Solution Techniques for Large-Scale Financial Planning Problems

Marco Colombo Jacek Gondzio Andreas Grothey

School of Mathematics
University of Edinburgh

ICCOPT II — MOPTA 2007
Hamilton, 16 August 2007

Scope of this talk

Multi-period financial planning problem

Importance of problem formulation

Exploiting structure and parallelism

Warm-start for stochastic programming problems

Financial planning problems

Why:

- Well studied area
- Useful application
- Possible to generate large-scale problems

Stochastic programming framework:

- Multi-period structure
- Uncertain returns

Multi-period financial planning problem

- A set of assets $\mathcal{J} = \{1, ..., J\}$ is given.
- Initial investment b.
- At every stage $t = 0, \ldots, T-1$ we can buy or sell any assets.
- The return of asset j at stage t is uncertain.

Competing objectives:

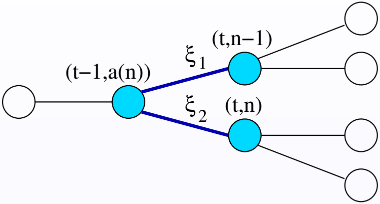
- maximize the final wealth
- minimize the associated risk

Mean-Variance formulation (Markowitz): $\max \boldsymbol{E}(X) - \rho \mathrm{Var}(X)$.

X value of the final portfolio

ρ investor's attitude to risk

Modelling with event tree



With asset $j \in \mathcal{J}$ at node $i = (t, n)$ we associate:

- $x^h_{i,j}$ position in asset j at node i
- $x^b_{i,j}, x^s_{i,j}$ amount of asset j bought/sold at node i
- v_j value of asset j
- $r_{j,t}$ return of asset j when held at time t
- L_i, C_i liabilities/cash contributions at node i

Asset and Liability Management Problem I

Objective:

$$\begin{aligned} \boldsymbol{E}(X) &= (1-c_t)\sum_{i\in L_T} p_i \sum_j v_j x^h_{i,j} = y \\ \mathrm{Var}(X) &= \sum_{i\in L_T} p_i (1-c_t)^2 \Big[\sum_j v_j x^h_{i,j}\Big]^2 - y^2 \end{aligned}$$

Constraints at each node i:

$$\begin{aligned} x^h_{i,j} &= (1+r_{i,j})x^h_{a(i),j} + x^b_{i,j} - x^s_{i,j} \quad (\textit{inventory}) \\ \sum_j (1+c_t) v_j x^b_{i,j} + L_i &= \sum_j (1-c_t) v_j x^s_{i,j} + C_i \qquad (\textit{cash balance}) \end{aligned}$$

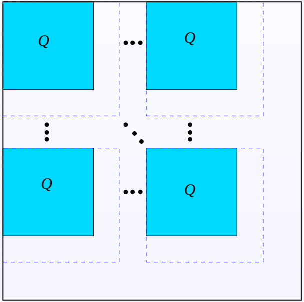
Asset and Liability Management Problem II

$$
\begin{aligned}
\max_{x,y\geq 0} \quad & y - \rho\Big[\sum_{i\in L_T} p_i[(1-c_t)\sum_j v_j x^h_{i,j}]^2 - y^2\Big] \\
\text{s.t.} \quad & (1-c_t)\sum_{i\in L_T} p_i \sum_j v_j x^h_{i,j} = y \\
& (1+r_{i,j})x^h_{a(i),j} = x^h_{i,j} - x^b_{i,j} + x^s_{i,j} \qquad \forall i, \forall j \\
& \sum_j (1+c_t)v_j x^b_{i,j} + L_i = \sum_j (1-c_t)v_j x^s_{i,j} + C_i \quad \forall i \\
& \sum_j (1+c_t)v_j x^b_{0,j} = b
\end{aligned}
$$

Structure of the objective I

Straightforward representation:

$$\begin{aligned}\boldsymbol{E}(X)-\rho\mathrm{Var}(X) &= \boldsymbol{E}(X) - \rho[\boldsymbol{E}(X^2) - \boldsymbol{E}(X)^2] \\ &= \sum_{i\in L_T} p_i \sum_j v_j x_{ij}^h - \rho\left[\sum_{i\in L_T} p_i \sum_j (v_j x_{ij}^h)^2 - [\sum_{i\in L_T} p_i \sum_j v_j x_{ij}^h]^2\right]\end{aligned}$$



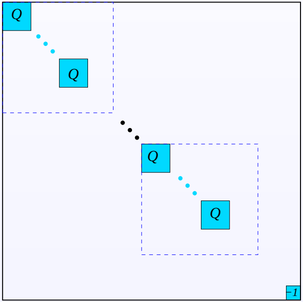
Dense, positive semidefinite Hessian

Structure of the objective II

Alternative representation:

$$\boldsymbol{E}(X) - \rho \mathrm{Var}(X) = y - \rho \Big[\sum_{i \in L_T} p_i \sum_j (v_j x_{ij}^h)^2 - y^2 \Big]$$

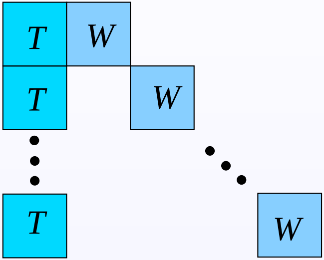
$$\text{where: } y = \sum_{i \in L_T} p_i \sum_j v_j x_{ij}^h$$

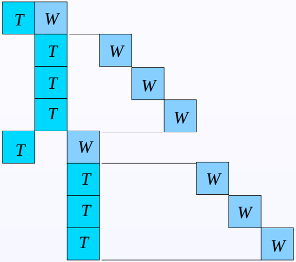


Sparse, indefinite Hessian

Structure in the constraint matrix

Stochastic programming problems give rise to matrices with block-angular structure:





$$T_i x^1 + W_i\, y_i = b_i \qquad\qquad T_{l_t} x_{a(l_t)} + W_{l_t}\, x_{l_t} = b_{l_t}$$

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
Core matrix	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

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
- We assume that structure is known!

OOPS is a general purpose (parallel) Interior Point solver

- Not tuned to any particular hardware or problem
- OOPS currently solves LP/QP problems
- NLP extension solves nonlinear financial planning problems

Performance of OOPS

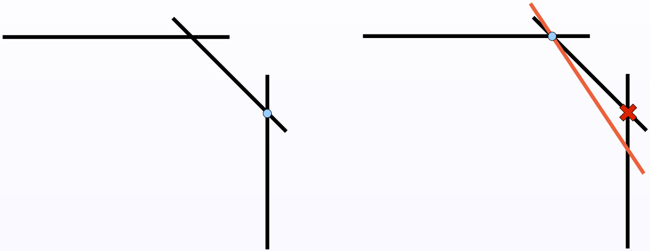
Problem	Stgs	Blks	Assets	Scens	Cons	Vars	iter	time	procs
ALM1	5	10	5	11k	66k	166k	14	86	1
ALM2	6	10	5	111k	666k	1.6M	22	387	5
ALM3	6	10	10	111k	1.2M	3.3M	29	1638	5
ALM4	5	24	5	346k	2.1M	5.2M	33	856	8
ALM5	4	64	12	266k	3.4M	9.6M	18	1195	8
ALM6	4	120	5	1.7M	10.4M	26.1M	18	1470	16
ALM7	4	120	10	1.7M	19.1M	52.2M	19	8465	16

Warm-start strategies

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

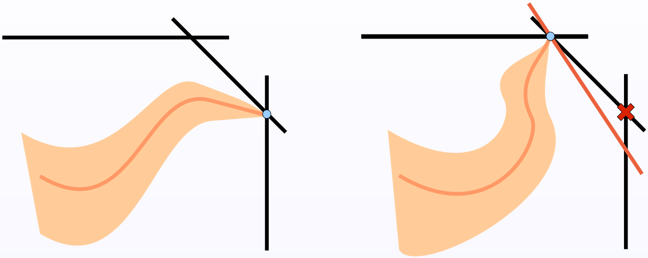
- Important if we are solving a sequence of problems
- Often we may expect that the solution to one problem is close to the solution of the next
- An advanced starting point may lead to reduced computational time than solving the problem from scratch

Warm-start with IPMs



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

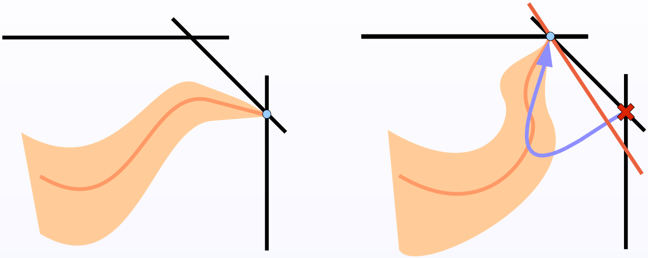
Warm-start with IPMs



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

- worst possible starting point

Warm-start with IPMs



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

- worst possible starting point
- some iterations to approach the new central path
- some iterations for optimality

Warm-start for stochastic problems

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.

Reduced event tree:

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

Main assumptions:

- No knowledge on the underlying stochastic processes
- Required to solve an instance with a specific tree
- We generate and solve the deterministic equivalent

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:

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

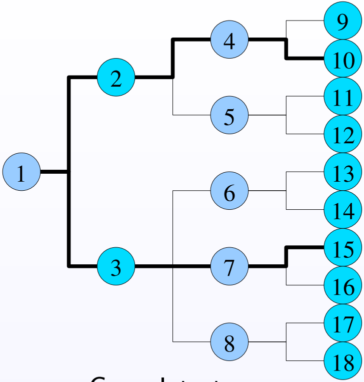
Distance between two scenarios:

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

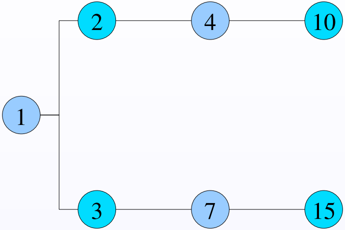
Representative scenario s^* is the one that minimizes the weighted distance from an average scenario $\bar{s}$:

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

Reduced-tree warm-start technique



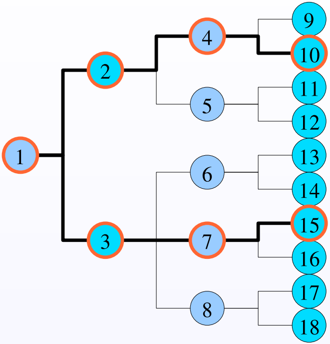
Complete tree



Reduced tree

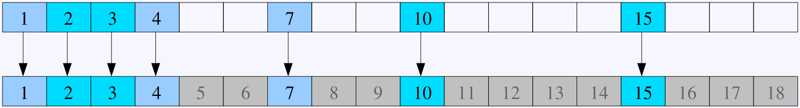
1. Solve the problem with a reduced scenario tree
2. Expand the solution found to construct a starting point for the complete formulation
3. Solve the problem with the complete scenario tree

Construction of the warm-start iterate

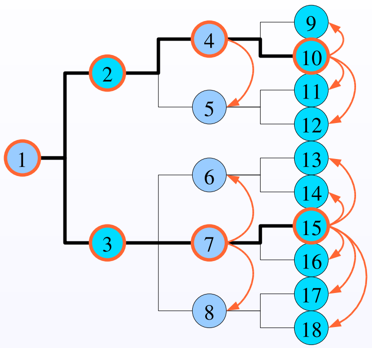


Nodes in the reduced tree:

the solution is already available

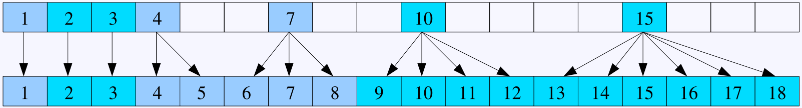


Construction of the warm-start iterate



Nodes in the reduced tree:
the solution is already available

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



Numerical results 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

Numerical results with OOPS (4 processors)

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

- Structure can be exploited both at the linear algebra level and algorithmically
- OOPS provides an efficient implementation of a structure-exploiting parallel software
- Reduced tree solutions contain valuable information to construct a good warm-start iterate
- IPMs can be used successfully warm-started

References

Gondzio, Grothey, Solving non-linear portfolio optimization problems with the primal–dual interior point method. EJOR 2006.

Colombo, Gondzio, Grothey, A warm-start approach for large-scale stochastic linear programs, submitted. Available on optimization-online.org.