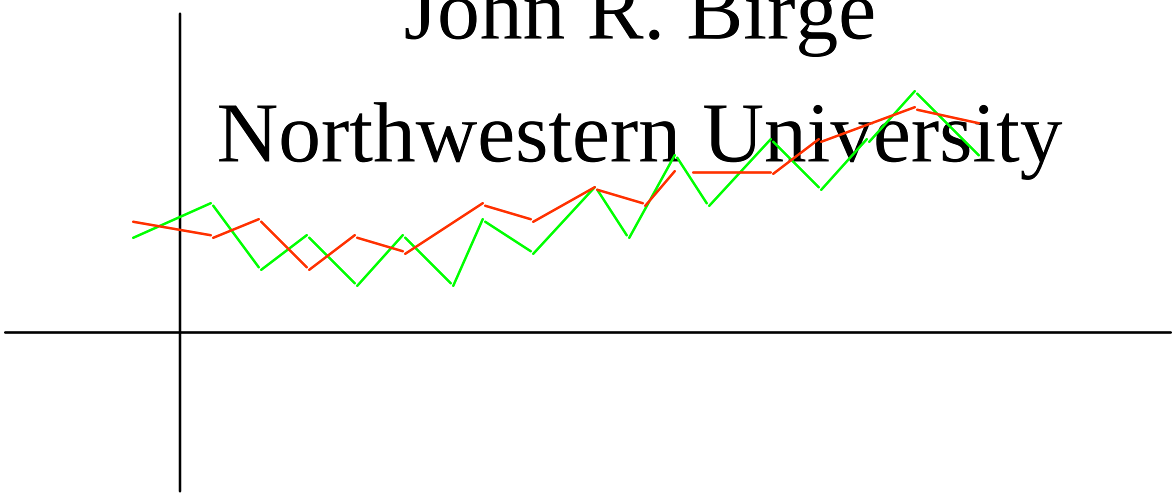
Optimal Dynamic Portfolio Management with Stochastic Programming

John R. Birge

Northwestern University

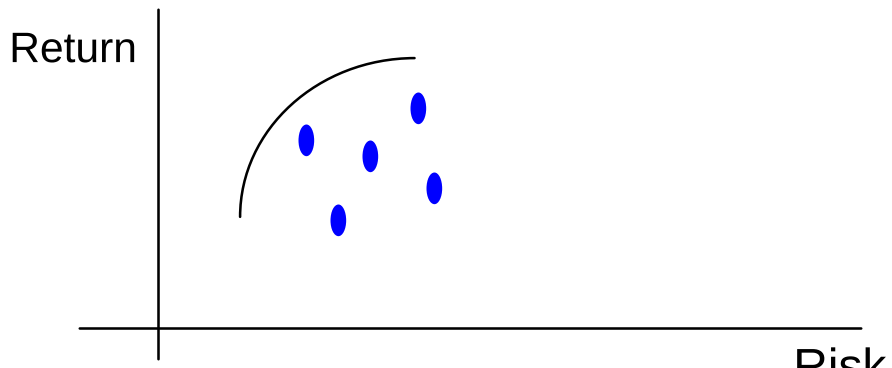


OUTLINE

- **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

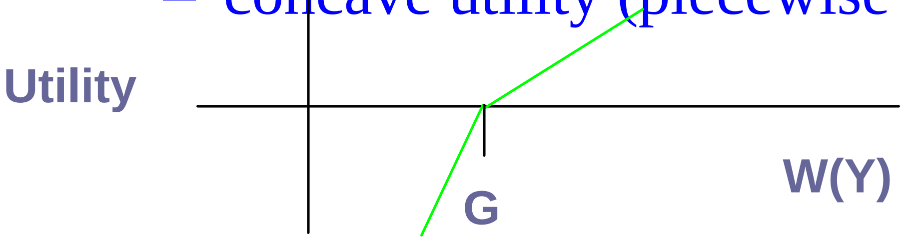
- **For a given set of assets, find**
  - **fixed percentages to invest in each asset**
  - **maintain same percentage over time**
- **Needs**
  - **rebalance as returns vary**
  - **cash to meet obligations**

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: ? ?? ??
  - Probability, p(?)
  - Groups, S^t_1, ..., S^t_{St} at t
- MULTISTAGE STOCHASTIC NLP FORM:

max ? ? p(? ???U(W(? , T))

s.t. (for all ?): ? k x(k,1, ?) = W(o) (initial)

? k r(k,t-1, ?) x(k,t-1, ?) - ? k x(k,t, ?) = 0 , all t >1;

? k r(k,T-1, ?) x(k,T-1, ?) - W(? , T) = 0, (final);

x(k,t, ?) >= 0, all k,t;

Nonanticipativity:

x(k,t, ? ') - x(k,t, ?) = 0 if ? ', ? ?? ?S^t_i for all t, i, ? ', ?

???????This says decision cannot depend on future.

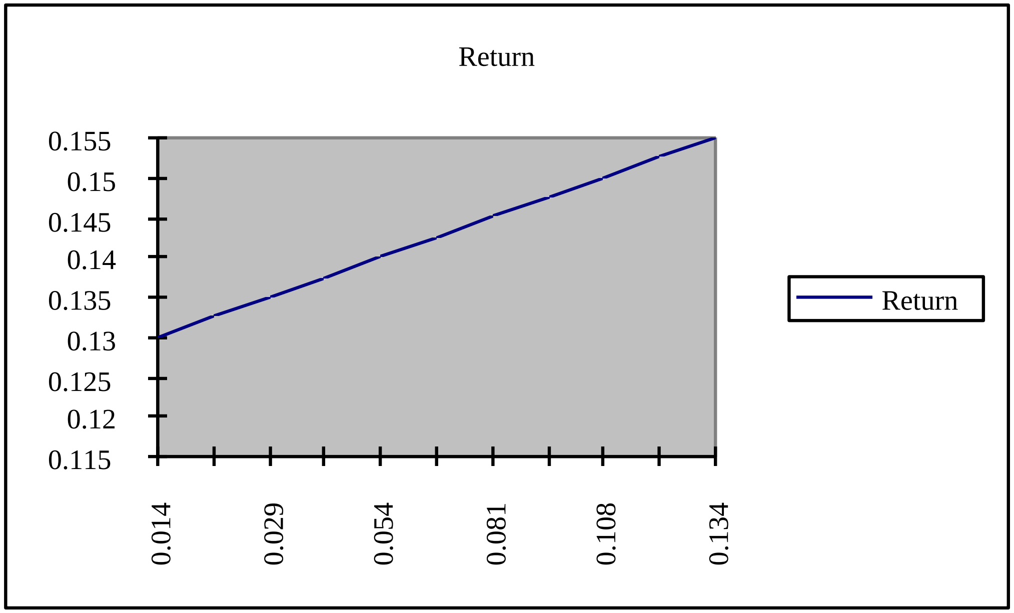
DATA and SOLUTIONS

- 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:

PERIOD	SCENARIO	STOCK	BONDS
1	1-8	41.5	13.5
2	1-4	65.1	2.17
2	5-8	36.7	22.4
3	1-2	83.8	0
3	3-4	0	71.4
3	5-6	0	71.4
3	7-8	64.0	0

Static Markowitz Solution

- **Find efficient frontier:**



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**

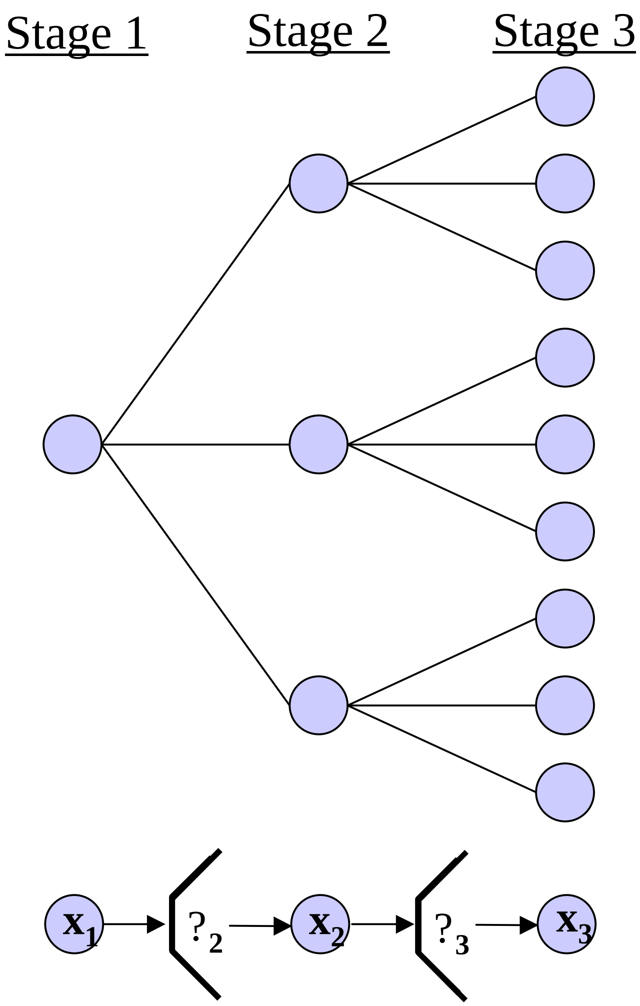
Current Study

- **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



$$
\begin{aligned}
\min \quad & c_1 x_1 + Q_2(x_1) \\
s.t. \quad & W_1 x_1 = h_1 \\
& x_1 \geq 0
\end{aligned}
$$

$$
Q_t(x_{t-1,a(k)}) = \sum_{\xi_{t,k} \in \Xi_t} prob(\xi_{t,k}) \, Q_{t,k}(x_{t-1,a(k)}, \xi_{t,k})
$$

$$
\begin{aligned}
Q_{t,k}(x_{t-1,a(k)}, \xi_{t,k}) = \min \quad & c_t(\xi_{t,k}) x_{t,k} + Q_{t+1}(x_{t,k}) \\
s.t. \quad & W_t x_{t,k} = h_t(\xi_{t,k}) - T_{t-1}(\xi_{t,k}) x_{t-1,a(k)} \\
& x_{t,k} \geq 0
\end{aligned}
$$

- $Q_{N+1}(x_N) = 0$, for all x_N,
- $Q_{t,k}(x_{t-1,a(k)})$ is a piecewise linear, convex function of $x_{t-1,a(k)}$

Nested Decomposition

- In each subproblem, replace expected recourse function $Q_{t,k}(x_{t-1,a(k)})$ with unrestricted variable $?_{t,k}$
  - Forward Pass:
    - Starting at the root node and proceeding forward through the scenario tree, solve each node subproblem

$$
\begin{array}{llllll}
\hat{Q}_{t,k}(x_{t?1,a(k)}, ?_{t,k}) ? & \min & c_t(?_{t,k}) x_{t,k} ? ?_{t,k} & & & \\
 & s.t. & W_t x_{t,k} & ? & h_t(?_{t,k}) ? T_{t?1}(?_{t,k}) x_{t?1,a(k)} & \\
 & & E_{t,k} x_{t,k} ? ?_{t,k} & ? & e_{t,k} & (optimality\ cuts) \\
 & & D_{t,k} x_{t,k} & ? & d_{t,k} & (feasibility\ cuts) \\
 & & x_{t,k} & ? & 0 &
\end{array}
$$

    - 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 → t-1.
  - Convergence achieved when

$$?_1 \quad ? \quad Q_2(x_1)$$

Pereira-Pinto Method

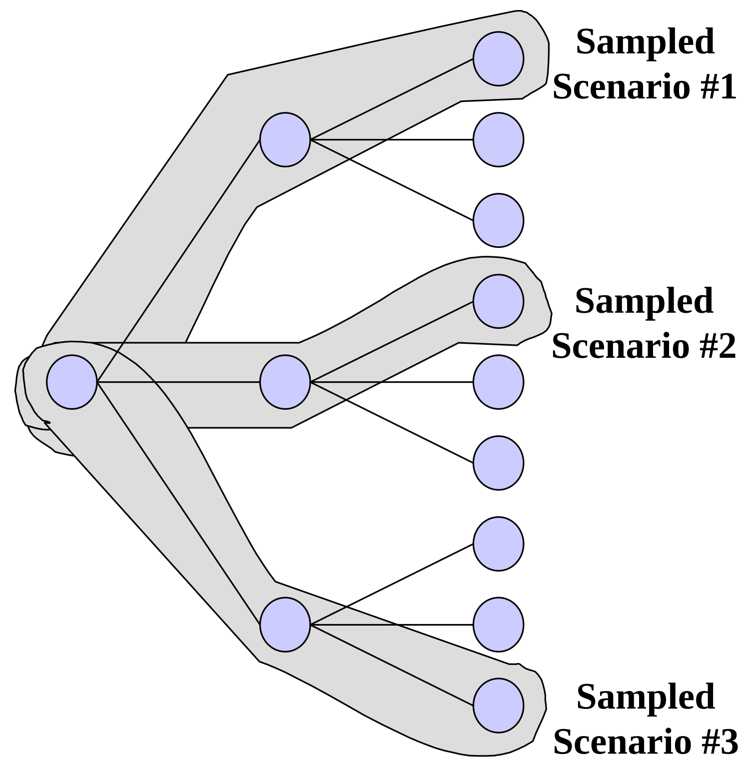
- Incorporates sampling into the general framework of the Nested Decomposition algorithm
- Assumptions:
  - relatively complete recourse
    - no feasibility cuts needed
  - serial independence
    - an optimality cut generated for any Stage t node is valid for all Stage t nodes
- Successfully applied 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 z is calculated using the total objective value obtained in each sampled scenario

the algorithm terminates if current first stage objective value $c_1x_1 + ?_1$ is within a specified confidence interval of z

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



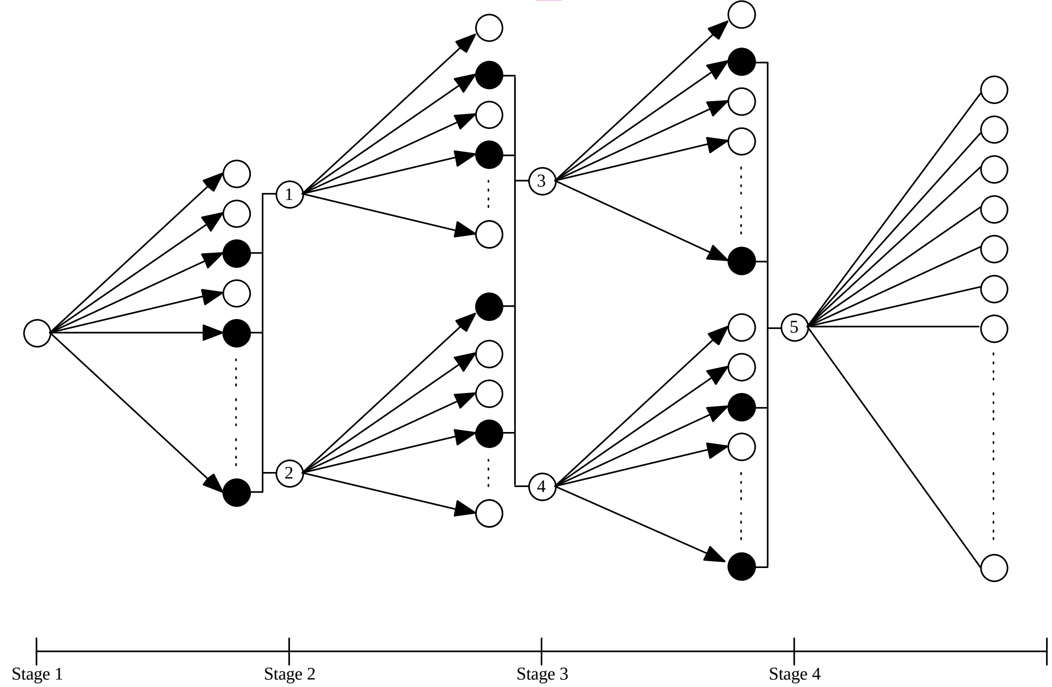
Pereira-Pinto Method

- 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



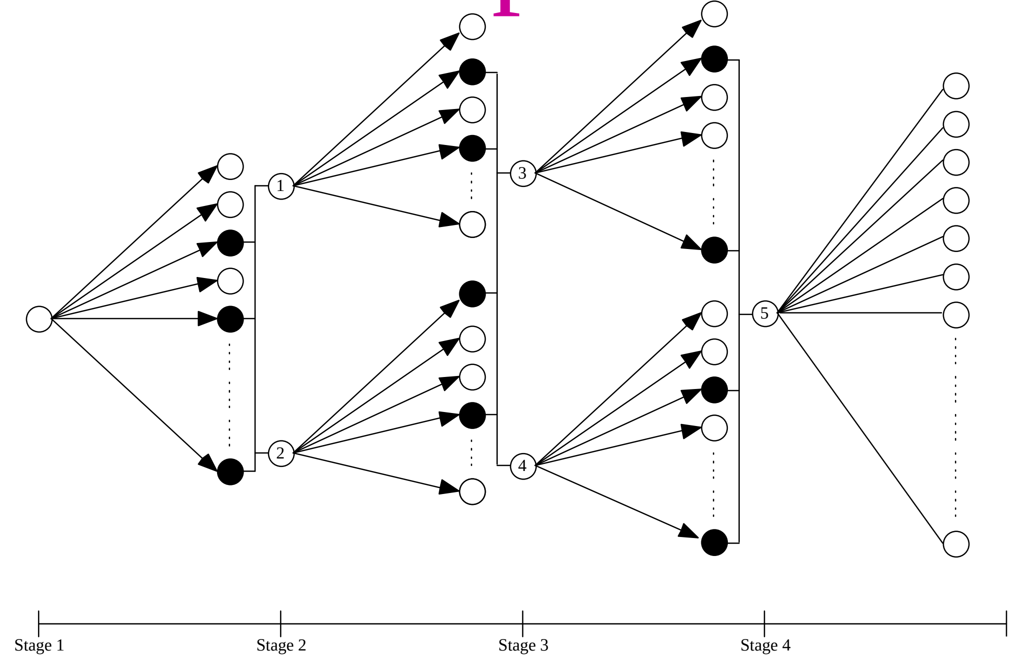
Forward Pass

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 $c_1x_1 + ?_1$ is within a specified confidence interval of $\bar{z}$, else, a new forward pass begins

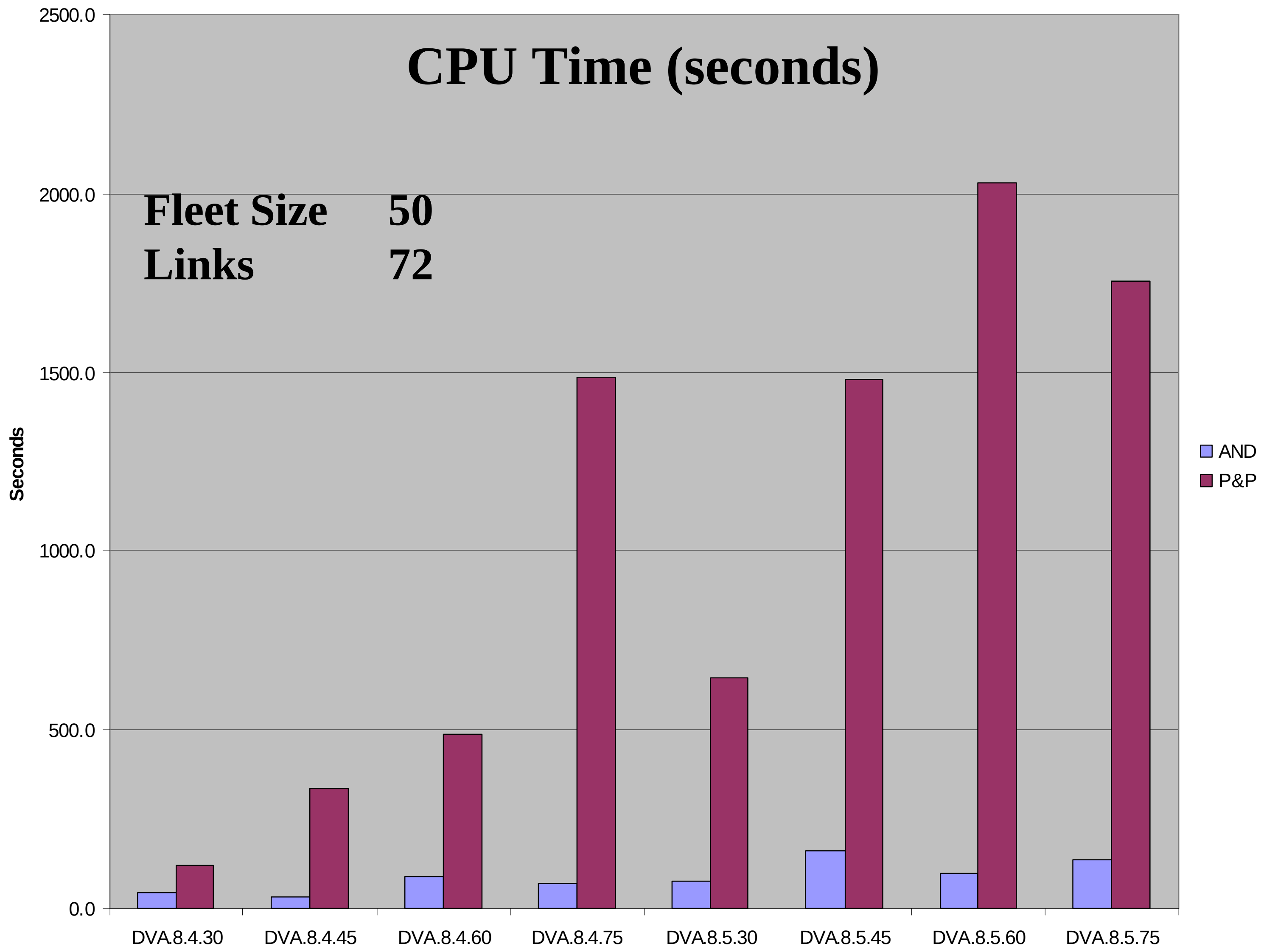
Computational Results

- 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

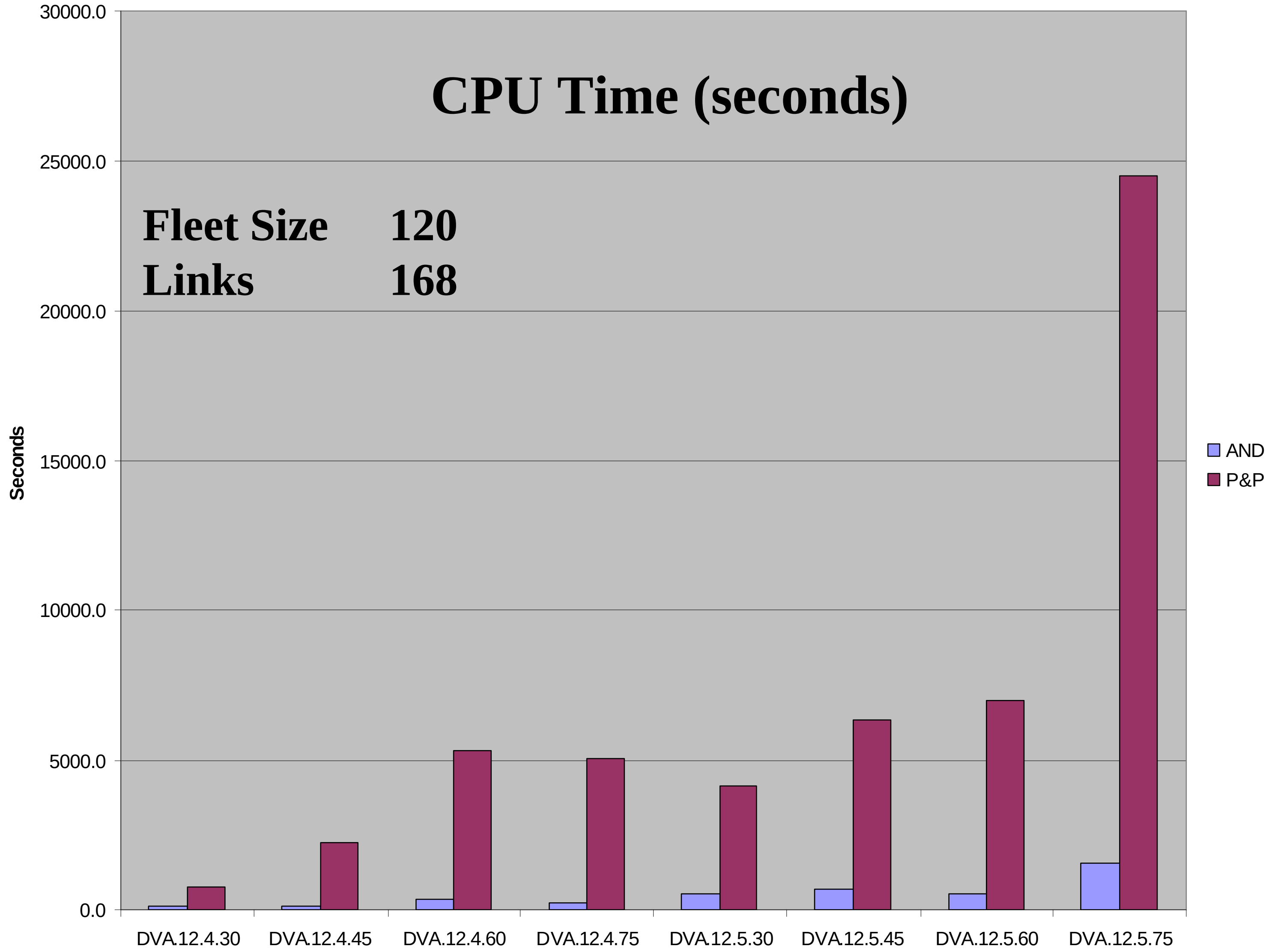
Computational Results

- 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

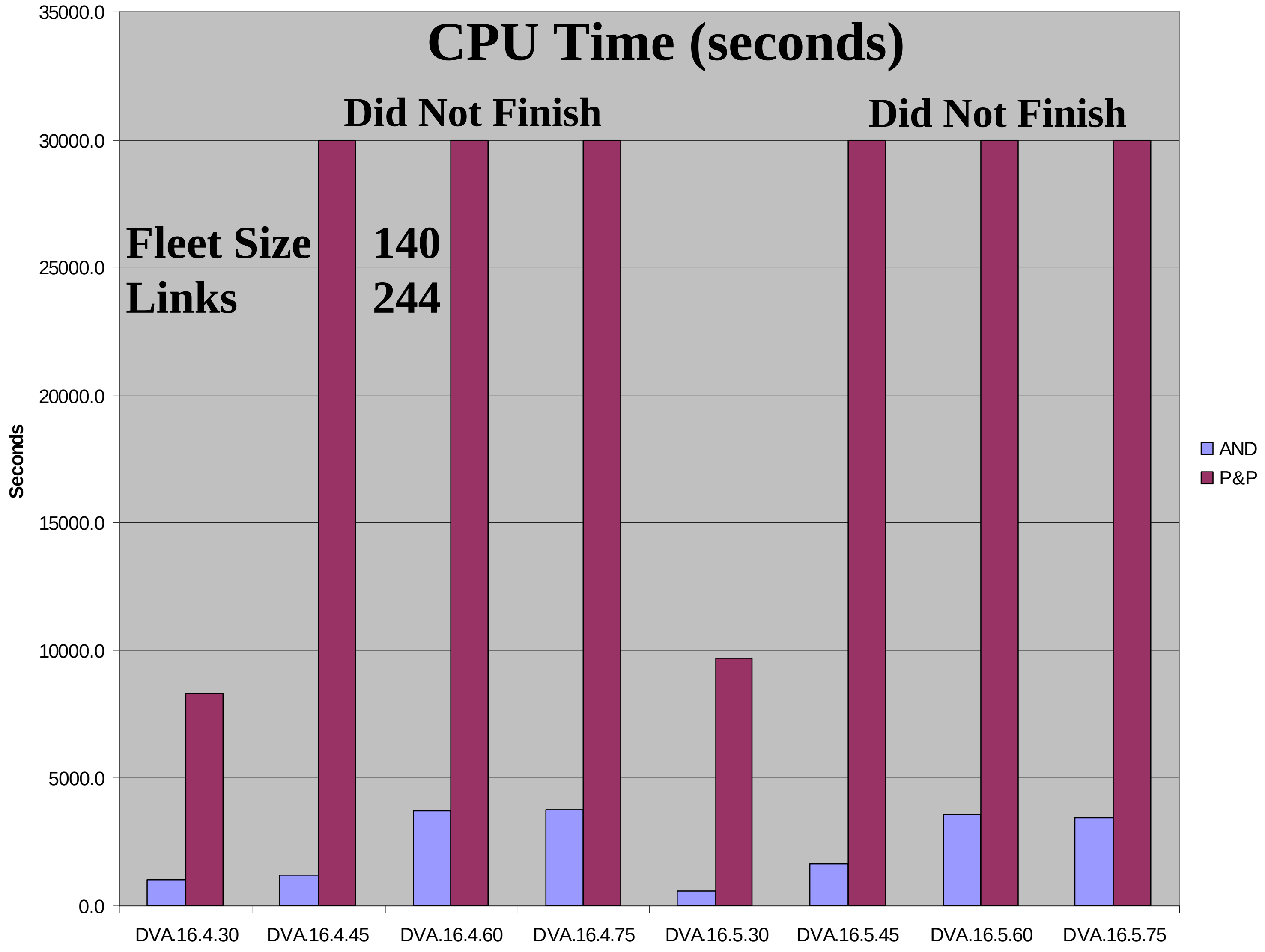
Computational Results (DVA.8)



Computational Results (DVA.12)



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