A decomposition-based warm-start method for stochastic programming

Marco Colombo   Andreas Grothey

School of Mathematics
University of Edinburgh

ERGO seminar, 4 March 2009

Introduction

Stochastic programming

- Model uncertainty through the analysis of possible future scenarios
- Alternating sequence of decisions and random realisations

Representation of stochasticity via an event tree, in which to each node of the tree we associate:

- a set of constraints
- an objective function
- the conditional probability of visit from the parent node

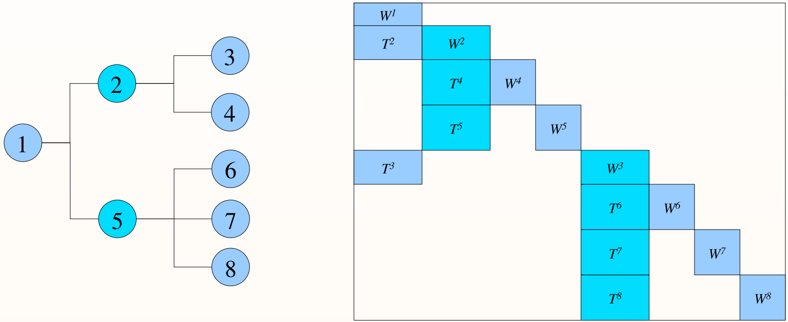
Formulation

A linear stochastic programming problem can be formalised as:

$$\begin{aligned} \min_x \quad & c^\top x + E_\xi[Q(x,\xi)] \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

$$\begin{aligned} Q(x,\xi) = \min_y \quad & q(\xi)^\top y(\xi) \\ \text{s.t.} \quad & T(\xi)x + W(\xi)y(\xi) = h(\xi) \\ & y(\xi) \geq 0 \end{aligned}$$

Structure of the deterministic equivalent



The curse of dimensions

The deterministic equivalent problem has extremely large size, even when starting from a small core.

Example: fxm	rows	cols	nonzeros
Deterministic model:	330	457	2,566
3 stages, 6 nodes:	6,200	9,492	54,589
4 stages, 16 nodes:	386,940	517,282	4,518,039

We can exploit the matrix/tree structures:

- Linear algebra: structure-exploiting parallel software OOPS
- Algorithmically: warm-start for stochastic problems in IPMs

Linear programming and optimality conditions

Karush-Kuhn-Tucker (KKT) conditions for optimality for an LP:

$$\begin{array}{rcl} Ax - b & = & 0 \\ A^\top y + s - c & = & 0 \\ \forall i: \; x_i s_i & = & 0 \\ x, s & \geq & 0 \end{array} \quad \Rightarrow \quad \begin{array}{c} \begin{bmatrix} Ax - b \\ A^\top y + s - c \\ XSe \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ x, s \geq 0 \end{array}$$

Linear programming and optimality conditions

Karush-Kuhn-Tucker (KKT) conditions for optimality for an LP:

$$\begin{aligned} Ax - b &= 0 \\ A^\top y + s - c &= 0 \\ \forall i:\ x_i s_i &= \mu \\ x, s &\geq 0 \end{aligned} \quad \Rightarrow \quad \begin{bmatrix} Ax - b \\ A^\top y + s - c \\ XSe \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mu e \end{bmatrix} \quad x, s \geq 0$$

IPMs perturb the complementarity conditions and solve a sequence of problems parametrised by μ.

As $\mu \to 0$ the solution traces a continuous path from the starting point to the optimal solution (central path).

Optimal partition

Consider a solution (x^*, y^*, s^*): by complementarity

$$x_i^* s_i^* = 0$$

Define two index sets:

$$\mathcal{B} = \{i : x_i^* \neq 0\}, \quad \mathcal{N} = \{i : s_i^* \neq 0\}$$

In interior point methods:

$$x_i^* \to 0 \quad \text{and} \quad s_i^* \to \hat{s}_i > 0$$

$$s_i^* \to 0 \quad \text{and} \quad x_i^* \to \hat{x}_i > 0$$

The objective of the algorithm is to discover the optimal partition. Keeping centrality is essential to avoid approaching the wrong partitioning.

Evolution of the partitioning

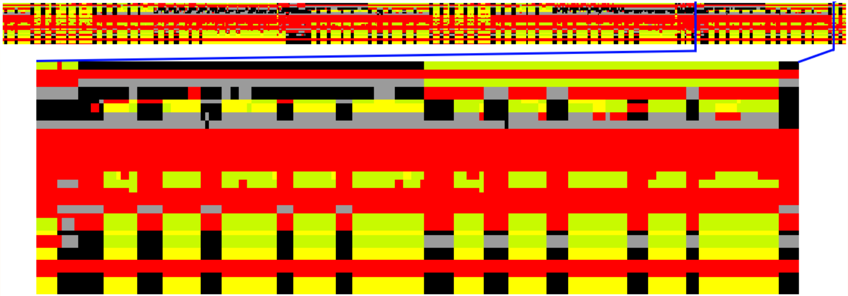
Tapia indicator:

$$\frac{x_i^{k+1}}{x_i^k} \rightarrow \begin{cases} 1 & \text{if } x_i \rightarrow \hat{x}_i > 0 \\ 0 & \text{if } x_i \rightarrow 0 \end{cases} \qquad 1 - \frac{s_i^{k+1}}{s_i^k} \rightarrow \begin{cases} 1 & \text{if } s_i \rightarrow 0 \\ 0 & \text{if } s_i \rightarrow \hat{s}_i > 0 \end{cases}$$

Evolution of the partitioning

Tapia indicator:

$$\frac{x_i^{k+1}}{x_i^k} \to \begin{cases} 1 & \text{if } x_i \to \hat{x}_i > 0 \\ 0 & \text{if } x_i \to 0 \end{cases} \qquad 1 - \frac{s_i^{k+1}}{s_i^k} \to \begin{cases} 1 & \text{if } s_i \to 0 \\ 0 & \text{if } s_i \to \hat{s}_i > 0 \end{cases}$$



Warm-start strategies

Use the solution to a problem instance to initialise the next.

Warm-start issues with IPMs:

- Point should be close to the solution
- Point should be away from the boundary

Current attempts:

- Store an “advanced” iterate (3–4 digits of accuracy)
- Take special care of centrality
- Restore primal and dual feasibility with independent directions
- Allow the iterates to become negative (with penalties)

Warm-start strategies for stochastic linear programming

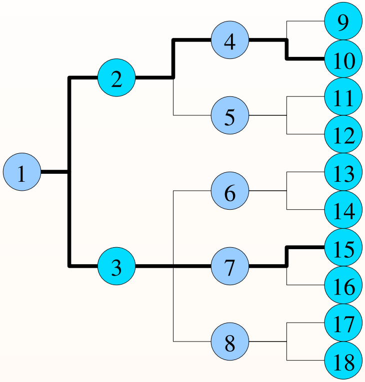
Problem setup:

- Required to solve an instance with a specific tree
- We generate and solve the deterministic equivalent
- Stochastic problems are given in SMPS format

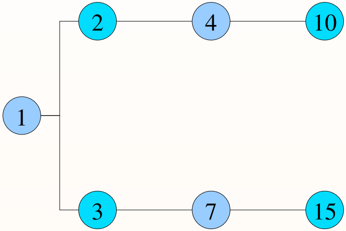
Strategies:

- Generate a reduced tree and use the smaller problem to generate a warm-start point
- Decompose the problem at the second stage, solve the subproblem independently and use their solution to generate a warm-start point

Scenario reduction



Complete tree



Reduced tree

Main steps of the algorithm

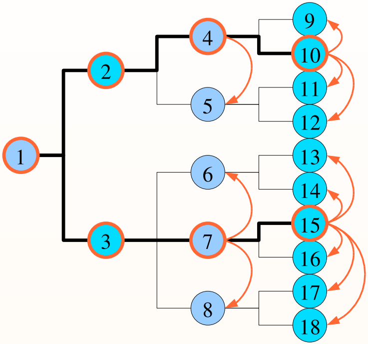
Exploit the structure of the stochastic program:

1. Find a reduced event tree
2. Solve the reduced deterministic equivalent with loose accuracy
3. Generate a warm-start iterate for the complete problem
4. Solve the complete problem to optimality

Features:

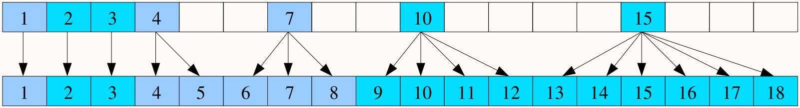
- The reduced problem is very easy to solve
- We exploit the structure to match the dimensions of the two problems

Construction of the warm-start iterate



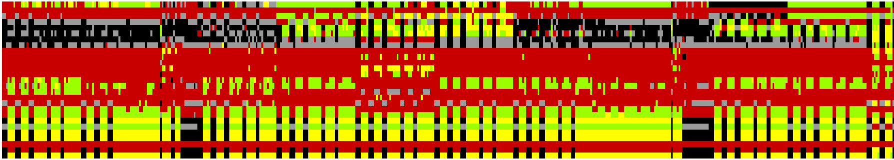
Nodes in the reduced tree:
the solution is already available

Remaining nodes:
copy the solution from the corresponding reduced-tree node

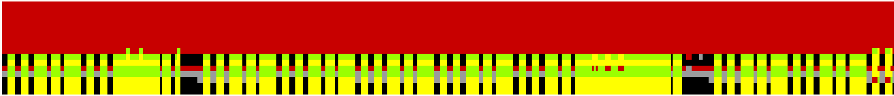


Comparison via Tapia indicators

Cold start (27 iterations)

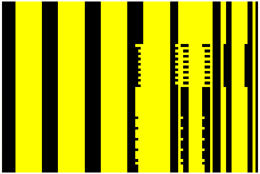


Warm start (16 iterations)



Active set by scenario

stocfor2



pltexpA2-16

fxm2-16

Second-stage problem

Given a first-stage decision x, define the second-stage problem as

$$Q(x, \xi) = \min\{q(\xi)^\top y : Wy = h(\xi) - T(\xi)x,\ y \geq 0\}$$

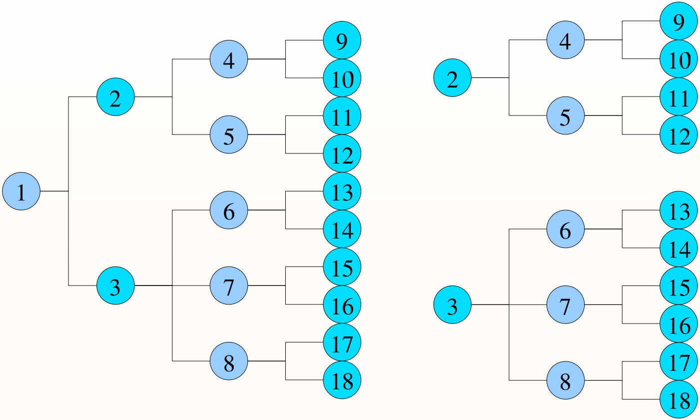
Write a stochastic linear programming problem with recourse:

$$\min E_\xi\{c^\top x + Q(x, \xi)\} \text{ s.t. } Ax = b,\ x \geq 0$$

Observations:

- The first-stage variables link all the blocks.
- For a fixed x, these terms can be eliminated, and we obtain a series of smaller, independent linear programs.

Decomposition



Subproblems

Solve each second-stage problem independently:

$$Q_i(x) = \min\{q_i^\top y_i : Wy_i = h_i - T_i x,\ y_i \geq 0\}$$

$Q_i(x)$ is the objective function value to a subproblem rooted at a second-stage node.

The deterministic equivalent becomes

$$\min\{c^\top x + \sum p_i Q_i(x) : Ax = b, x \geq 0\}$$

Benders Decomposition

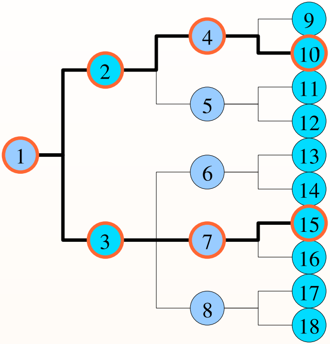
- Find subgradients for $Q_i(x)$ and construct a convex estimate of $Q(x)$.
- Generate cutting planes which are added to the master problem.
- Minimization of $Q(x)$ is recast into a minimization of its piecewise linear approximation.
- When the gap between lower and upper bounds to the solution falls below some preset tolerance, the solution phase stops.

Generating a warm-start point

Our warm-start approach is configured as follows:

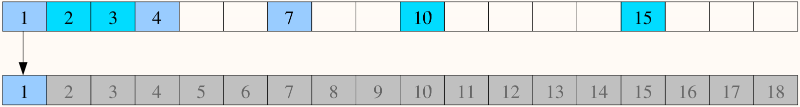
1. Solve a problem on a tree with a small number of scenarios
2. Set up independent subproblems rooted at the second-stage nodes, and solve them
3. Use the solution of the subproblems to initialise the corresponding blocks of the complete solution (this is the warm-start iterate)
4. Solve the complete problem with the warm-start point

Construction of the warm-start iterate

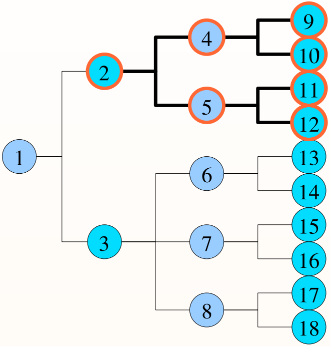


Reduced-tree problem:

Initialise the first-stage variables



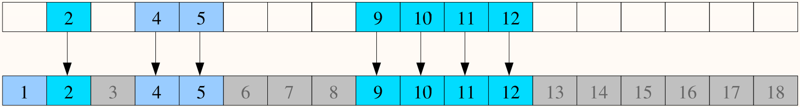
Construction of the warm-start iterate



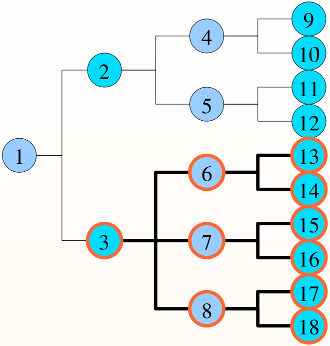
Reduced-tree problem:

Initialise the first-stage variables

First subproblem



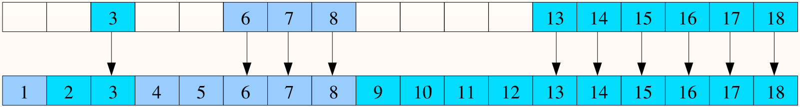
Construction of the warm-start iterate



Reduced-tree problem:
Initialise the first-stage variables

First subproblem

Second subproblem



Computational setup

Reduced problem:

- The reduced problem contains as many scenarios as there are branches at the first stage
- Solved to a tolerance of 5.0e-1 or 5.0e-3

Decomposed subproblems:

- Solved to different tolerances: 5.0e-1, 5.0e-2 or 5.0e-3

Complete problem:

- Solved to a tolerance of 1.0e-7

Computations: Linux PC with a 3.0GHz Pentium processor and 2GB of RAM.

Numerical results

Problem	Cold		Warm	
fxm3-6	24	4.8	19	3.3
fxm3-16	62	62.7	39	44.6
pltexpA4-6	68	47.9	—	—
swing8-4	40	191.3	43	182.8

Problem	dec-1		dec-2		dec-3	
fxm3-6	12	4.7	9	4.4	13	5.3
fxm3-16	76	83.1	21	41.4	22	45.0
pltexpA4-6	47	37.1	—	—	94	74.2
swing8-4	23	182.3	22	182.4	25	194.3

(reduced tol. 5.0-1)

Problem	dec-1		dec-2		dec-3	
fxm3-6	20	5.7	13	5.7	11	4.9
fxm3-16	103	117.4	63	84.9	13	38.4
pltexpA4-6	—	—	—	—	—	—
swing8-4	26	207.6	24	225.3	28	235.1

(reduced tol. 5.0-3)

Conclusions

- Proposed a technique to generate starting points for multi-stage stochastic linear programs
- Savings in iterations but not in computational time (strong problem dependency of the success rate)
- More test problems are needed