Stage aggregation to warm-start interior point methods

Marco Colombo   Andreas Grothey

School of Mathematics
University of Edinburgh

APMOD 2008, Bratislava

Linear programming and optimality conditions

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

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

Linear programming and optimality conditions

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

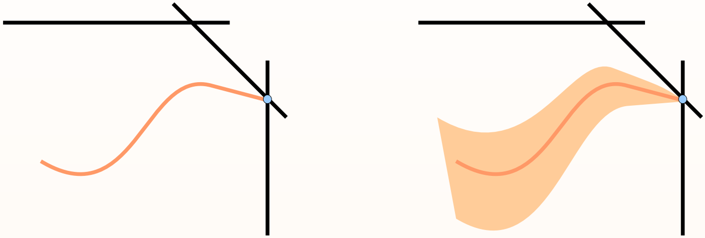
$$\begin{aligned} Ax - b &= 0 \\ A^T y + s - c &= 0 \\ \forall i:\ x_i s_i &= \mu \\ x, s &\geq 0 \end{aligned} \quad \Rightarrow \quad \begin{bmatrix} Ax - b \\ A^T y + s - c \\ XSe \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mu e \end{bmatrix} \quad x, s \geq 0$$

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

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

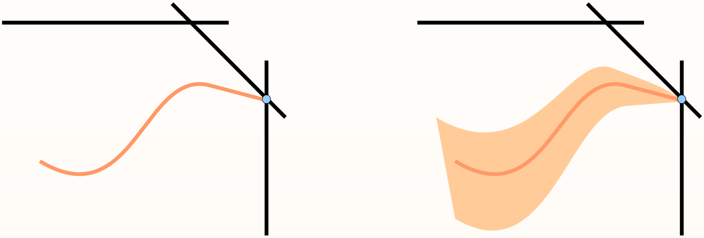
Centrality

IPMs follow the central path to find the optimal solution.



Centrality

IPMs follow the central path to find the optimal solution.

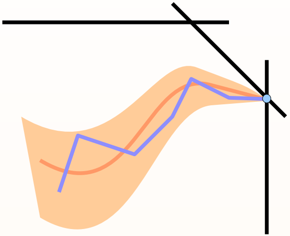


Polynomial complexity:

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

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

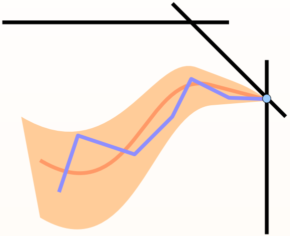
Good behaviour and bad behaviour



Good:

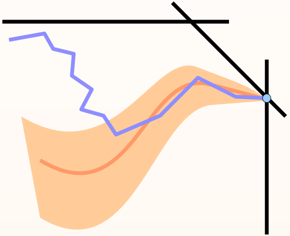
- central starting point
- remain in the neighbourhood of the central path in all iterations

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.

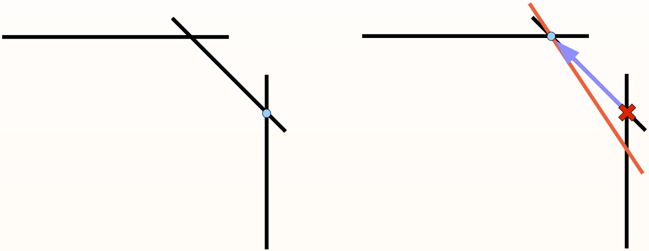
- Important if we are solving a sequence of problems
- The solution to one problem is close to the solution of the next
- Reduced computational time from an advanced starting point

Common understanding:

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

Warm-start with the simplex method

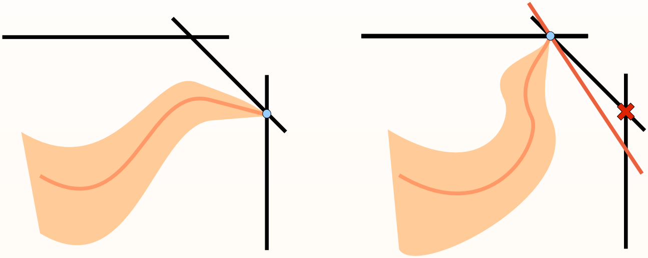
The solution of a problem is a vertex:



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

Warm-start with interior point methods

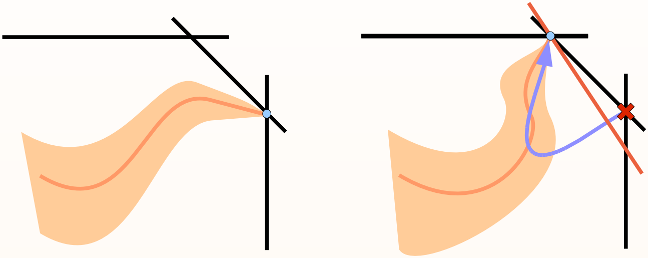
The solution of a problem is arbitrarily close to a vertex:



- 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)

Research

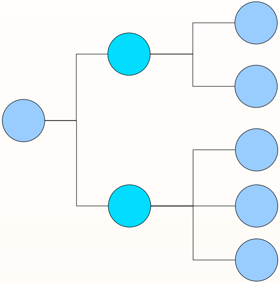
Some of the papers on warm-start for interior point methods:

- Mitchell, Todd '92
- Hipolito '93
- Lustig, Marsten, Shanno '94
- Gondzio '98
- Gondzio, Vial '99
- Yıldırım, Wright '02
- Gondzio, Grothey '03
- John, Yıldırım '06
- Benson, Shanno '06
- Grothey, Gondzio '06

Message:

- Warm-starting IPMs is possible
- Warm-starting can save around 50–60% of iterations

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

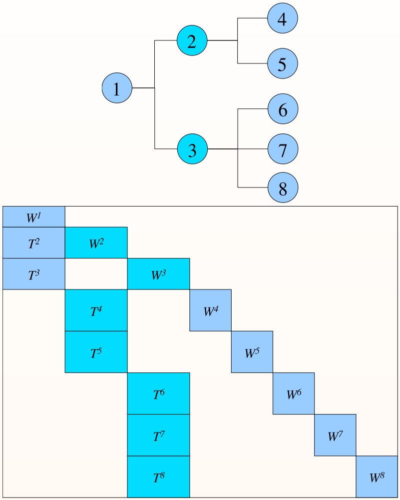
Model of the dynamics of the system (at node l_t):

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

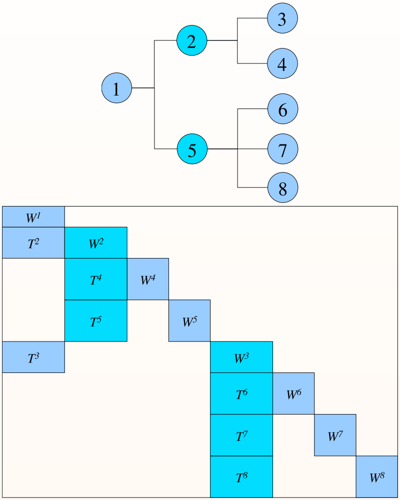
Complete deterministic equivalent formulation

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

Structure of the deterministic equivalent



Breadth-first ordering



Depth-first ordering

Solution methods

We consider solving the deterministic equivalent directly with interior point methods:

- IPM solvers are available in the community (CPLEX Barrier, PCx, IPOPT, 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 Solver

OOPS is a parallel interior point solver for LP/QP 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

Reduced and aggregated event tree

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

We should use the solution to a smaller instance of the problem to generate a warm-start point.

Strategies:

- Scenario reduction: choose a (very) small number of scenarios
- Stage aggregation: merge some of the stages of the problem

Main steps of the algorithm

Exploit the structure of the stochastic program:

1. Generate a small event tree (reduction, aggregation)
2. Solve the small 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

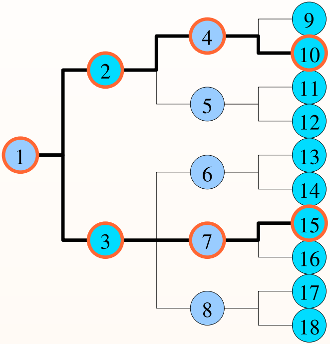
Exploit the structure of the stochastic program:

1. Generate a small event tree (reduction, aggregation)
2. Solve the small deterministic equivalent with loose accuracy
3. Generate a warm-start iterate for the complete problem
4. Solve the complete problem to optimality

Features:

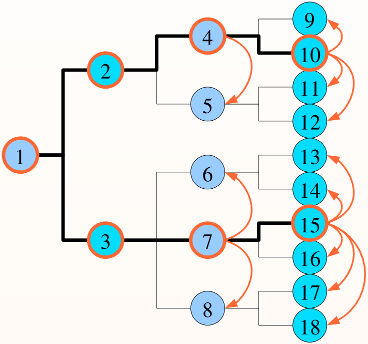
- The small problem is very easy to solve
- We exploit the structure to match the dimensions of the two problems

Warm-start iterate from reduction



Nodes in the reduced tree:
the solution is already available

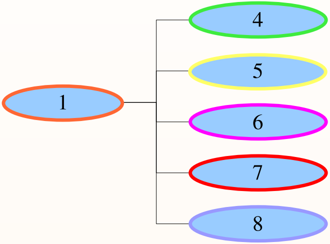
Warm-start iterate from reduction



Nodes in the reduced tree:
the solution is already available

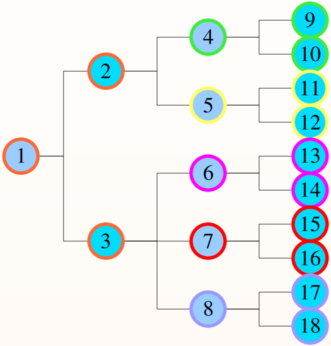
Remaining nodes:
copy the solution from the corresponding reduced-tree node

Warm-start iterate from aggregation



Nodes in the aggregated tree:
correspond to more than one stage

Warm-start iterate from aggregation



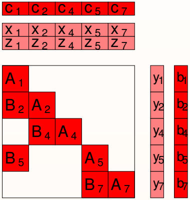
Nodes in the aggregated tree:
correspond to more than one stage

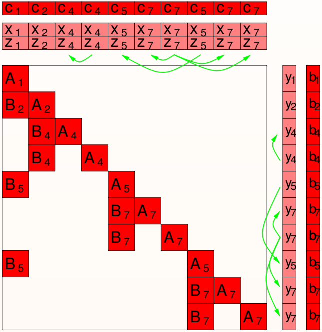
Once disaggregated:
they are initialised from the corresponding aggregated-tree node

Warm-start analysis I

The analysis of the warm-start procedure is based on two steps:

1. Reduced problem ⇒ Expanded problem

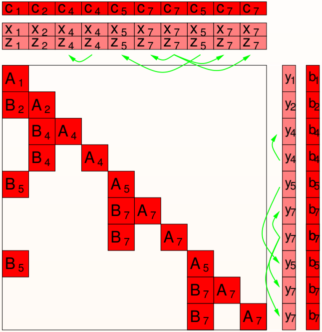


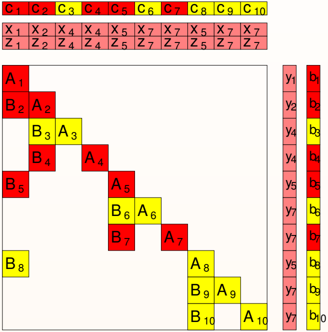


- We can construct primal/dual feasible starting point
- This point is not central: we are duplicating constraints!

Warm-start analysis II

2. Expanded problem ⇒ Complete problem





- We can bound changes in problem data (scenarios changes)
- Determine conditions for a successful warmstart

Pricing of swing options

An electricity swing option is a contract that allows the holder to buy between $\underline{e}$ and $\bar{e}$ units of electricity up to time T for a price of K (per unit).

- S_t: spot price for electricity at time t
- K: strike price of the option
- e_t: total usage of option up to period t
- p_t: electricity used in period t
- ρ: ratchet

Swing pricing model

Haarbrüker, Kuhn, Valuation of electricity swing options by multistage stochastic programming, Working paper, 2004.

The pricing model is a multistage stochastic program:

$$
\begin{array}{lll}
\min_{e,p} & \boldsymbol{E}_\xi \left[\sum_{t=1}^{T} (S_t(\xi_t) - K) p_t(\xi_t)\right] & \\
\text{s.t.} & \underline{e} \le e_T(\xi_T) \le \bar{e} & \\
 & e_t(\xi_t) - e_{t-1}(\xi_{t-1}) = p_t(\xi_{t-1}) & \\
 & \underline{p}_t \le p_t(\xi_t) \le \bar{p}_t, & t = 1, \ldots, T \\
 & |p_t(\xi_t) - p_{t-1}(\xi_{t-1})| \le \rho, & t = 1, \ldots, T
\end{array}
$$

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 problem optimality tolerance: 5.0×10^{-1}
- Complete problem optimality tolerance: 5.0×10^{-4}

Numerical results I

16 scenarios in the reduced tree:

Problem data			Cold start		Warm start	
Name	Stgs	Scens	Iters	Seconds	Iters	Seconds
swingcs5-4	6	1024	10	0.7	4	0.4
swingcs8-4	9	65536	14	114.6	5	46.3
swingcs9-4	10	262144	24	481.1	5	103.7

The number of iterations in the warm-start case is not sensitive to the size of the reduced tree!

Numerical results II

Various levels of aggregation of the tree:

Problem	2 Aggr		4 Aggr		7 Aggr	
Name	Iters	Seconds	Iters	Seconds	Iters	Seconds
swingcs5-4	5	0.5	4	0.4	4	0.3
swingcs8-4	11	110.8	8	65.5	5	46.3
swingcs9-4	5	259.7	5	111.1	5	92.8

Stage aggregation allows for a successful warmstart on these prototype models.

Conclusions and future work

- Aggregated tree solutions contain valuable information to construct a good warm-start iterate for IPMs (50–60% savings in iterations)
- Extend the approach to a multi-start procedure
- Theoretical complexity of such a scheme
- Integrate into a structured modelling language

Workshop on linear and nonlinear stochastic programming

Edinburgh, 3–5 September 2008

Solution methods for linear and nonlinear stochastic programming, modelling issues and modelling systems, scenario generation, stochastic integer programming, numerical and theoretical treatment of risk measures . . .

Keynote lectures: G. Consigli, M. Dempster, A. McNeil, G. Mitra, G. Pflug, M. Steinbach.

Scholarships of up to EUR 300 each available (deadline: 15 June)

http://www.icms.org.uk/workshops/cariplo