Computational Methods for Large-Scale Stochastic Dynamic Programs

John R. Birge

The University of Chicago Graduate School of Business

www.ChicagoGSB.edu/fac/john.birge

Theme

- Stochastic dynamic programs are:
  - big (exponential growth in time and state)
  - general (can model many situations)
  - structured (useful properties somewhere)
- Some hope for solution by:
  - modeling the “right” way
  - using structure wisely
  - approximating (with some guarantees/bounds)

Outline

- General Model – Observations
- Overview of approaches
- Factorization/sparsity (interior point/barrier)
- Decomposition
- Lagrangian and ADP methods
- Conclusions

General Stochastic Programming Model: Discrete Time

- Find $x=(x_1,x_2,\ldots,x_T)$ and p to

minimize $E_p [\sum_{t=1}^{T} f_t(x_t,x_{t+1},p)]$

s.t. $x_t \in X_t$, x_t **nonanticipative,** $p \in P$ **(distribution class)**

$P[h_t (x_t,x_{t+1},p_t) <= 0] >= a$ **(chance constraint)**

General Approaches:

- **Simplify distribution (e.g., sample) and form a mathematical program:**
  - **Solve step-by-step (dynamic program)**
  - **Solve as single large-scale optimization problem**
- **Use iterative procedure of sampling and optimization steps**

What about Continuous Time?

- Sometimes very useful to develop overall structure of value function
- May help to identify a policy that can be explored in discrete time (e.g., portfolio no-trade region)
- Analysis can become complex for multiple state variables
- Possible bounding results for discrete approximations (e.g., FEM approach)

Simplified Finite Sample Model

- Assume p is fixed and random variables represented by sample ξ^i_t for $t=1,2,..,T$, $i=1,\ldots,N_t$ with probabilities p^i_t ,$a(i)$ an *ancestor* of i, then model becomes (no chance constraints):

minimize $\Sigma_{t=1}^{T} \Sigma_{i=1}^{Nt} p^i_t \, f_t(x^{a(i)}_t, x^i_{t+1}, \xi^i_t)$

s.t. $x^i_t \in X^i_t$

Observations?

- Problems for different i are similar – solving one may help to solve others
- Problems may decompose across *i* and across *t* yielding
  - smaller problems (that may scale linearly in size)
  - opportunities for parallel computation.

Outline

- General Model – Observations
- Overview of approaches
- Factorization/sparsity (interior point/barrier)
- Decomposition
- Lagrangian and ADP methods
- Conclusions

.

Solving As Large-scale Mathematical Program

- Principles:
  - Discretization leads to mathematical program but large-scale
  - Use standard methods but exploit structure
- Direct methods
  - Take advantage of sparsity structure
    - Some efficiencies
  - Use similar subproblem structure
    - Greater efficiency
- Size
  - Unlimited (infinite numbers of variables)
  - Still solvable (caution on claims)

Standard Approaches

- Sparsity structure advantage
  - Partitioning
  - Basis factorization
  - Interior point factorization
- Similar/small problem advantage
  - DP approaches
    - Decomposition:
      - Benders, l-shaped (Van Slyke – Wets)
      - Dantzig-Wolfe (primal version)
      - Regularized (Ruszczynski)
    - Various sampling schemes (Higle/Sen stochastic decomposition, abridged nested decomposition)
    - Approximate DP (Bertsekas, Tsitsiklis, Van Roy..)
  - Lagrangian methods

Outline

- General Model – Observations
- Overview of approaches
- Factorization/sparsity (interior point/barrier)
- Decomposition
- Lagrangian methods
- Conclusions

Sparsity Methods: Stochastic Linear Program Example

- Two-stage Linear Model:

  $X_1 = \{x_1 | A\,x_1 = b, x_1 >= 0\}$

  $f_0(x_0,x_1) = c\,x_1$

  $f_1(x_1, x_2^i, \xi_2^i) = q\,x_2^i$ if $T\,x_1 + W\,x_2^i = \xi_2^i$,

  $x_2^i >= 0$; $+\infty$ otherwise

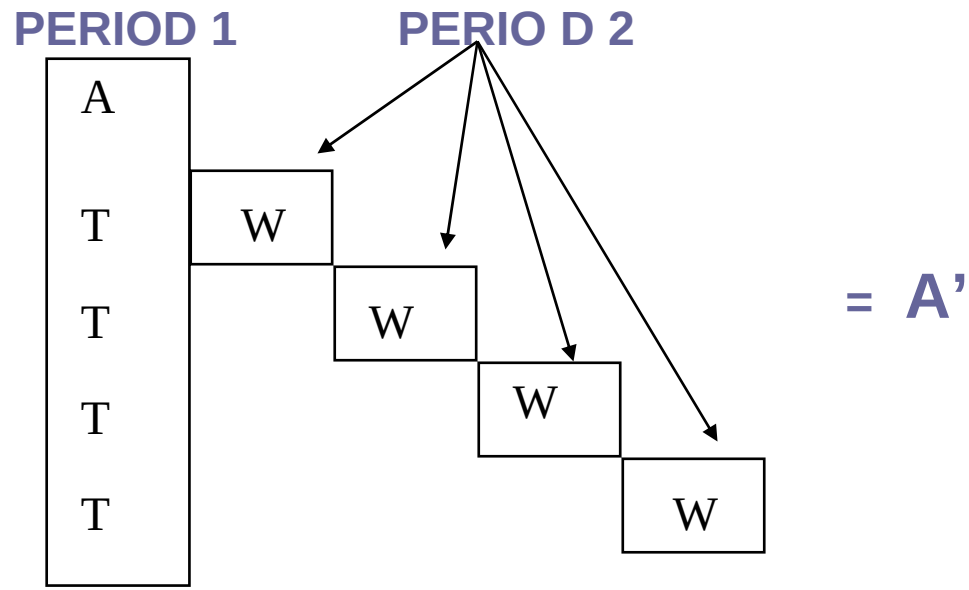
- Result: $\min c\,x_1 + \Sigma_{i=1}^{N_1} \mathbf{p}_2^i\, q\, x_2^i$

  $s.\,t.\ A\,x_1 = b, x_1 >= 0$

  $T\,x_1 + W\,x_2^i = \xi_2^i, x_2^i >= 0$

LP-BASED METHODS

- USING BASIS STRUCTURE



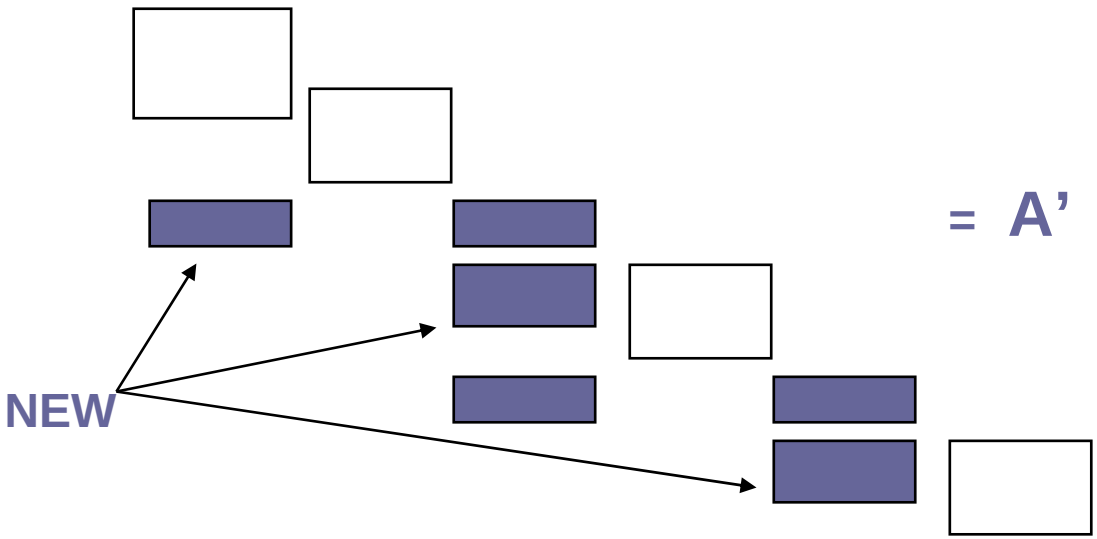
- **MODEST GAINS FOR SIMPLEX**

INTERIOR POINT MATRIX STRUCTURE

$A'D^2A'^T =$  **COMPLETE FILL-IN**

Alternatives For Interior Points

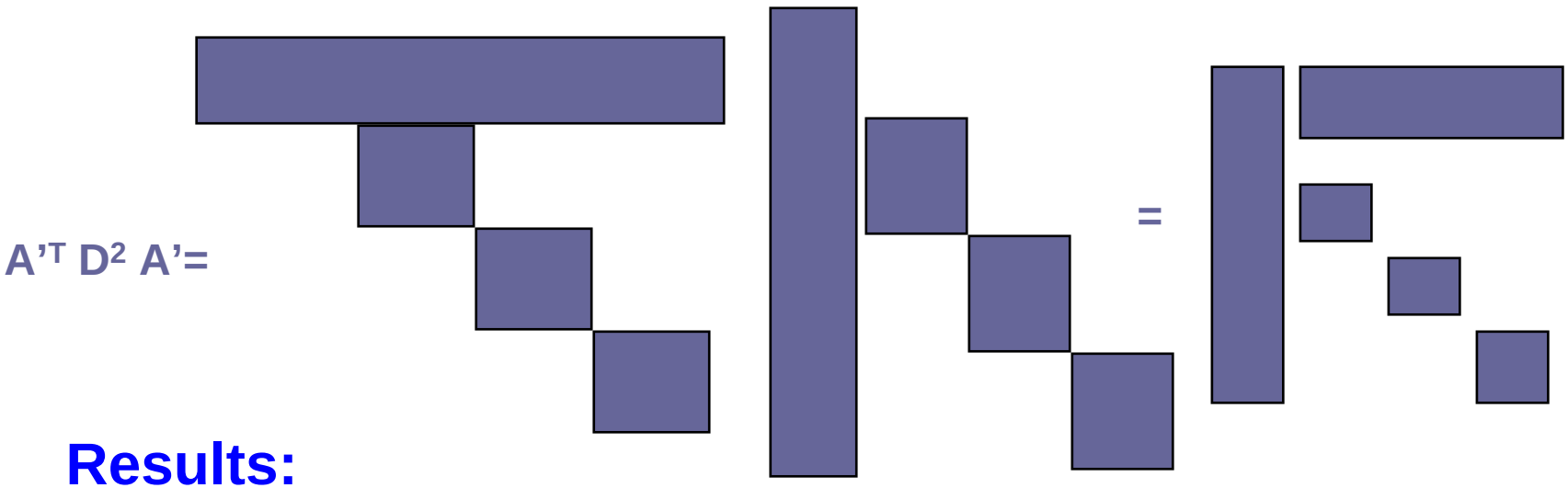
- Variable splitting (Mulvey et al.)
  - Put in explicit nonanticipativity contraints



- **Result**
  - **Reduced fill-in but larger matrix**

Other Interior Point Approaches

- Use of dual factorization or modified schur complement



Results:

- **Speedups of 2 to 20**
- **Some instability => indefinite system (Vanderbei et al. Czyzyk et al.)**

Outline

- General Model – Observations
- Overview of approaches
- Factorization/sparsity (interior point/barrier)
- Decomposition
- Lagrangian and ADP methods
- Conclusions

Similar/Small Problem Structure: Dynamic Programming View

- Stages: $t=1,\ldots,T$
- States: x_t -> $B_t x_t$ (or other transformation)
- Value function:

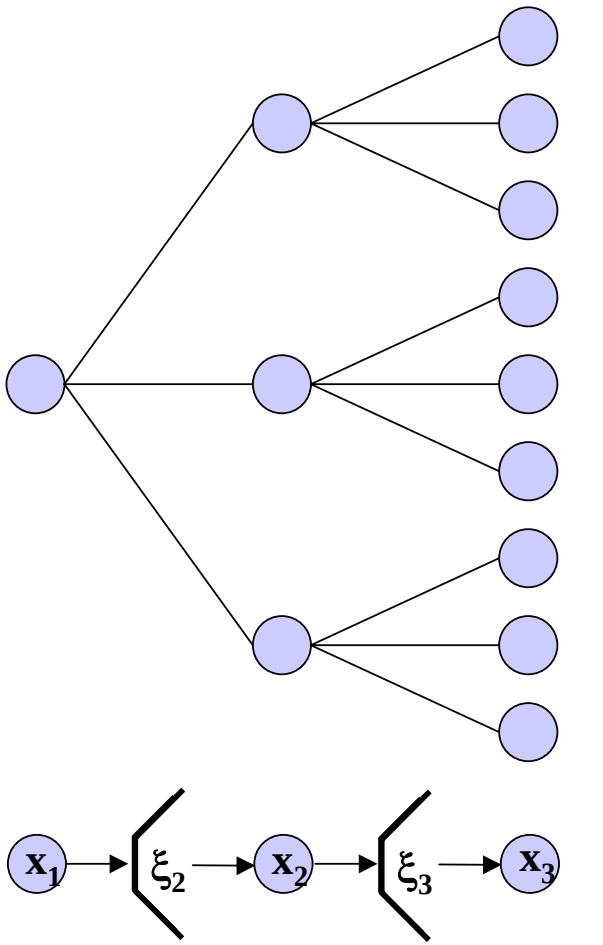
  $Q_t(x_t) = E[Q_t(x_t,\xi_t)]$ where

  ξ_t is the random element and

  $Q_t(x_t,\xi_t) = \min f_t(x_t,x_{t+1},\xi_t) + Q_{t+1}(x_{t+1})$

  $s.t.\ x_{t+1} \in X_{t+1t}(,\xi_t) \quad x_t$ given
- Solve : iterate from T to 1

Linear Model Structure



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

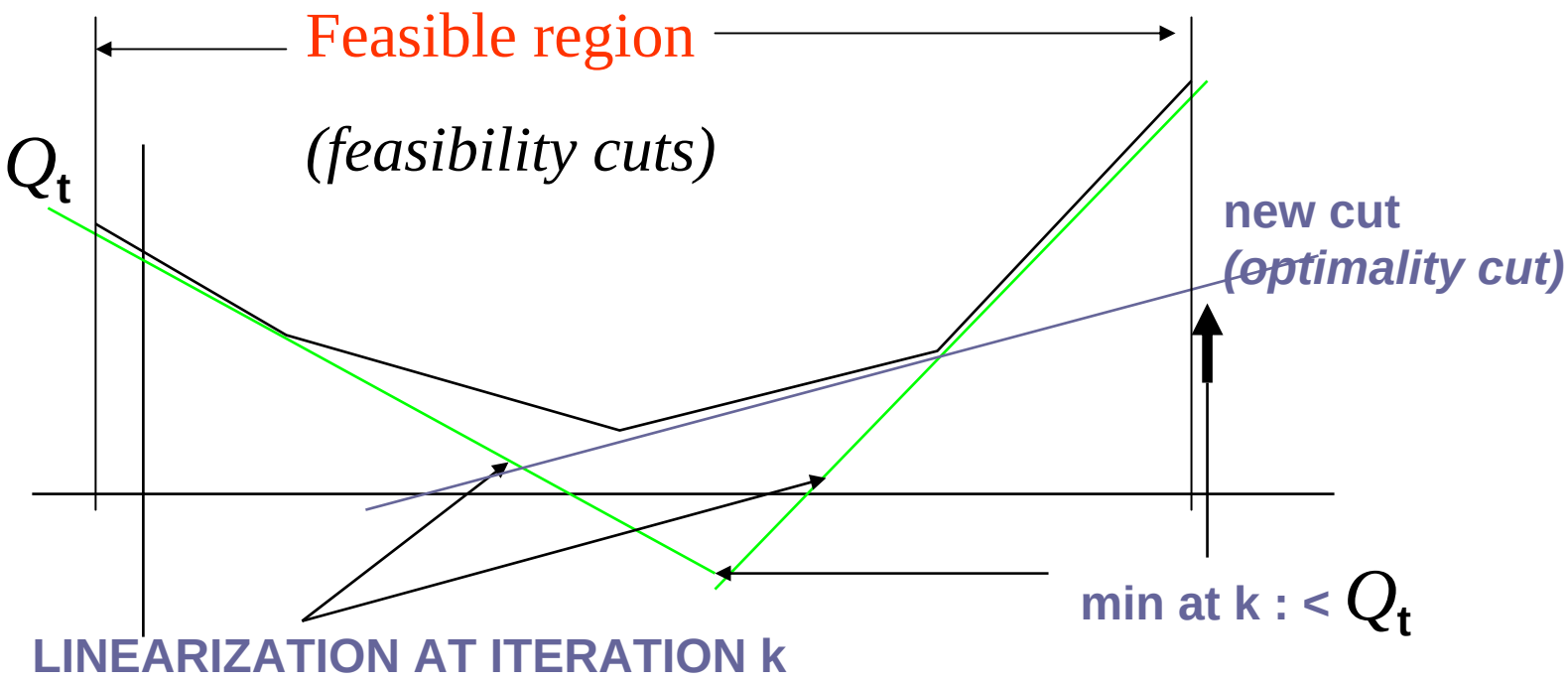
$$Q_t\left(x_{t-1,a(k)}\right) = \sum_{\xi_{t,k} \in \Xi_t} prob\left(\xi_{t,k}\right) Q_{t,k}\left(x_{t-1,a(k)}, \xi_{t,k}\right)$$

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

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

Decomposition Methods

- Benders idea
  - Form an outer linearization of Q_t
  - Add *cuts* on function :



Nested Decomposition

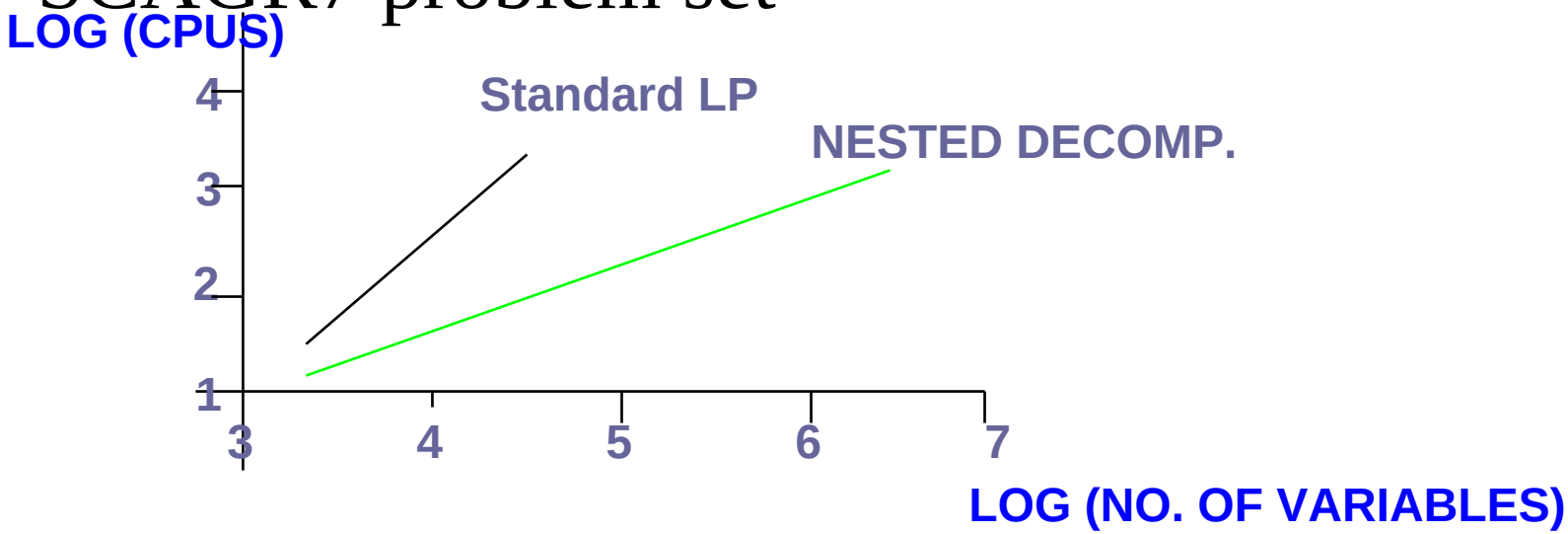
- In each subproblem, replace expected recourse function $Q_{t,k}(x_{t-1,a(k)})$ with unrestricted variable $\theta_{t,k}$
  - Forward Pass:
    - Starting at the root node and proceeding forward through the scenario tree, solve each node subproblem

$$\hat{Q}_{t,k}\left(x_{t-1,a(k)},\xi_{t,k}\right)=\begin{array}{lllll}\min & c_t\left(\xi_{t,k}\right)x_{t,k}+\theta_{t,k} & & & \\ s.t. & W_t x_{t,k} & = & h_t\left(\xi_{t,k}\right)-T_{t-1}\left(\xi_{t,k}\right)x_{t-1,a(k)} & \\ & E_{t,k}x_{t,k}+\theta_{t,k} & \geq & e_{t,k} & \left(optimality\ \ cuts\right) \\ & D_{t,k}x_{t,k} & \geq & d_{t,k} & \left(feasibilit\ y\ cuts\right) \\ & x_{t,k} & \geq & 0 & \end{array}$$

    - 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 $\theta_1 = Q_2(x_1)$

Sample Results

- SCAGR7 problem set



PARALLEL: 60-80% EFFICIENCY IN SPEEDUP

Other problems: similar results

- **Only < order of magnitude speedup with STORM**
  - **two-stages - little commonality in subproblems**

Decomposition Enhancements

- Optimal basis repetition
  - Take advantage of having solved one problem to solve others
  - Use *bunching* to solve multiple problems from root basis
  - *Share* bases across levels of the scenario tree
  - Use solution of single scenario as *hot start*
- Multicuts
  - Create cuts for each descendant scenario
- Regularization
  - Add quadratic term to keep close to previous solution
- Sampling
  - Stochastic decomposition (Higle/Sen)
  - Importance sampling (Infanger/Dantzig/Glynn)
  - Multistage (Pereira/Pinto, Abridged ND)

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

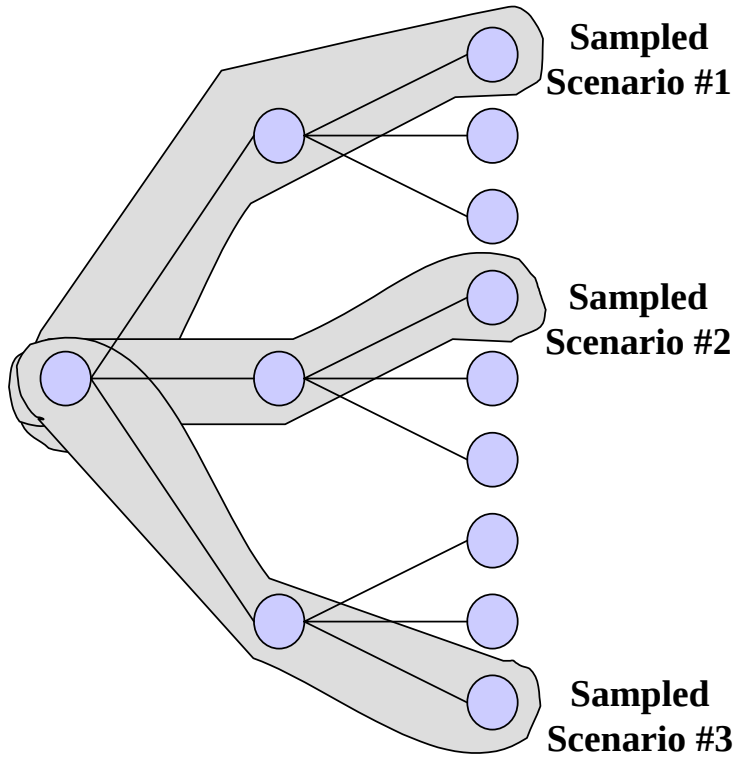
1. Randomly select *H* *N*-Stage scenarios
2. Starting at the root, a forward pass is made through the sampled portion of the scenario tree (solving ND subproblems)
3. A statistical estimate of the first stage objective value is calculated using the total objective value obtained in each sampled scenario

z

the algorithm terminates if current first stage objective value $c_1x_1 + \theta_1$ 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

z



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

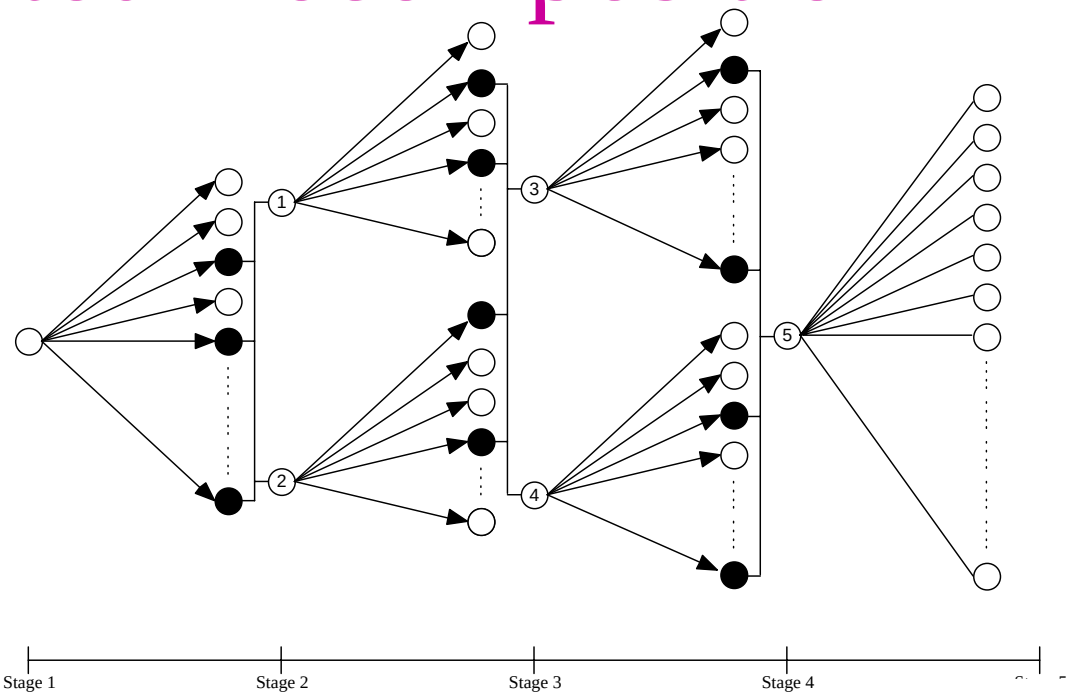
Abridged Nested Decomposition

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)**



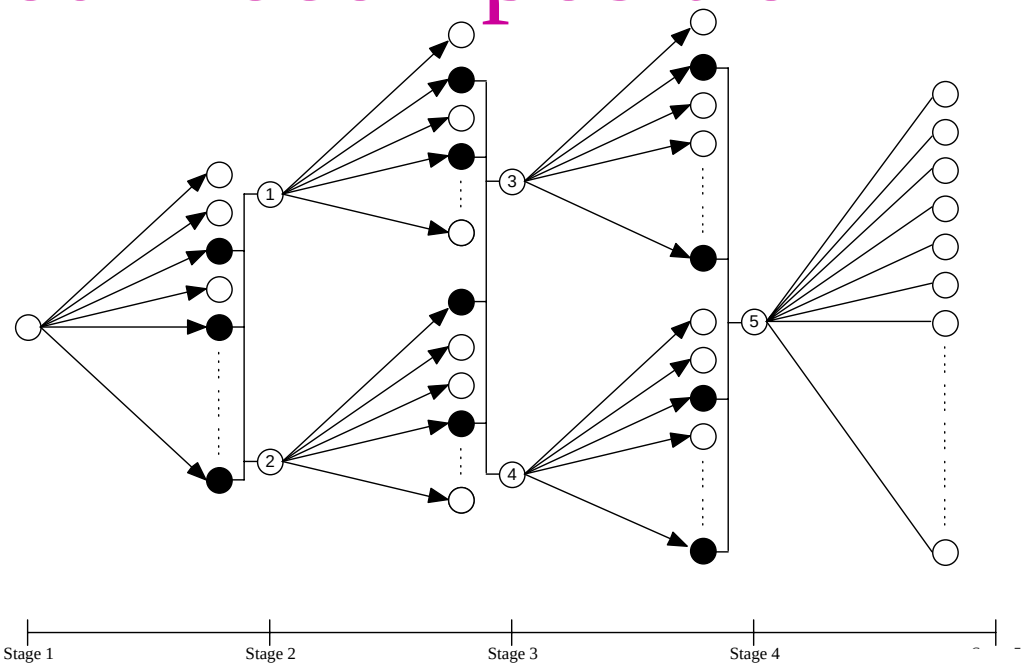
4. For each selected Stage t-1 subproblem solution, sample Stage t subproblems and solve selected subset

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

Abridged Nested Decomposition

Backward Pass

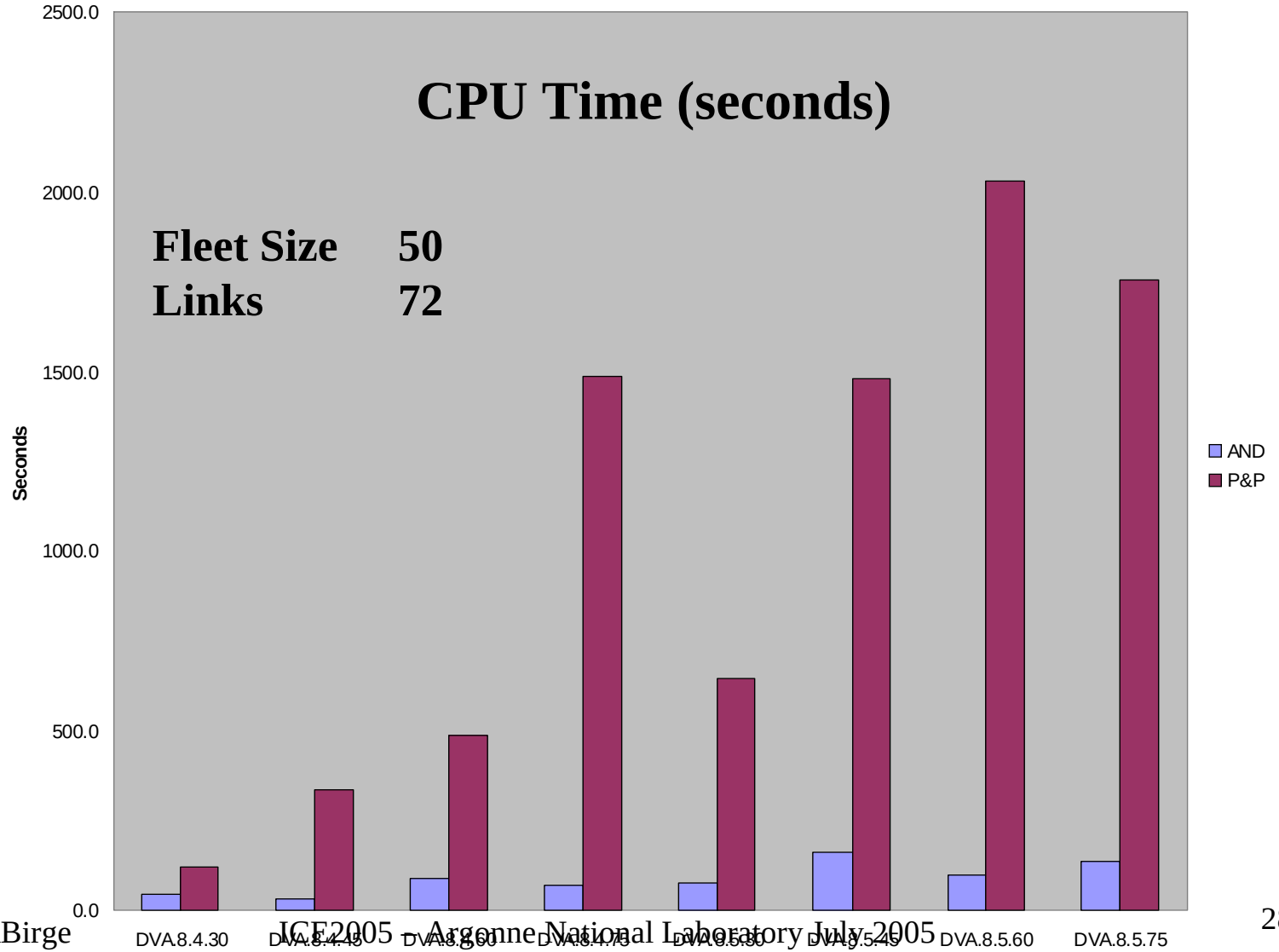
1. **Starting in first branching node of Stage t = N-1, solve all Stage t+1 descendant nodes and construct new optimality cut for all stage t subproblems. Repeat for all sampled nodes in Stage t, then repeat for t = t - 1**



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 $c_1x_1 + \theta_1$ is within a specified confidence interval of $\bar{z}$, else, a new forward pass begins

Computational Results (DVA.8)



What About Infinite Horizon Problems?

- Example: (Very) long-term investor (example: university endowment)
  - Payout from portfolio over time (want to keep payout from declining)
  - Invest in various asset categories
  - Decisions:
    - How much to payout (consume)?
    - How to invest in asset categories?
- How to model?

Infinite Horizon Formulation

- Notation:

  x – current state ($x \in X$)

  u (or u_x) – current action given x (u (or u_x) $\in U(x)$)

  δ – single period discount factor

  $P_{x,u}$ – probability measure on next period state y depending on x and u

  $c(x,u)$ – objective value for current period given x and u

  $V(x)$ – value function of optimal expected future rewards given current state x

- Problem: Find V such that

  $V(x) = max_{u \in U(x)}\{c(x,u) + \delta E_{Px,u}[V(y)]\}$

  for all $x \in X$.

Decomposition (Cutting Plane) Approach

- Define an upper bound on the value function

$$V^0(x) \geq V(x) \;\forall x \in X$$

- Iteration k: upper bound V^k

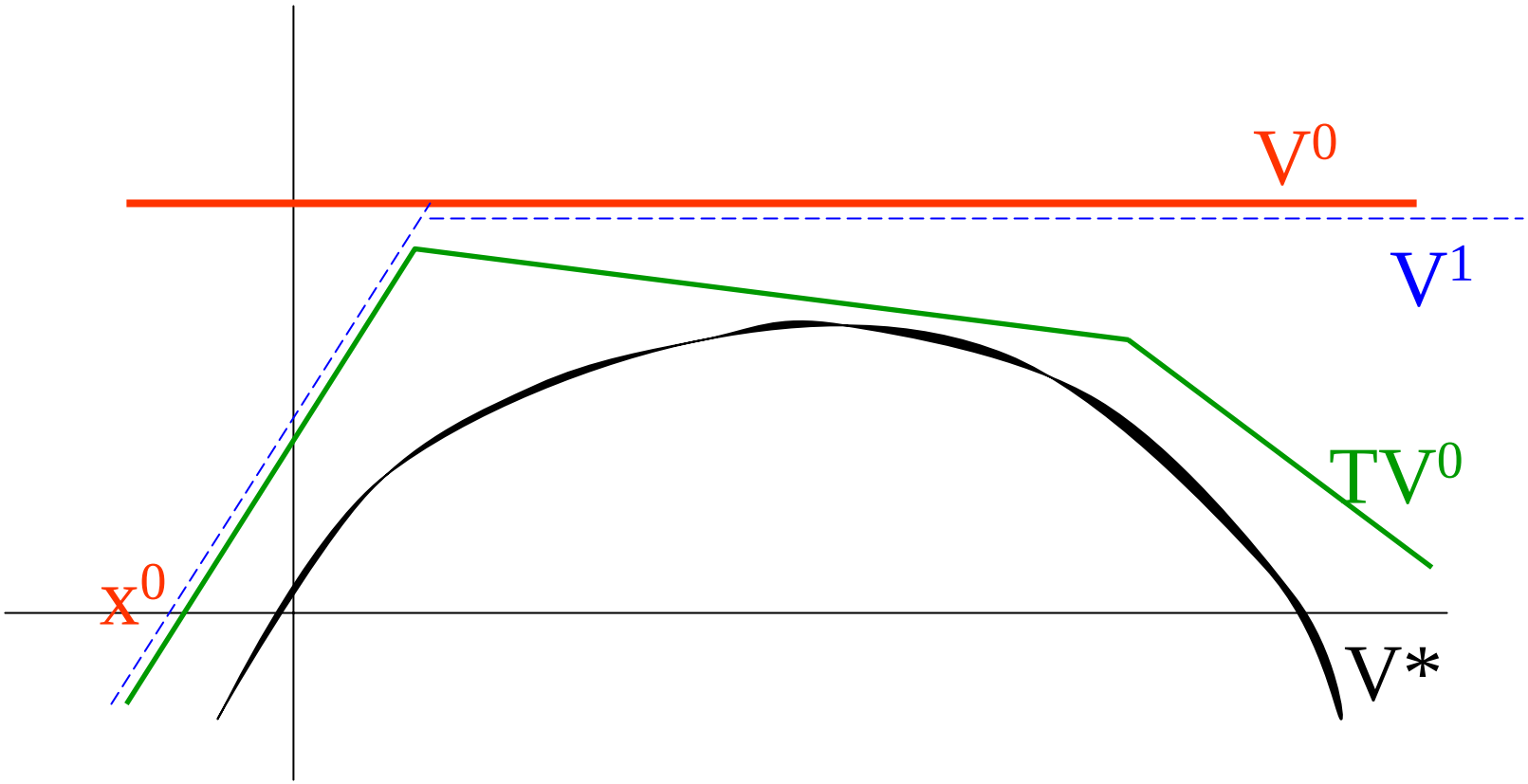
  Solve for some x^k

  $$TV^k(x^k) = max_u\, c(x^k,u) + \delta E_{Pxk,u}[V^k(y)]$$

  Update to a better upper bound V^{k+1}

- Update uses an outer linear approximation on U^k

Successive Outer Approximation



Properties of Approximation

- $V^* \leq TV^k \leq V^{k+1} \leq V^k$
- Contraction

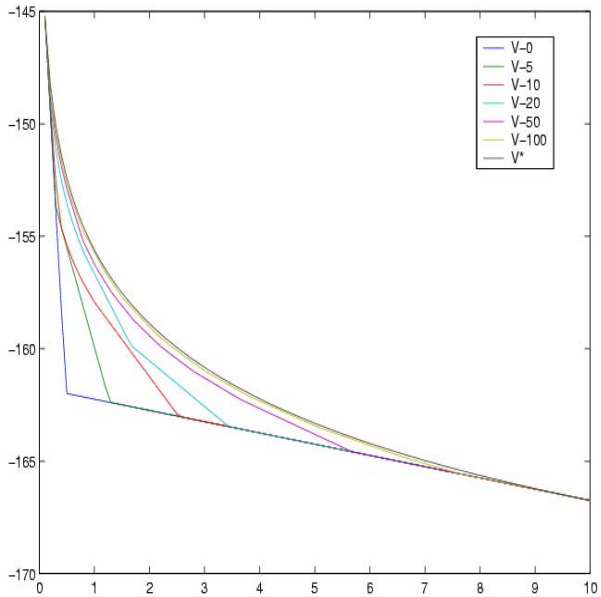
  $|| TV^k - V^* ||_\infty \leq \delta ||V^k - V^*||_\infty$
- Unique Fixed Point

  $TV^* = V^*$

$\Rightarrow$ if $TV^k \geq V^k$, then $V^k = V^*$.

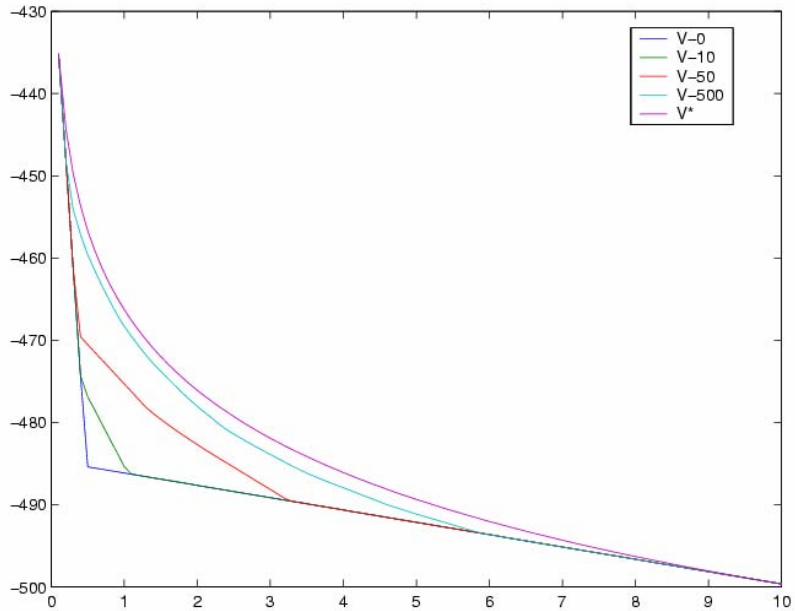
Results

- Sometimes convergence is fast



Results

- Sometimes convergence is slow



Outline

- General Model – Observations
- Overview of approaches
- Factorization (interior point/barrier)
- Decomposition
- Lagrangian and ADP methods
- Conclusions

Lagrangian-based Approaches

- General idea:
  - Relax nonanticipativity (or perhaps other constraints)
  - Place in objective
  - Separable problems

$$\text{MIN} \quad E\left[\sum_{t=1}^{T} f_t(x_t, x_{t+1})\right] \quad \text{s.t.} \quad x_t \in X_t, \; x_t \text{ nonanticipative}$$

$$\rightarrow \quad \text{MIN} \quad E\left[\sum_{t=1}^{T} f_t(x_t, x_{t+1})\right] + E[\underline{w}_t x] + r/2\|x - \underline{x}\|^2 \quad x_t \in X_t$$

Update: w_t; **Project**: x into N - nonanticipative space as $\underline{x}$

Convergence: Convex problems - Progressive Hedging Alg. (Rockafellar and Wets)

Advantage: Maintain problem structure (e.g., network)

Lagrangian Methods and Integer Variables

- Idea: Lagrangian dual provides bound for primal but
  - Duality gap
  - PHA may not converge
- Alternative: Standard augmented Lagrangian
  - Convergence to dual solution
  - Less separability
  - May obtain simplified set for branching to integer solutions
- Problem structure: Power generation problems
  - Especially efficient on parallel processors
  - Decreasing duality gap in number of generation units

Approximate Dynamic Programming: Infinite Horizon

- Use LP solution of dynamic (Bellman) equation:

  max (d,V) s.t. $TV \geq V$ for distribution d on x

- Approximate V with finite set of basis functions Φ_j, weights λ_j
- LP for finite set becomes: Find λ to

  max $(d,\Phi\lambda)$ s.t. $T\Phi\lambda \geq \Phi\lambda$

Solving ADP Form

- Bounds available (Van Roy, De Farias)
- Discretizations:
  - Discrete state space x
  - Use structure to reduce constraint set
- Use Duality:
  - Dual Form:

$\min_{\mu} \max_{\lambda} (d, \Phi\lambda) + (\mu, T\Phi\lambda - \Phi\lambda)$

Combine with outer approximation?

Outline

- General Model – Observations
- Overview of approaches
- Factorization (interior point/barrier)
- Decomposition
- Lagrangian and ADP methods
- Conclusions

Some Open Issues

- Models
  - Impact on methods
  - Relation to other areas
- Approximations
  - Use with sampling methods
  - Computation constrained bounds
  - Solution bounds
- Solution methods
  - Exploit specific structure
  - Links to approximations

Criticisms

- Unknown costs or distributions
  - Find all available information
  - Can construct bounds over all distributions
    - Fitting the information
  - Still have known errors but alternative solutions
- Computational difficulty
  - Fit model to solution ability
  - Size of problems increasing rapidly

Conclusions

- Stochastic programs structure:
  - Repeated problems
  - Nonzero pattern for sparsity
  - Use of decomposition ideas
- Results
  - Take advantage of the structure
  - Speedups of orders of magnitude