Decomposition with IPMs: Applications in Asset Liability Modeling

Jacek Gondzio
Dept. of Maths & Stats, Univ. of Edinburgh
Email: gondzio@maths.ed.ac.uk
URL: http://www.maths.ed.ac.uk/~gondzio

Padova, April, 2002

Two Parts:

- Decomposition with Interior Point Methods.
- Asset Liability Modeling.

- Large strucured problems.
- Decomposition (**Idea**):
  Replace the solution of the large problem by the solution of the sequence of smaller problems.
- Two- and Multi-stage SLPs.
- Benders Decomposition
  (Multistage Stochastic Programs).
- Use IPMs in Decomposition:
  - feasibility reached before optimality;
  - control of the distance to optimality;
  - "central" prices;
  - reoptimization with μ-centers.
- Decomposition and Distributed Computing.

Stochastic LP with recourse

The two-stage stochastic program

$$\begin{array}{llcrcl} \min\limits_{x \in X} & c^T x & + & E_\xi\{q^T y(\xi)\} & & \\ \text{s.t.} & T(\xi)\, x & + & W y(\xi) & = & h(\xi), \\ & x \geq 0, & & y(\xi) \geq 0, & & \forall \xi \in \Xi. \end{array}$$

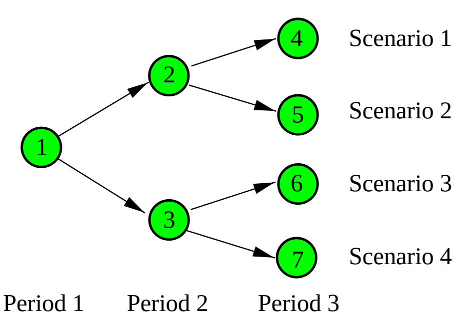
Assume random data has a joint finite discrete distribution $\{(\xi^k, p_k), k=1..N\}$ with $\sum_k p_k = 1$. We have stochastic program with fixed recourse

$$\begin{array}{llcrcll} \min\limits_{x \in X} & c^T x & + & \sum\limits_{k=1}^{N} p_k q_k^T y_k & & & \\ \text{s.t.} & T(\xi^k)\, x & + & W y_k & = & h(\xi^k), & k = 1..N, \\ & x \geq 0, & & y_k \geq 0, & & & k = 1..N. \end{array}$$

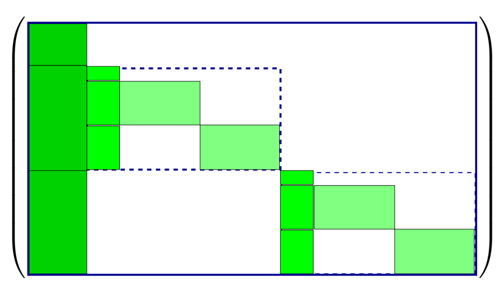
The deterministic equivalent formulation

$$\begin{array}{llllllll} \min\limits_{x \in X} & c^T x & +p_1 q_1^T y_1 & +p_2 q_2^T y_2 & \dots & +p_N q_N^T y_N & & \\ \text{s.t.} & T_1 x & +W y_1 & & & & = & h_1 \\ & T_2 x & & +W y_2 & & & = & h_2 \\ & \vdots & & & \ddots & & & \vdots \\ & T_N x & & & & +W y_N & = & h_N \\ & x \geq 0, & y_1 \geq 0, & y_2 \geq 0, & \dots & y_N \geq 0. & & \end{array}$$

Multi-stage Stochastic Programming



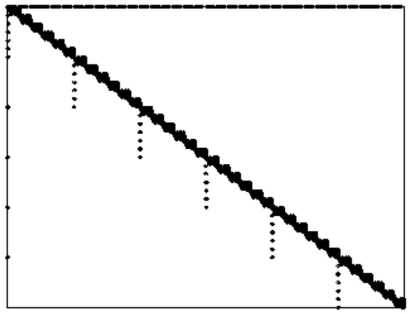
The structured constraint matrix



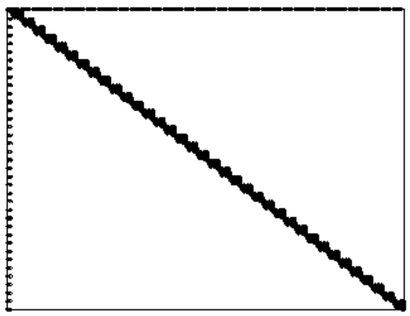
Symmetrical event tree with p realizations at each node and $T+1$ periods corresponds to

$$p^T$$

scenarios.



Multistage stochastic linear program.



Reordered matrix of the multistage SLP.

The deterministic equivalent formulation

$$
\begin{array}{llllll}
\min\limits_{x\in X} & c^T x & +p_1 q_1^T y_1 & +p_2 q_2^T y_2 & \ldots & +p_N q_N^T y_N \\
\text{s.t.} & Ax & & & & = b \\
 & T_1 x & +W y_1 & & & = h_1 \\
 & T_2 x & & +W y_2 & & = h_2 \\
 & \vdots & & & \ddots & \vdots \\
 & T_N x & & & +W y_N & = h_N \\
 & x \ge 0, & y_1 \ge 0, & y_2 \ge 0, & \ldots & y_N \ge 0.
\end{array}
$$

can be rewritten in the equivalent form

$$\min\{c^T x + \sum_{j=1}^{N} Q_j(x) \mid Ax = b,\, x \ge 0\}.$$

$Q_j(x)$, $j = 1, 2, \ldots, N$, is the optimal objective function of the **recourse problem**

$$Q_j(x) = \min\{p_j q_j^T y_j \mid W y_j = h_j - T_j x,\, y_j \ge 0\}.$$

The function $Q_j(x)$ is piecewise linear and convex.

There are two cases for a given x:

Case 1. The subproblem j is *feasible*. It then has an optimal solution $\hat{y}_j$ and $Q_j(x) < +\infty$.

Case 2. The subproblem j is *infeasible*. We then set $Q_j(x) = +\infty$.

Subproblems in SLP answer with cuts

For a given first-stage decision x_0, we get:

Primal		Dual	
min	$q^T y_j$	max	$(h - T_j x_0)^T u_j$
s.t.	$W y_j = h - T_j x_0,$	s.t.	$W^T u_j \le q,$
	$y_j \ge 0;$		u_j free.

Case 1.
The subproblem j is **feasible**. It then has an optimal solution $\hat{y}_j$ and $Q_j(x) < +\infty$.

Let $\hat{y}_j$ be the primal optimal solution and $\hat{u}_j$ be the dual optimal solution. Obviously

$$Q_j(x_0) = q^T \hat{y}_j = (h - T_j x_0)^T \hat{u}_j.$$

For any x we could write (using the dual):

$$Q_j(x) = \sup\{(h - T_j x)^T u_j \,|\, W^T u_j \le q\}.$$

Thus for a given (feasible) $u_j = \hat{u}_j$ we construct the subgradient inequality for $Q_j(x)$:

$$Q_j(x) \ge (h - T_j x)^T \hat{u}_j, \quad \forall x.$$

Using $Q_j(x_0) = (h - T_j x_0)^T \hat{u}_j$ we can rewrite it:

$$Q_j(x) \ge Q_j(x_0) - \hat{u}_j^T T_j (x - x_0), \quad \forall x.$$

This inequality is called the **optimality cut**.

Subproblems in SLP answer with cuts

For a given first-stage decision x_0, we get:

Primal		Dual	
min	$q^T y_j$	max	$(h - T_j x_0)^T u_j$
s.t.	$W y_j = h - T_j x_0,$	s.t.	$W^T u_j \le q,$
	$y_j \ge 0;$		u_j free.

Case 2.
The subproblem j is **infeasible**. We then set $Q_j(x) = +\infty$.

Since the primal is infeasible the dual must be unbounded. Let $\tilde{u}_j$ be its ray of unboundedness.

For any *feasible* first-stage variable x there exists a feasible recourse action y such that

$$W y = h - T_j x, \quad y \ge 0.$$

Scalar multiplication of this inequality with $\tilde{u}_j$ (note that $W^T \tilde{u}_j \le 0$) yields

$$\tilde{u}_j^T (h - T_j x) = \tilde{u}_j^T W y \le 0, \quad \forall x,$$

or in an equivalent form:

$$\tilde{u}_j^T T_j x \ge \tilde{u}_j^T h, \quad \forall x.$$

This inequality is called the **feasibility cut**.

Let us observe, that

$$\bar{\theta}(x) = c^T x + \sum_{j=1}^{N} Q_j(x)$$

is an upper bound for the optimal solution of the original problem. Note that it may take the $+\infty$ value if at least one subproblem is infeasible.

The original problem can be replaced with the so-called **restricted master program**:

$$\begin{aligned}
\min \;\; & c^T x \; + \; \theta, \\
\text{s.t.} \;\; & Ax \; = \; b, \quad x \; \geq \; 0, \\
& \theta \geq \sum_{j=1}^{N} z_j, \\
& z_j \geq Q_j(x^k) - (\hat{u}_j^k)^T T_j (x - x^k), \; \forall j \leq N, \; \forall k : Q_j(x^k) < \infty, \\
& 0 \geq (\tilde{u}_j^k)^T h_j - (\tilde{u}_j^k)^T T_j x, \qquad \forall j \leq N, \; \forall k : Q_j(x^k) = \infty.
\end{aligned}$$

The optimal solution $\hat{x}$ of this problem is a candidate for the next query point. The optimal objective of this problem is a lower bound $\underline{\theta}$ for the optimal solution of the original problem.

Subproblems are solved for several *query points* $\{x^k\}_{k=1,2,\ldots,\kappa}$. They produce cuts (either the optimality or the feasibility ones).

These cuts are then appended to the master. The master is solved producing $\hat{x}$ and the optimal objective $\underline{\theta}$. Its optimal solution $\hat{x}$ becomes a new query point x^{k+1} sent to subproblems.

The algorithm continues until

$$\bar{\theta}(x) - \underline{\theta} \leq \epsilon.$$

Note that in every outer iteration of the decomposition method, N subproblems have to be solved. These subproblems are completely independent (**straightforward parallelization**).

Subproblem $j = 1, 2, \ldots, N$ depends on the first stage variables x. The modification of x changes only its right-hand side

$$RHS_j = h_j - T_j x^k,$$

so the re-optimization technique is useful.

IPMs in Decomposition

Important features of IPMs:

- feasibility reached before optimality;
- easy control of the distance to optimality.

Where can we use them?

- Early termination in the master problem:
  - "central" prices;
  - degeneracy avoided.
- Early termination in the subproblems:
  - ε-subgradients.

What do we need?

- reoptimization (warm starting).

J. Gondzio,
Warm start of the primal-dual method applied in the cutting plane scheme, *Math Prog*, 83 (1998), pp. 125–143.

J. Gondzio and J.-P. Vial,
Warm start and ε-subgradients in cutting plane scheme for block-angular linear programs, *Comp Opt and Appl*, 14 (1999), pp. 17–36.

First Order Optimality Conditions

The first order optimality conditions (FOC)

$$\begin{aligned}
Ax &= b, \\
A^T y + z &= c, \\
XZe_n &= \mu e_n,
\end{aligned}$$

where $X = diag\{x_j\}$, $Z = diag\{z_j\}$
and $e_n = (1, \cdots, 1) \in \mathcal{R}^n$.

Analytic center (μ-center): a (unique) point $(x(\mu), y(\mu), z(\mu))$, $x(\mu) > 0$, $z(\mu) > 0$ that satisfies FOC.

Parameter μ in the analytic center controls the distance to optimality.

$$c^T x - b^T y = c^T x - x^T A^T y = x^T (c - A^T y) = x^T z = n\mu.$$

Primal-dual algorithm terminates when the duality gap drops below a predetermined relative optimality tolerance ϵ, i.e. when

$$|c^T x - b^T y| \leq \epsilon(|c^T x| + 1).$$

For a given first-stage decision x_0, we get:

Primal

$$\begin{array}{ll} \min & q^T y_j \\ \text{s.t.} & W y_j = h - T_j x_0, \\ & y_j \geq 0; \end{array}$$

Dual

$$\begin{array}{ll} \max & (h - T_j x_0)^T u_j \\ \text{s.t.} & W^T u_j \leq q, \\ & u_j \text{ free.} \end{array}$$

Suppose the subproblem j is **feasible**. It then has an optimal solution $\hat{y}_j$ and $Q_j(x) < +\infty$.

Let $\hat{y}_j$ be the primal feasible and $\hat{u}_j$ be the dual feasible solution such that

$$|q^T \hat{y}_j - (h - T_j x_0)^T \hat{u}_j| \leq \varepsilon,$$

hence

$$(h - T_j x_0)^T \hat{u}_j \leq Q_j(x_0) \leq (h - T_j x_0)^T \hat{u}_j + \varepsilon.$$

For any x we could write (using the dual):

$$Q_j(x) = \sup\{(h - T_j x)^T u_j \,|\, W^T u_j \leq q\}.$$

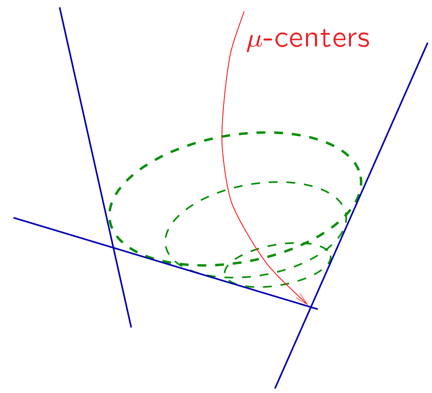
Thus for a given (feasible) $u_j = \hat{u}_j$ we construct the subgradient inequality for $Q_j(x)$:

$$Q_j(x) \geq (h - T_j x)^T \hat{u}_j, \quad \forall x.$$

Using $Q_j(x_0) \leq (h - T_j x_0)^T \hat{u}_j + \varepsilon$ we get:

$$Q_j(x) \geq Q_j(x_0) - \hat{u}_j^T T_j (x - x_0) - \varepsilon, \quad \forall x.$$

This inequality is called the **ε-optimality cut**.



Let $(\hat{x}; \hat{y}, \hat{z})$ be the exact μ-center

Ellipsoid in the dual space,
with the primal-dual scaling $\hat{D}^2 = \hat{X}\hat{Z}^{-1}$

$$\hat{E} = \{y \in \mathcal{R}^m : \|\hat{D}A^T(y - \hat{y})\| \leq \mu^{1/2}\}.$$

Ellipsoid from approximate μ-center

Let $(\bar{x}; \bar{y}, \bar{z})$ be an approximation of μ-center

$$\|(\bar{X}\bar{z}/\mu) - e\| \leq p \leq 1.$$

The corresponding ellipsoid

$$\bar{E} = \{y \in \mathcal{R}^m : \|\bar{D}A^T(y - \bar{y})\| \leq \mu^{1/2}(1-p)^{1/2}\}.$$

$\bar{E}$ approaches $\hat{E}$ as p goes to zero.

Let $(\tilde{x}; \tilde{y}, \tilde{z})$ be another approx. of μ-center

$$\beta\mu \leq \tilde{x}_j \tilde{s}_j \leq (1/\beta)\mu, \quad \beta \in (0, 1].$$

The corresponding ellipsoid

$$\tilde{E} = \{y \in \mathcal{R}^m : \|\tilde{D}A^T(y - \tilde{y})\| \leq \mu^{1/2}\beta^{1/2}\}.$$

$\tilde{E}$ approaches $\hat{E}$ as β goes to 1.

Observation:

$$\hat{E}, \bar{E}, \tilde{E} \subset \mathcal{D} = \{y \in \mathcal{R}^m : A^T y \leq c\}.$$

Close to Optimality Analytic Center

- nearly-optimal
  – to optimize fast
- far from the boundary
  – to absorb larger perturbations

A point in the neighborhood of the central path:

$$\begin{array}{rcl} \|Ax - b\| & \leq & \epsilon_p(\mu)(\|b\| + 1), \\ \|A^T y + s - c\| & \leq & \epsilon_d(\mu)(\|c\| + 1), \\ \beta\mu \leq \; x_j s_j & \leq & \frac{1}{\beta}\mu, \qquad \forall i. \end{array}$$

Split solution method:

- find 1- (2- or 3-) digit optimal μ-center and save it for future warm start;
- continue to get required 6- (8-) digit optimal solution.

For a new (perturbed) problem:

- update μ-center for the new problem (start from the earlier saved μ-center);
- get required 6- (8-) digit optimal solution.

Step 1: restore μ-center

- use multiple centrality correctors

Step 2: get optimal solution

- use standard primal-dual method

- Asset liability modeling
  - risk management
  - multiple decision stages
  - curse of dimensionality
  - very large-scale optimization
- Numerical results
  - problem generation
  - storage management
  - distributed computing
- Conclusions

Asset Liability Management

Assume we are given an initial capital W_0, the planning horizon T, and the goal wealth at the end of the planning horizon W_T.

At every period $t = 0, 1, \ldots, T-1$, we:

- contribute C_t to the portfolio;
- pay liabilities L_t;
- rebalance the portfolio (buy/sell assets).

Financial Planning Problem:
Decide which assets to buy/sell at time t.

Cash balance at time t:

$$\text{cash inflow } (t) = \text{ cash outflow } (t),$$

where:
inflow: borrowing, assets sold and contributions;
outflow: lending, assets bought, liabilities.

Financial Planning Problem

Consider a multi-period financial planning problem. At every stage $t = 0, ..., T-1$ we can buy or sell different assets from the set $\mathcal{J} = \{1, ..., J\}$ (e.g. bonds, stock, real estate), we can lend the money to other parties or borrow it from the bank.
The return of asset j at stage t is **uncertain**.

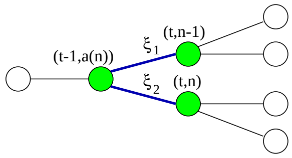
We have an initial sum S_0 to invest and we want to maximize the expected final wealth S_T (or to maximize its expected utility $U(S_T)$).

Asset Liability Modeling
Suppose that at every stage t, we contribute a certain amount of cash C_t to the portfolio and we pay a certain liability L_t.
Such a financial planning problem is called the *asset liability management problem*.

This problem is of crucial importance to **pension funds** and **insurance companies**.

Note a **dynamic** aspect of decisions to be taken: the portfolio is to be re-balanced at every stage.
Note a **stochastic** aspect of the problem: the returns of assets are uncertain.

We model this problem using an event tree



and decision variables associated with its nodes (t, n). We assume that applications $a(t, n)$ and $s(t, n)$ which define *ancestor* and *son* of node (t, n), respectively are known.

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

- $x_{j,t,n}$ the position in asset j in node (t, n);
- $x^b_{j,t,n}$ the amount of asset j bought in (t, n);
- $x^s_{j,t,n}$ the amount of asset j sold in (t, n).

For any $t : 1 \leq t \leq T$, we write the **inventory equation** for asset j at node (t, n)

$$x_{j,t,n} = (1 + r_{j,t,n}) \cdot x_{j,t-1,a(t,n)} + x^b_{j,t,n} - x^s_{j,t,n},$$

where $r_{j,t,n}$ is a return of asset j corresponding to moving from node $(t-1, a(t, n))$ to node (t, n) in the event tree.

Initial inventory equation for asset j:

$$x_{j,0,1} = x^{initial}_j + x^b_{j,0,1} - x^s_{j,0,1}.$$

Let P_j be the initial price of asset j. We assume that the **transaction costs** are proportional to the value of asset bought or sold. To buy $x^b_{j,t,n}$ of asset j at stage t we have to pay

$$(1 + \gamma_b) \cdot P_j \cdot x^b_{j,t,n}.$$

Analogously, for selling $x^s_{j,t,n}$ of asset j at stage t we get

$$(1 - \gamma_s) \cdot P_j \cdot x^s_{j,t,n}.$$

Let $m_{t,n}$, $m^b_{t,n}$ and $m^l_{t,n}$ be the *cash* hold, borrowed and lent at time t at node n, respectively. The borrowing and lending have return rates $r^b_{t,n}$ and $r^l_{t,n}$, respectively. For example, for the money $m^l_{t-1,a(t,n)}$ lent at stage $t-1$, we receive back $(1+r^l_{t,n}) \cdot m^l_{t-1,a(t,n)}$ at stage t.

The **cash balance equation** states that the cash *inflow* is equal to the cash *outflow*

$$C_{t,n} + m^b_{t,n} + (1 + r^l_{t,n}) m^l_{t-1,a(t,n)} + \sum_{j=1}^{J} (1 - \gamma_s) P_j x^s_{j,t,n}$$
$$= L_{t,n} + m^l_{t,n} + (1 + r^b_{t,n}) m^b_{t-1,a(t,n)} + \sum_{j=1}^{J} (1 + \gamma_b) P_j x^b_{j,t,n}$$

at any node (t, n), $t = 1, ..., T$, $n = 1, ..., N(t)$.

Special Cash Balance Equations

Initial cash balance

$$S_0 + C_{0,1} + m^b_{0,1} + \sum_{j=1}^{J} (1 - \gamma_s) P_j x^s_{j,0,1}$$
$$= L_{0,1} + m^l_{0,1} + \sum_{j=1}^{J} (1 + \gamma_b) P_j x^b_{j,0,1}$$

i.e., we assume that there was an initial portfolio of assets, so we can also *sell* at stage $t = 0$.

Final cash balance

$$C_{T,n} + m^b_{T,n} + (1 + r^l_{T,n}) m^l_{T-1,a(T,n)} + \sum_{j=1}^{J} (1 - \gamma_s) P_j x^s_{j,T,n}$$
$$= S_T + L_{T,n} + m^l_{T,n} + (1 + r^b_{T,n}) m^b_{T-1,a(T,n)} + \sum_{j=1}^{J} (1 + \gamma_b) P_j x^b_{j,T,n}$$

i.e., we assume that the final portfolio can include assets, so we can also *buy* at stage T.

These two constraints may be used for example in asset liability management problem for a pension fund or an insurance company. (Indeed, life does not end at time T.)

Additional Constraints

Different companies may define particular **policy restrictions** for asset mix. For example, the weights of asset mix can be bounded:

$$w^{lo}_j \cdot \sum_{j=1}^{J} x_{j,t,n} \leq x_{j,t,n} \leq w^{up}_j \cdot \sum_{j=1}^{J} x_{j,t,n}, \quad \forall j, t, n.$$

Also the contributions are bounded:

$$C^{lo}_{t,n} \leq C_{t,n} \leq C^{up}_{t,n}, \quad \forall t, n.$$

Total asset value at the end of period t,

$$A_{t,n} = \sum_{j=1}^{J} (1 + r_{j,t,n}) P_j x_{j,t-1,a(t,n)}$$
$$+ (1 + r^l_{t,n}) m^l_{t-1,a(t,n)} - (1 + r^b_{t,n}) m^b_{t-1,a(t,n)}$$

should not decrease below the minimum level of funding ratio F^{min} for liabilities at this period. To get more flexibility in modeling (to ensure complete recourse), we allow a deficit $Z_{t,n}$:

$$A_{t,n} \geq F^{min} \cdot L_{t,n} - Z_{t,n}, \quad \forall t, n,$$

for which we shall penalize in the objective. The final value of assets at the end of the planning horizon should cover the final liabilities and ensure the final wealth of at least S_T, hence

$$A_{T,n} \geq F^{end} \cdot L_{T,n} + S_{T,n}, \quad \forall n.$$

In a simple financial planning problem we usually maximize the expected value of the final portfolio converted into cash.

In asset liability management problem we may be more flexible:

- we accept (small) deficits, $Z_{t,n}$;
- we can increase the contributions, $C_{t,n}$;
- we can borrow cash, $m^b_{t,n}$, etc.

Suppose we:

- penalize for deficits; and
- minimize contributions.

Hence we get the following **objective**

$$\min \sum_{t=0}^{T-1} \sum_{n=1}^{N(t)} \pi_{t,n} C_{t,n} + \lambda \sum_{t=1}^{T} \sum_{n=1}^{N(t)} \pi_{t,n} \frac{Z_{t,n}}{L_{t,n}}.$$

Future is **uncertain**.
Multiple stage decision problem.

One way of modeling the dynamic asset liability management problems is to apply the stochastic programming approach.

Bradley and Crane (1972)
Kusy and Ziemba (1986)
Carino et al. (1994)
Mulvey and Vladimirou (1992)
Zenios (1995)
Consigli and Dempster (1998).

Uncertain asset returns
discrete approx. of conditional distributions
p realizations for the one-period asset returns.

Multiple stage decisions
T is the number of portfolio rebalancing dates.

Number of scenarios:

$$p^T$$

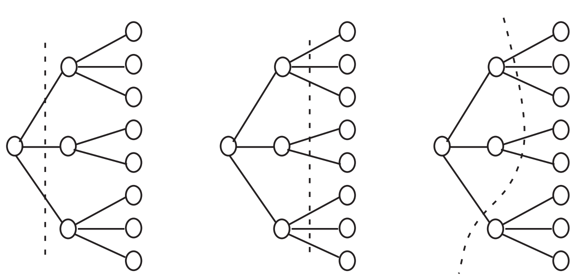
How to Decompose

Multi-stage stochastic programs are difficult.
Nested structure requires nested decomposition.
But there are no convincing results that *nested* Benders decomposition can solve large problems.

Our approach:

- assemble stages;
- use Benders decomposition (for two-stage problems);
- use IPMs to solve (large) LPs.

From multi-stage to two-stage LP:



(a) (b) (c)

ALM: Academic Example

Joint work with:
E. Fragnière, R. Sarkissian and J.-P. Vial
Logilab, University of Geneva, Switzerland.

Financial planning model inspired by
J. Birge and G. Infanger.

Toy model: The portfolio management problem with five decision variables per period (we can invest in 4 securities and cash) and a single budget constraint.

Prices of securities computed by a multivariate log-normal random generator implemented in MATLAB.

The model does not include transaction costs so myopic policies would be sufficient to achieve optimality, Hakansson (1971,1974).

There are up to **7** stages and up to **10** random outcomes per period: **one million scenarios**.

Public domain model (added to GAMS library):
`http://ecolu-info.unige.ch/~logilab/SetWeb/fragnier.gms`

Joint work with:
R. Kouwenberg
Erasmus Univ., Rotterdam, The Netherlands.
JG & RK:
High Performance Computing in Asset Liability Management, *Operations Research* 49 (2001), pp. 879–891.

Dutch pension fund
Boender (1997), Kouwenberg (1998).

Goal: provide the participants with a benefit payment equal to 70% of their final salary.
Participants and employers pay contributions to the pension fund each year prior to retirement.
The pension fund decides how to invest these payments; it has to fulfill long term obligations and to meet short term solvency requirements.

The model recommends an investment policy and a contribution policy, given the economic expectations and the preferences of pension fund.

7-stage stochastic program generated with GAMS modeling language. Up to 10^6 scenarios.

Problem	Events	Scen.	Rows	Columns
P6R9	3	3^6	1094	3279
P6R16	4	4^6	5462	15018
P6R25	5	5^6	19532	50781
P6R36	6	6^6	55988	139968
P6R49	7	7^6	137258	338339
P6R64	8	8^6	299594	711534
P6R81	9	9^6	597872	1395033
P6R100	10	10^6	1111112	2555556

Parallel decomposition. 10 Pentium Pro PCs, 200 MHz each, 64 MB RAM + 384 MB swap.

Problem	Events	SubProbs	Procs	Time [s]
P6R9	3	9	3	8
P6R16	4	16	4	20
P6R25	5	25	5	49
P6R36	6	36	6	100
P6R49	7	49	7	512
P6R64	8	64	8	1851
P6R81	9	81	9	6656
P6R100	10	100	10	10325

Warm Start with HOPDM

Between subsequent outer iterations subproblems differ uniquely in the right-hand sides

$$RHS_j = h_j - T_j x^k.$$

Use re-optimization technique.

	Subproblem Sizes			IPM Iters	
Problem	Rows	Cols	Nonz.	Avr.	Last
P6R16	346	2000	4884	7	5
P6R25	786	4418	11115	6	4
P6R36	1560	8602	22054	9	5
P6R49	2806	15260	39639	13	7
P6R64	4686	25220	66144	12	8
P6R81	7386	39430	104179	15	8
P6R100	11116	58958	156690	17	10

Violent changes of RHS occur if x^k and x^{k+1} differ significantly (in early iterations of decomposition).

When x^k converges to optimum, changes of RHS are smaller and re-optimization is easier.

Efficiency of LEQGEN

p	p^T	Rows	Columns	Non-zeros
4	4^6	17,741	35,484	92,800
5	5^6	58,586	117,174	304,648
6	6^6	158,623	317,248	821,108
7	7^6	372,548	745,098	1,921,564
8	8^6	786,425	1,572,852	4,044,472
9	9^6	1,527,886	3,055,774	7,838,720
10	10^6	2,777,771	5,555,544	14,222,188
11	11^6	4,783,208	9,566,418	24,447,508
12	12^6	7,872,133	15,744,268	40,175,024
13	13^6	12,469,250	24,938,502	63,552,952

p	p^T	Det. Equiv.	SubProb. at time 2	SubProb. at time 3
4	4^6	20.30 Mb	1.28 Mb	0.33 Mb
5	5^6	66.93 Mb	2.69 Mb	0.55 Mb
6	6^6	180.96 Mb	5.04 Mb	0.85 Mb
7	7^6	424.55 Mb	8.67 Mb	1.25 Mb
8	8^6	895.42 Mb	14.00 Mb	1.77 Mb
9	9^6	1738.39 Mb	21.47 Mb	2.40 Mb
10	10^6	3158.56 Mb	31.59 Mb	3.18 Mb
11	11^6	5436.10 Mb	44.94 Mb	4.10 Mb
12	12^6	8942.67 Mb	62.11 Mb	5.19 Mb
13	13^6	14159.45 Mb	83.79 Mb	6.46 Mb

p	SubPr	Par. jobs	Outer iters	Decomp. CPU time	LEQGEN CPU time
SubPbs **stored**, split between 2 and 3.					
4	16	4	11	62	3
5	25	5	9	203	4
6	36	6	9	393	7
7	49	7	10	1194	13
≥8	64	8	NA	NA	NA
SubPbs **regenerated**, split between 2 and 3.					
4	16	4	11	68	32
5	25	5	9	219	37
6	36	6	9	422	64
7	49	7	10	1166	128
8	64	8	10	2329	221
9	81	9	9	4261	339
10	100	10	10	9644	572
11	121	11	10	16102	1213
≥12	144	12	NA	NA	NA
SubPbs **regenerated**, split between 3 and 4.					
4	64	4	11	78	56
5	125	5	11	235	114
6	216	6	10	380	182
7	343	7	10	767	277
8	512	8	10	1438	408
9	729	9	10	2612	614
10	1000	10	11	4876	988
11	1331	11	9	6296	1311
12	1728	12	9	10256	2071
13	2197	13	8	14138	2294

Parsytec CC16 Parallel Machine.

Problem with 6 trading dates and 6 realizations at each node of the event tree.

$6^6 = 46,656$ scenarios:

m = 158,623, n = 317,248, nonz = 821,108.

Procs	CPU time	Speed-ups w.r.t.	
		1 Proc	4 Procs
1	2810	1	-
2	1384	2.03	-
3	880	3.19	-
4	570	4.93	4
5	497	5.65	4.59
6	396	7.10	5.76
7	370	7.59	6.16
8	320	8.78	7.13
9	264	10.64	8.64
10	260	10.81	8.77
11	257	10.93	8.87
12	205	13.71	11.12

Imperfect speed-ups on 5,7,8,10 or 11 procs.

Conclusions

- Dynamic asset liability management:
  uncertainty of the asset returns;
  dynamic structure of portfolio rebalancing dates.

- Very large stochastic LPs need:
  reliable optimization;
  fast model generation;
  high-performance computing.

- Very large LP:
  $12,5 \times 10^6$ rows and 25×10^6 columns
  solved in about 5 hours.