a(k)} \\
      & \quad E_{t,k} x_{t,k} + e_t \hat{?}_{t,k} + d_{t,k} \xi_t \xrightarrow{?} \text{feasibility cuts?} \\
      & \quad x_{t,k} \xrightarrow{?} 0
      \end{align*}$$
    - Add feasibility cuts as infeasibilities arise
  - **Backward Pass**
    - Starting in top node of Stage $t = N-1$, use optimal dual values in descendant Stage $t+1$ nodes to construct new optimality cut. Repeat for all nodes in Stage $t$, resolve all Stage $t$ nodes, then $t \xrightarrow{t} t-1$.
  - Convergence achieved when
    $$Q_1 \xrightarrow{?} Q_2 x_1 \xrightarrow{?} 0$$
Pereira-Pinto Method

• Incorporates sampling into the general framework of the Nested Decomposition algorithm

• Assumptions:
  – relatively complete recourse
    • no feasibility cuts needed
  – serial independence
    • an optimality cut generated for any Stage t node is valid for all Stage t nodes

• Successfully applied to multistage stochastic water resource problems
Pereira-Pinto Method

1. Randomly select \( H N \)-Stage scenarios

2. Starting at the root, a forward pass is made through the sampled portion of the scenario tree (solving ND subproblems)

3. A statistical estimate of the first stage objective value is calculated using the total objective value obtained in each sampled scenario. The algorithm terminates if the current first stage objective value \( c_1 x_1 + \beta \) is within a specified confidence interval of

4. Starting in sampled node of Stage \( t = N - 1 \), solve all Stage \( t + 1 \) descendant nodes and construct new optimality cut. Repeat for all sampled nodes in Stage \( t \), then repeat for \( t = t - 1 \)
Pereira-Pinto Method

• Advantages
  – significantly reduces computation by eliminating a large portion of the scenario tree in the forward pass

• Disadvantages
  – requires a complete backward pass on all sampled scenarios
    • not well designed for bushier scenario trees
Abridged Nested Decomposition

• Also incorporates sampling into the general framework of Nested Decomposition
• Also assumes relatively complete recourse and serial independence
• Samples both the subproblems to solve and the solutions to continue from in the forward pass
Abridged Nested Decomposition

**Forward Pass**

1. Solve root node subproblem

2. Sample Stage 2 subproblems and solve selected subset

3. Sample Stage 2 subproblem solutions and branch in Stage 3 only from selected subset (i.e., nodes 1 and 2)

4. For each selected Stage t-1 subproblem solution, sample Stage t subproblems and solve selected subset

5. Sample Stage t subproblem solutions and branch in Stage t+1 only from selected subset
Abridged Nested Decomposition

Backward Pass
1. Starting in first branching node of Stage $t = N-1$, solve all Stage $t+1$ descendant nodes and construct new optimality cut for all stage $t$ subproblems. Repeat for all sampled nodes in Stage $t$, then repeat for $t = t - 1$

Convergence Test
1. Randomly select $H \, N$-Stage scenarios. For each sampled scenario, solve subproblems from root to leaf to obtain total objective value for scenario
2. Calculate statistical estimate of the first stage objective value $\bar{z}$
   - algorithm terminates if current first stage objective value $c_I x_I + ?_I$ is within a specified confidence interval of $\bar{z}$ else, a new forward pass begins
Computational Results

- Implementation of Pereira & Pinto Method and Abridged Nested Decomposition
  - written in C, uses CPLEX to solve subproblems
- Pereira & Pinto Method
  - uses a sample size of 30 for each problem
- Abridged Nested Decomposition
  - number of Stage t subproblems solved from each Stage t-1 branching value: 15
  - initial number of Stage t branching values: 2
    - number of Stage t branching values increases with each failed convergence test
- Both methods terminate when first stage objective value is within one standard deviation of statistical estimate
Computational Results

• Initial Test Problems
  – Dynamic Vehicle Allocation (DVA) problems of various sizes
    • set of homogeneous vehicles move full loads between set of sites
    • vehicles can move empty or loaded, remain stationary
    • demand to move load between two sites is stochastic
  – DVA.x.y.z
    • $x$ number of sites (8, 12, 16)
    • $y$ number of stages (4, 5)
    • $z$ number of distinct realizations per stage (30, 45, 60, 75)
  – largest problem has > 30 million scenarios
Computational Results (DVA.8)

CPU Time (seconds)

Fleet Size  50
Links     72

Seconds

DVA.8.4.30  DVA.8.4.45  DVA.8.4.60  DVA.8.4.75  DVA.8.5.30  DVA.8.5.45  DVA.8.5.60  DVA.8.5.75
Computational Results (DVA.12)

CPU Time (seconds)

Fleet Size 120
Links 168
Computational Results (DVA.16)
Additional Features for Portfolio Problems

• Serial independence
  – Increments are generally serial
  – Formulation is complex to address problem directly
  – Slows computation speed

• Using structure
  – Can still use structure but assume not correlation of returns over time
  – Currently under development
Conclusions

• Static portfolio models have problems with:
  – benchmark targets
  – transaction costs and taxes

• Dynamic stochastic programming models can address difficulties
  – variety of objectives
  – can use structure to meet additional requirements

• Computation of large problems using decomposition and special structure