A warm-start approach for large-scale stochastic linear programs

Marco Colombo Jacek Gondzio Andreas Grothey

THE UNIVERSITY OF EDINBURGH

School of Mathematics
University of Edinburgh

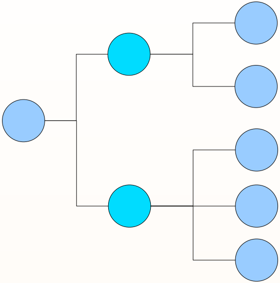
SPXI, Vienna

Introduction

Stochastic programming

- Model uncertainty through the analysis of possible future scenarios
- Alternating sequence of decisions and random realisations
- Robust decision making
- And much more!

Event tree



To each node of the tree we associate:

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

Notation

t stage

l_t index of a node of stage t

$a(l_t)$ ancestor of node l_t

n^{l_t} node data: $\{T^{l_t}, W^{l_t}, h^{l_t}, q^{l_t}, p^{l_t}\}$

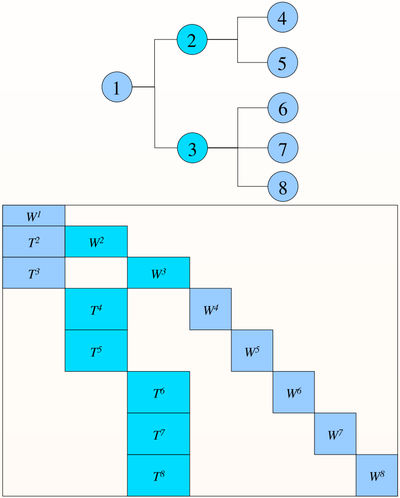
Model of the dynamics of the system (at node l_t):

$$\begin{aligned} \min \quad & \sum_{l_t} p^{l_t} (q^{l_t})^T x^{l_t} \\ \text{s.t.} \quad & T^{l_t} x^{a(l_t)} + W^{l_t} x^{l_t} = h^{l_t} \\ & x^{l_t} \geq 0 \end{aligned}$$

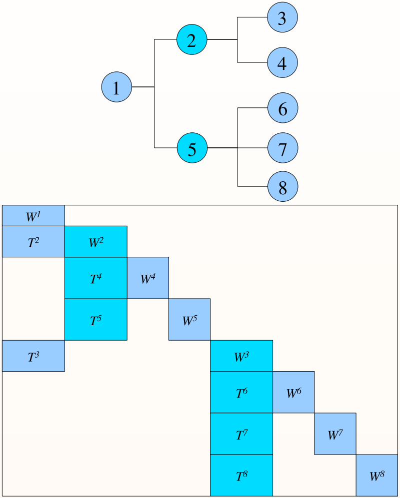
Complete deterministic equivalent formulation

$$
\begin{array}{llcll}
\min & (q^{l_1})^T x^{l_1} & + \displaystyle\sum_{l_2=L_1+1}^{L_2} p^{l_2}(q^{l_2})^T x^{l_2} & + \ldots + \displaystyle\sum_{l_T=L_{T-1}+1}^{L_T} p^{l_T}(q^{l_T})^T x^{l_T} & \\
\text{s.t.} & & W^{l_1} x^{l_1} = h^{l_1} & & \\
& T^{l_2} x^1 & + \quad W^{l_2} x^{l_2} = h^{l_2} & & l_2 = L_1+1, \ldots, L_2, \\
& & \vdots \qquad\qquad \vdots & & \\
& T^{l_T} x^{a(l_T)} & + \quad W^{l_T} x^{l_T} = h^{l_T} & & l_T = L_{T-1}+1, \ldots, L_T, \\
& & x^{l_t} \geq 0 & & l_t = 1, \ldots, L_T.
\end{array}
$$

Structure of the deterministic equivalent



Breadth-first ordering



Depth-first ordering

The curse of dimensions

The deterministic equivalent formulation produces problems of 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

- A detailed description produces robust decisions
- Detailed event trees can be very large
- The dimensions involved explode

However, remember the presence of structure!

The way forward

Enter interior point methods:

- IPM solvers are available in the community (CPLEX Barrier, PCx, HOPDM, etc.)
- Competitiveness of IPMs grows with the problem size
- Parallel implementations are possible

And we can exploit the structure:

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

OOPS - Object Oriented Parallel (Interior Point) Solver

OOPS is a parallel IPM LP/QP solver (with NLP extensions) that can exploit the structure in the linear algebra.

Key advantages of exploiting the structure in the problem:

- Faster linear algebra
- Reduced memory use (by use of implicit factorization)
- Possibility to exploit (massive) parallelism
- Assumption that the structure is known

Talk by Andreas Grothey in session FA2.

Linear programming and optimality conditions

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

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

Linear programming and optimality conditions

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

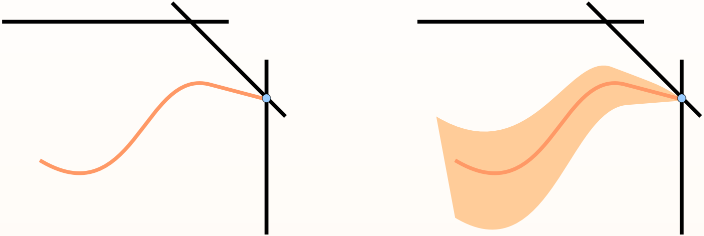
$$\begin{aligned} Ax - b &= 0 \\ A^T y + s - c &= 0 \\ \forall i: \; x_i s_i &= \mu \\ x, s &\geq 0 \end{aligned} \quad \Rightarrow \quad \begin{bmatrix} Ax - b \\ A^T y + s - c \\ XSe \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mu e \end{bmatrix} \quad x, s \geq 0$$

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

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

Centrality

IPMs follow the central path to find the optimal solution.

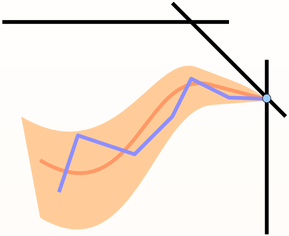


Polynomial complexity:

in theory: $\mathcal{O}(\sqrt{n})$ or $\mathcal{O}(n)$ iterations

in practice: $\mathcal{O}(\ln n)$ iterations

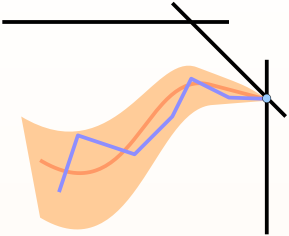
Good behaviour and bad behaviour



Good:

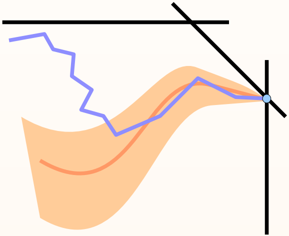
- central starting point
- remain in the neighbourhood of the central path in all iterations

Good behaviour and bad behaviour



Good:

- central starting point
- remain in the neighbourhood of the central path in all iterations



Bad:

- iterate close to the boundary
- many iterations spent in retrieving centrality before converging

Warm-start strategies

A warm-start strategy uses the solution to a problem instance to initialise the next problem.

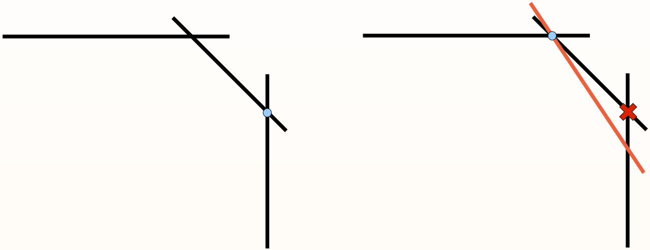
- Important if we are solving a sequence of problems
- The solution to one problem is close to the solution of the next
- Reduced computational time from an advanced starting point

Common understanding:

- Warm-start is good with the simplex method
- Warm-start is bad with IPMs (?)

Warm-start with the simplex method

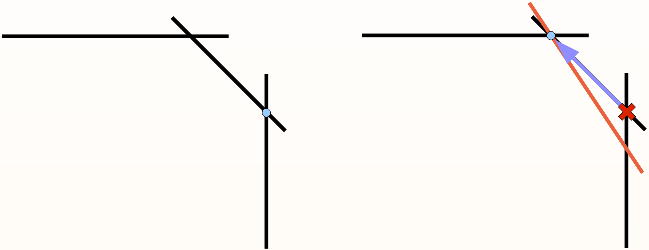
The solution of a problem is a vertex:



- ideal starting point for the modified instance

Warm-start with the simplex method

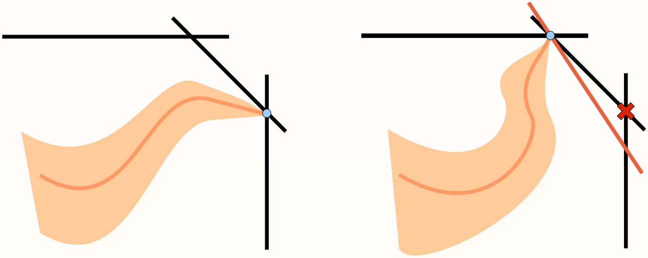
The solution of a problem is a vertex:



- ideal starting point for the modified instance
- optimality recovered in a few (very cheap) iterations

Warm-start with interior point methods

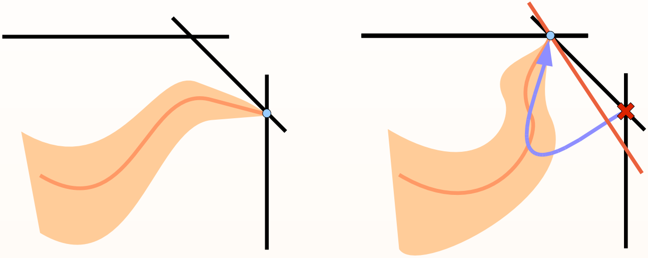
The solution of a problem is arbitrarily close to a vertex:



- worst possible starting point

Warm-start with interior point methods

The solution of a problem is arbitrarily close to a vertex:



- worst possible starting point
- need to recover centrality before attaining optimality

Warm-start issues with IPMs

Contradictory requirements:

- 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)

Assumptions and setup

Main assumptions:

- No knowledge on the underlying stochastic processes
- An event tree is given

Problem setup:

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

Reduced event tree

Observation:

Very detailed event trees provide a fine-grained solution to a problem that could have been solved more coarsely with a much smaller tree.

Idea:

Use the solution to a smaller instance of the problem to generate a warm-start point.

Reduced tree generation

Two complementary strategies:

1. Span the breadth of the tree
   - Choose some of the nodes at stage k (where k is small)
   - Choose all their ancestors up to the root node

Reduced tree generation

Two complementary strategies:

1. Span the breadth of the tree
   - Choose some of the nodes at stage k (where k is small)
   - Choose all their ancestors up to the root node
2. Choose the most representative scenario in each subtree
   - Define a "scenario distance"
   - Minimize the distance from an average scenario

Scenario distance and representative scenarios

Distance between two nodes at period t:

$$d(n^{i_t}, n^{j_t}) = \|T^{i_t} - T^{j_t}\| + \|W^{i_t} - W^{j_t}\| + \|h^{i_t} - h^{j_t}\| + \|q^{i_t} - q^{j_t}\|$$

Scenario distance and representative scenarios

Distance between two nodes at period t:

$$d(n^{i_t}, n^{j_t}) = \|T^{i_t} - T^{j_t}\| + \|W^{i_t} - W^{j_t}\| + \|h^{i_t} - h^{j_t}\| + \|q^{i_t} - q^{j_t}\|$$

Distance between two scenarios:

$$D(s_i, s_j) = \sum_{t=1}^{T} d(n^{i_t}, n^{j_t}), \quad i_t \in s_i,\ j_t \in s_j$$

Scenario distance and representative scenarios

Distance between two nodes at period t:

$$d(n^{i_t}, n^{j_t}) = \|T^{i_t} - T^{j_t}\| + \|W^{i_t} - W^{j_t}\| + \|h^{i_t} - h^{j_t}\| + \|q^{i_t} - q^{j_t}\|$$

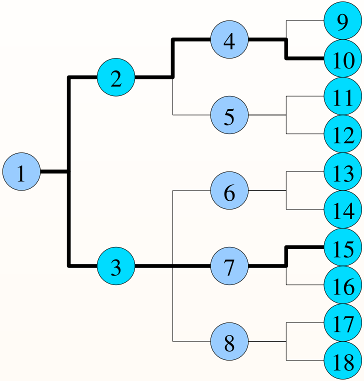
Distance between two scenarios:

$$D(s_i, s_j) = \sum_{t=1}^{T} d(n^{i_t}, n^{j_t}), \quad i_t \in s_i,\ j_t \in s_j$$

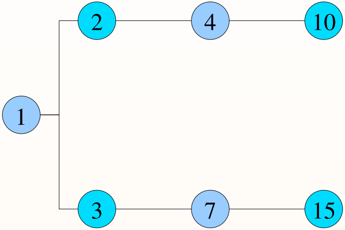
Representative scenario s^* is the one that minimizes the weighted distance from an average scenario $\bar{s}$:

$$s^* = s_k, \quad k = \arg\min_{i \in S} (1 - p_i) D(s_i, \bar{s})$$

Scenario reduction



Complete tree



Reduced tree

Main steps of the algorithm

Exploit the structure of the stochastic program:

1. Find a reduced event 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

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

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

Main steps of the algorithm

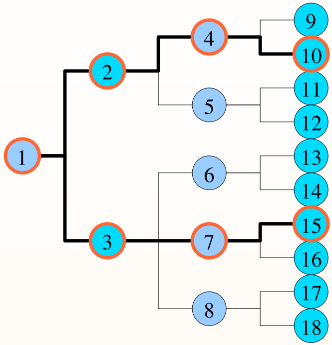
Exploit the structure of the stochastic program:

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:

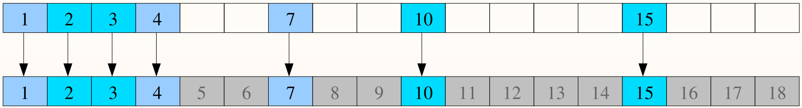
- The reduced problem is very easy to solve
- We exploit the structure to match the dimensions of the two problems

Construction of the warm-start iterate

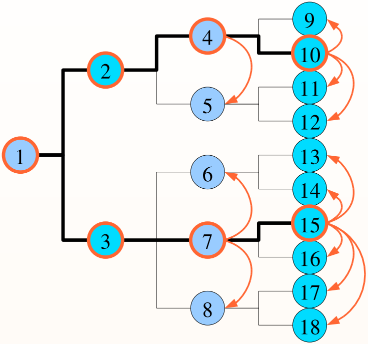


Nodes in the reduced tree:

the solution is already available

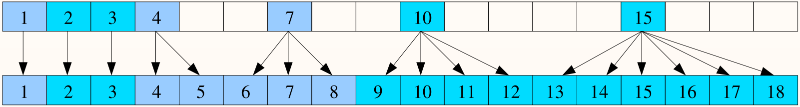


Construction of the warm-start iterate



Nodes in the reduced tree:
the solution is already available

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



Numerical results I

Collection of standard SMPS problems solved with HOPDM:

- 2 scenarios in the reduced tree
- Reduced problem optimality tolerance: 5.0×10^{-1}
- Complete problem optimality tolerance: 5.0×10^{-8}

Numerical results with HOPDM

Problem data			Cold start		Warm start	
Name	Stgs	Scens	Iters	Seconds	Iters	Seconds
fxm2-16	2	16	22	1.2	13	1.0
fxm3-6	3	36	30	1.5	17	1.3
fxm3-16	3	256	40	31.1	20	20.7
fxm4-6	4	216	30	8.2	22	8.3
fxm4-16	4	4096	41	218.3	27	182.6
pltexpA3-16	3	256	26	153.8	14	87.8
pltexpA4-6	4	216	36	55.8	16	27.5
pltexpA5-6	5	1296	81	772.0	30	311.5
storm27	2	27	41	95.4	22	53.2
storm125	2	125	73	107.3	36	69.1
storm1000	2	1000	107	1498.3	45	831.5

Capacity assignment problem with uncertain demand

$$\min_x \; E_d[f(x,d)] \quad \text{s.t.} \quad \sum_{l\in\mathcal{A}} c_l x_l \le M, \;\; x \ge 0,$$

$$\begin{aligned} f(x,d) = \min \quad & \sum_{k\in\mathcal{D}} (d_k - \sum_{p\in P_k} z_p) \\ \text{s.t.} \quad & \sum_{k\in\mathcal{D}} \sum_{p\in\mathcal{P}_k : l\in p} z_p \le x_l \quad && \forall l \in \mathcal{A} \\ & \sum_{p\in\mathcal{P}_k} z_p \le d_k && \forall k \in \mathcal{D} \\ & z_p \ge 0 \end{aligned}$$

Numerical results II

Problems formulated as SMPS and solved with OOPS:

- 2 scenarios in the reduced tree (serial) or 4 scenarios (parallel)
- Reduced problem optimality tolerance: 5.0×10^{-1}
- Complete problem optimality tolerance: 5.0×10^{-4}

Numerical results with OOPS (serial)

Problem data			Cold start		Warm start	
Name	Stgs	Scens	Iters	Seconds	Iters	Seconds
mnx-200	2	200	13	12.9	7	7.3
mnx-800	2	800	17	58.8	10	39.5
mnx-1600	2	1600	19	131.1	10	68.8
jlg-200	2	200	45	164.9	17	39.5
jlg-800	2	800	27	353.4	10	152.9
jlg-1600	2	1600	32	855.3	13	360.6
mgntA-100	2	100	28	260.0	14	156.2
mgntA-200	2	200	50	877.1	35	690.6
mgntA-400	2	400	40	1470.3	14	572.5
mgntB-100	2	100	23	511.1	14	318.0
mgntB-200	2	200	25	909.4	8	332.4
mgntB-400	2	400	29	2154.5	7	538.1

Numerical results with OOPS (parallel)

Problem data			Cold start		Warm start	
Name	Stgs	Scens	Iters	Seconds	Iters	Seconds
mnx-200	2	200	13	4.6	7	3.5
mnx-800	2	800	17	18.8	10	10.7
mnx-1600	2	1600	19	50.3	10	31.4
jlg-200	2	200	45	49.9	17	20.7
jlg-800	2	800	29	130.5	10	50.1
jlg-1600	2	1600	35	286.1	14	129.7
mgntA-100	2	100	28	76.9	14	51.6
mgntA-200	2	200	50	256.4	34	195.3
mgntA-400	2	400	40	410.9	14	181.6
mgntB-100	2	100	23	137.5	14	103.9
mgntB-200	2	200	25	284.2	8	140.5
mgntB-400	2	400	29	605.5	7	211.6

Conclusions and future work

- Reduced tree solutions contain valuable information to construct a good warm-start iterate for IPMs
- Savings in computational time for all but the smallest instances
- Exploit the knowledge on the underlying stochastic process if available
- Extend the approach to a multi-start procedure

Conclusions and future work

- Reduced tree solutions contain valuable information to construct a good warm-start iterate for IPMs
- Savings in computational time for all but the smallest instances
- Exploit the knowledge on the underlying stochastic process if available
- Extend the approach to a multi-start procedure

Colombo, Gondzio, Grothey,
A warm-start approach for large-scale stochastic linear programs,
Technical Report MS-06-004, School of Mathematics, University of Edinburgh.
Available on optimization-online.org.