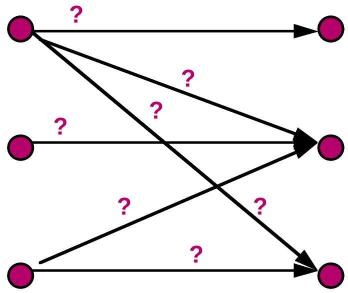
Methods for Stochastic Dynamic Network Problems

John R. Birge
Christopher J. Donohue
University of Michigan



Slide Number 1

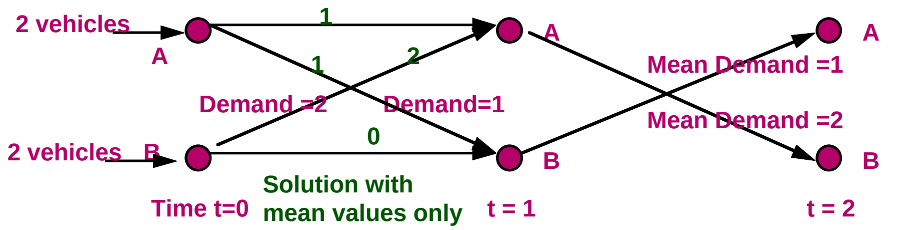
Outline

- Example - Vehicle Allocation
- Formulation
- General Approximations
- New Bounds using Structure
- Computational Methods
  - Stochastic Out-of-Kilter
  - Decomposition implementation
- Results
- Conclusions

Slide Number 2

Example: Vehicle Allocations

- **GOAL:** maximize revenue from loads carried (minus costs of all movements)
- **DECISIONS:** Number of vehicles to move between each pair of locations at each time (may be empty or loaded if demand is sufficient)



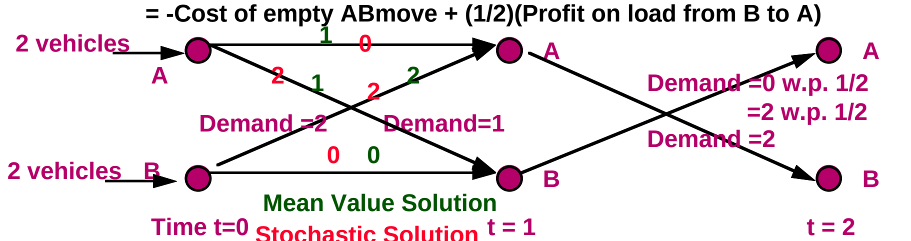
Slide Number 3

Value of the Stochastic Solution

- **Problems with Mean Value Solution:**
  - Demand from B to A may have mean 1 but may be 0 with prob. 1/2 and 2 with prob. 1/2
  - Leaving one vehicle at A means that a second load is lost with prob. 1/2
  - Moving a second vehicle from A to B at time 0 yields:

Value of the Stochastic Solution

= -Cost of empty ABmove + (1/2)(Profit on load from B to A)



Slide Number 4

Formulation

- **FORMULATION:**

  Min $E_s[\Sigma_t [-r(t)u(t,l,s)+p(t)u(t,e,s)]\ \ ]$

s.t. $x(t,s)+ T(u(t,l,s)+u(t,e,s)) = x(t+1,s)$

  $u(t,l,s) \leq d(s)$ (random demand)

  $u,x \geq 0$

  u,x are NONANTICIPATIVE (cannot depend on future)

  (ALSO INTEGRAL)

- **x - state (no. in each location)**
- **u - control (components ij) with l for loaded, e for empty**
- **s - demand scenario**

Slide Number 5

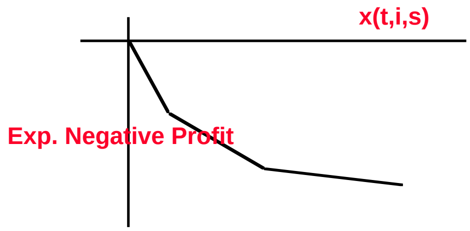
Dynamic Programming View

- **STAGES: t=1,...,T**
- **STATES: x_t (or other transformation)**
- **Value Function:**
  - ∠ $\Psi_t(x_t) = E[\psi_t(x_t,\xi_t)]$ where
  - ∠ ξ_t is the random element and
  - ∠ $\psi_t(x_t,\xi_t) = \min f_t(x_t,x_{t+1},\xi_t) + \Psi_{t+1}(x_{t+1})$
  - – s.t. $x_{t+1} \in X_{t+1t}(,\xi_t)$ x_t given
- **Frequent Assumptions:**
  - Convexity
  - Relaxed integrality
- **Basic Problem: how to evaluate $\Psi_t(x_t)$?**

Slide Number 6

Common Approaches

- **Use of convexity:**
  - Bounds in general: Jensen (lower), Edmundson-Madansky (upper)
  - Using network structure: Powell, Frantzeskakis, Cheung
- **Basic idea: at each node i approximate the expected revenue as a function of the state, x(t,i,s)**



Slide Number 7

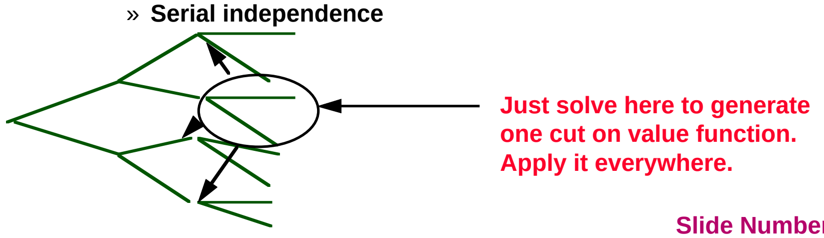
Problems in Previous Approaches

- **Lower Bounds:**
  - Generally separable - losing interaction effects
  - Accuracy overall
- **Upper Bounds**
  - Inefficient (exponential in number of arcs)
  - Inaccurate in general
- **Objectives in new bounds**
  - Include interactions
  - Increase efficiency and accuracy in upper bound

Slide Number 8

Lower Bounding Procedure

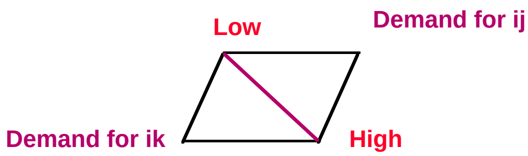
- **Idea:**
  - Use cut generation as in decomposition approaches to generate higher dimensional approximation of value function graph
  - Solve large system from each node
  - Use new assumptions:
    - » Relatively complete recourse (can always find feasible solution in future)
    - » Serial independence



Slide Number 9

Upper Bounding Procedure

- **Idea: Use a property called: convex marginal returns**
  - occurs if the marginal return from any individual state element cannot increase if some other element is increased
  - holds for all arcs with common end-nodes
- **Result: An upper bound is obtained by combining results from all high and all low demands**



Slide Number 10

Upper Bounding Results

- **Transportation and vehicle allocation problems of varying sizes (sample)**

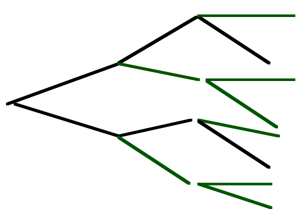
No of Random Arcs	Nodes	UB-LB/LB(%)
105	15	2.2
30	9	0.5
46	14	10.5
15	8	2.3

- **Extensions: Can use the method in constructing a stochastic out-of-kilter method**
- Problem: Exponent is reduced from arcs to nodes but still exponential no. of solutions to consider

Slide Number 11

Alternative: Decomposition and Sampling

- **Use procedure (modified) from Pereira and Pinto**
- **Observation:**
  - **As long as scenarios are chosen to approach true probability distribution, the sample mean of the sum of one period values provides an asymptotic upper bound**
  - **Only need to generate branches - not the whole tree**



Slide Number 12

Decomposition Computational Results

- **Small sample problems (up to 10,000 variables)- CPUs**

Variables	Nested Decomp.	Det. Eq.	Stoch. Out-of-Kilter
460	1.6	1.2	13.0
790	2.1	2.4	42.4
1450	2.9	6.8	122.8
2800	3.4	20.9	477.8
5380	2.1	75.2	1778.1
6600*	12.6	278.4	8423.2

*** different problem form.**

- **Solutions include problems with 10+ million variables**

Slide Number 13

Conclusions

- **Value:** stochastic models can have significant value for a solution
- **Structure:** stochastic networks have structure that enables efficiency in bounding and solution procedures
- **Computation:** Decomposition procedures using the problem structure have achieved significant efficiencies over direct deterministic approaches

http://www-personal.engin.umich.edu/~jrbirge

Slide Number 14