School of Mathematics

Large Scale Optimization with Interior-Point Methods

Jacek Gondzio
Email: `J.Gondzio@ed.ac.uk`
URL: `http://www.maths.ed.ac.uk/~gondzio`

joint work with **Andreas Grothey**

Ongoing Project

Exploiting structure in very large scale optimization.

Many of you will have certainly seen earlier results.

Outline

- **Interior Point Methods**
  - complementarity conditions
  - linear algebra: LP, QP and NLP
- **Very Large Scale Optimization**
  - implicit inverse representation
  - from sparsity to block-sparsity
  - structured optimization problems
  - **OOPS**: Object-Oriented Parallel Solver
- **Applications**
  - financial planning problems (nonlinear risk measures)
  - utility distribution planning
  - data mining (nonlinear kernels in SVMs)
  - PDE-constrained optimization
- **Conclusions**

Complementarity $x_j \cdot s_j = 0 \quad \forall j = 1, 2, ..., n.$

Simplex Method makes a guess of optimal partition:

For *basic* variables, $s_B = 0$ and

$$(x_B)_j \cdot (s_B)_j = 0 \quad \forall j \in \mathcal{B}.$$

For *non-basic* variables, $x_N = 0$ hence

$$(x_N)_j \cdot (s_N)_j = 0 \quad \forall j \in \mathcal{N}.$$

Interior Point Method uses ε-mathematics:

Replace $x_j \cdot s_j = 0 \quad \forall j = 1, 2, ..., n$
by $x_j \cdot s_j = \mu \quad \forall j = 1, 2, ..., n.$

Force convergence $\mu \rightarrow 0$.

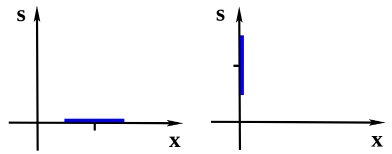
First Order Optimality Conditions

Simplex Method:

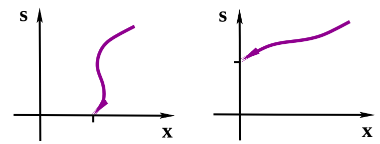
$$\begin{aligned} Ax &= b \\ A^T y + s &= c \\ XSe &= 0 \\ x,\ s &\geq 0. \end{aligned}$$

Interior Point Method:

$$\begin{aligned} Ax &= b \\ A^T y + s &= c \\ XSe &= \mu e \\ x,\ s &\geq 0. \end{aligned}$$



Basic: x > 0, s = 0 Nonbasic: x = 0, s > 0



"Basic": x > 0, s = 0 "Nonbasic": x = 0, s > 0

Wright, Primal-Dual Interior-Point Methods, SIAM, 1997.

Stochastic Programming Problems

$\longrightarrow$(**PhD Thesis** of **Marco Colombo**, talk tomorrow)

		Number of Iterations		
Scenarios	Variables	standard	correctors	warm-started
100	105K	23	20	7
200	209K	64	25	9
800	836K	28	22	11
1200	1.6M	33	26	12

Theory: IPMs converge in $\mathcal{O}(\sqrt{n})$ or $\mathcal{O}(n)$ iterations
Practice: IPMs converge in $\mathcal{O}(\log n)$ iterations

... but one iteration may be expensive!

Interior Point Methods

Marsten, Subramanian, Saltzman, Lustig and Shanno:
"Interior point methods for linear programming:
Just call Newton, Lagrange, and Fiacco and McCormick!",
Interfaces 20 (1990) No 4, pp. 105–116.

- **Fiacco & McCormick (1968)**
  inequality constraints $\longrightarrow$ logarithmic barrier;
  a sequence of unconstrained minimizations
- **Lagrange (1788)**
  equality constraints $\longrightarrow$ multipliers;
- **Newton (1687)**
  solve unconstrained minimization problems;

KKT systems in IPMs for LP, QP and NLP

LP $$\begin{bmatrix} \Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ d \end{bmatrix}$$

QP $$\begin{bmatrix} Q + \Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ d \end{bmatrix}$$

NLP $$\begin{bmatrix} Q(x,y) + \Theta_P^{-1} & A(x)^T \\ A(x) & -\Theta_D \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ d \end{bmatrix}$$

The rest of the talk
$\longrightarrow$ focuses on linear algebra issues.

KKT Systems Arising in IPMs

Quasidefinite matrix: $H = \begin{bmatrix} Q & A^T \\ A & -F \end{bmatrix}$

where Q and F are positive definite.

Vanderbei, *SIOPT* 5 (1995) pp 100-113:
"Symmetric QDFM's are strongly factorizable."

For any **QDFM** there exists a **Cholesky-like** factorization

$$H = LDL^T,$$

where D is **diagonal** but **not positive definite**:
D has n positive pivots and m negative pivots.

Primal-Dual Regularization

Altman & G., *OMS* 11-12 (1999) 275-302.

Replace $H = \begin{bmatrix} Q & A^T \\ A & -F \end{bmatrix}$ by $H_R = \begin{bmatrix} Q & A^T \\ A & -F \end{bmatrix} + \begin{bmatrix} R_p & 0 \\ 0 & -R_d \end{bmatrix}.$

Interpretation: proximal terms added to primal/dual objectives;
Dynamic regularization: correct only suspicious pivots.

Inspired by:

Saunders, in Adams and Nazareth, eds, pp 92-100, SIAM 1996.
Saunders and Tomlin, Tech Rep SOL 96-4, Stanford, Dec 1996.

Primal Regularization

Primal Problem

$$\begin{array}{ll} \min & z_P = c^T x + \frac{1}{2} x^T Q x - \mu \sum_{j=1}^n \ln x_j \\ \text{s.t.} & Ax = b, \;\; x \ge 0 \end{array}$$

$$\rightarrow \quad \begin{bmatrix} Q+\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ d \end{bmatrix}.$$

Primal Regularized Problem

$$\begin{array}{ll} \min & z_P + \frac{1}{2}(x - x_0)^T R_p (x - x_0) \\ \text{s.t.} & Ax = b, \;\; x \ge 0 \end{array}$$

$$\rightarrow \quad \begin{bmatrix} Q+\Theta^{-1}+R_p & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f' \\ d \end{bmatrix}.$$

Dual Regularization

Dual Problem

$$\begin{array}{ll} \max & z_D = b^T y - \frac{1}{2} x^T Q x + \mu \sum_{j=1}^n \ln s_j \\ \text{s.t.} & A^T y + s - Qx = c, \;\; s \ge 0 \end{array}$$

$$\rightarrow \quad \begin{bmatrix} Q+\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ d \end{bmatrix}.$$

Dual Regularized Problem

$$\begin{array}{ll} \max & z_D - \frac{1}{2}(y - y_0)^T R_d (y - y_0) \\ \text{s.t.} & A^T y + s - Qx = c, \;\; s \ge 0 \end{array}$$

$$\rightarrow \quad \begin{bmatrix} Q+\Theta^{-1} & A^T \\ A & -R_d \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f' \\ d \end{bmatrix}.$$

Structured Problems

Observation:

Truly large scale problems are not only sparse...
$\rightarrow$ such problems are structured

Structure is displayed in:

- **Jacobian matrix** A
- **Hessian matrix** Q

Structure can be exploited in:

- **IPM Algorithm**$\longrightarrow$(talk by **Marco Colombo** tomorrow)
- **Linear Algebra of IPM**$\longrightarrow$(focus of the rest of this talk)

Minimum Degree Ordering

Sparse Matrix **Pivot** h_{11} **Pivot** h_{22}

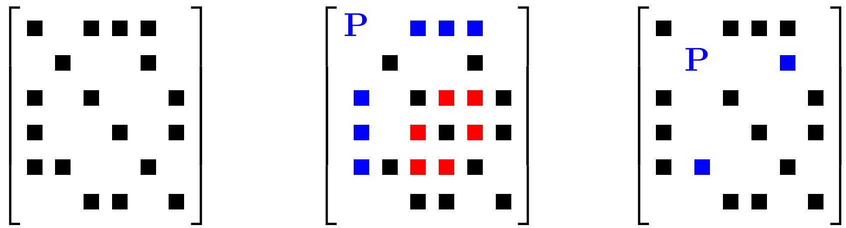
$$H = \begin{bmatrix} x & & x & x & x & \\ & x & & & x & \\ x & & x & & & x \\ x & & & x & & x \\ x & x & & & x & \\ & & x & x & & x \end{bmatrix} \quad \begin{bmatrix} \mathbf{p} & & \mathbf{x} & \mathbf{x} & \mathbf{x} & \\ & x & & & x & \\ \mathbf{x} & & x & \mathbf{f} & \mathbf{f} & x \\ \mathbf{x} & & \mathbf{f} & x & \mathbf{f} & x \\ \mathbf{x} & x & \mathbf{f} & \mathbf{f} & x & \\ & & x & x & & x \end{bmatrix} \quad \begin{bmatrix} x & & x & x & x & \\ & \mathbf{p} & & & \mathbf{x} & \\ x & & x & & & x \\ x & & & x & & x \\ x & \mathbf{x} & & & x & \\ & & x & x & & x \end{bmatrix}$$

Minimum degree ordering:
choose a diagonal element corresponding to a row with the *minimum* number of nonzeros.
Permute rows and columns of H accordingly.

From Sparsity to Block-Sparsity:

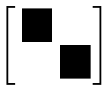
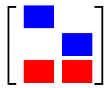
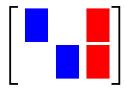
Apply minimum degree ordering to **(sparse) blocks**:

Block-Sparse Matrix **Pivot Block** H_{11} **Pivot Block** H_{22}

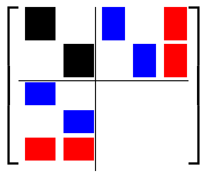
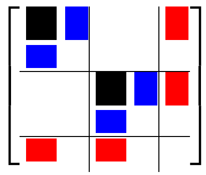
$H =$ 

Choose a diagonal block-pivot corresponding to a block-row with the *minimum* number of blocks.
Permute block-rows and block-columns of H accordingly.

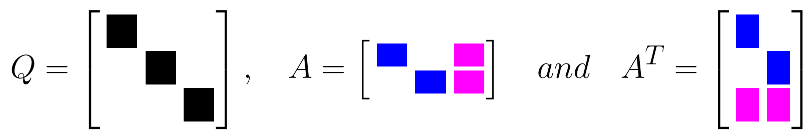
Primal Block-Angular Structure:

$Q =$  , $A =$  *and* $A^T =$ 

Reorder blocks: $\{1, 3;\ 2, 4;\ 5\}$.

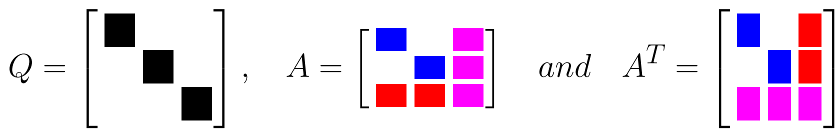
$H =$  , $PHP^T =$ 

Dual Block-Angular Structure:

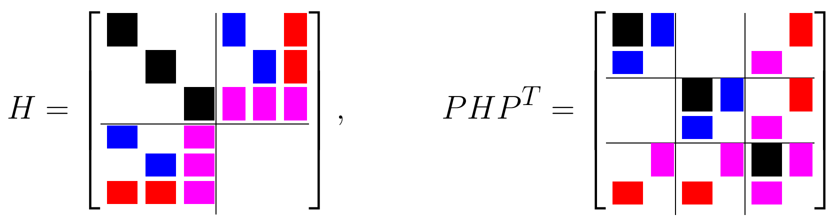


Reorder blocks: $\{1, 4;\ 2, 5;\ 3\}$.

Row & Column Bordered Block-Diag Structure:



Reorder blocks: $\{1, 4;\ 2, 5;\ 3, 6\}$.



Example: Bordered Block-Diagonal Structure

$$\underbrace{\begin{pmatrix}\Phi_1 & & & B_1^\top\\ & \ddots & & \vdots\\ & & \Phi_n & B_n^\top\\ B_1 & \dots & B_n & \Phi_0\end{pmatrix}}_{\Phi} =$$

$$= \underbrace{\begin{pmatrix}L_1 & & & \\ & \ddots & & \\ & & L_n & \\ L_{1,0} & \dots & L_{n,0} & L_0\end{pmatrix}}_{L} \underbrace{\begin{pmatrix}D_1 & & & \\ & \ddots & & \\ & & D_n & \\ & & & D_0\end{pmatrix}}_{D} \underbrace{\begin{pmatrix}L_1^\top & & & L_{1,0}^\top\\ & \ddots & & \vdots\\ & & L_n^\top & L_{n,0}^\top\\ & & & L_0^\top\end{pmatrix}}_{L^\top}$$

The blocks $\Phi_i,\ i = 0, 1, ..., n$ are KKT systems.

Example: Bordered Block-Diagonal Structure

- Cholesky-like factors obtained by Schur-complement:

$$\begin{aligned}\Phi_i &= L_i D_i L_i^\top\\ L_{i,0} &= B_i L_i^{-\top} D_i^{-1}, \quad i = 1..n\\ C &= \Phi_0 - \textstyle\sum_{i=1}^n L_{i,0} D_i L_{i,0}^\top \;=\; L_0 D_0 L_0^\top\end{aligned}$$

- And the system $\Phi x = b$ is solved by

$$\begin{aligned}z_i &= L_i^{-1} b_i\\ z_0 &= L_0^{-1}(b_0 - \textstyle\sum L_{i,0} z_i)\\ y_i &= D_i^{-1} z_i\\ x_0 &= L_0^{-\top} y_0\\ x_i &= L_i^{-\top}(y_i - L_{i,0}^\top x_0)\end{aligned}$$

- Operations (Cholesky, Solve, Product) performed on sub-blocks

Abstract Linear Algebra for IPMs

Execute the operation

"solve (reduced) KKT system"

in IPMs for LP, QP and NLP.

It works like the "backslash" operator in MATLAB.

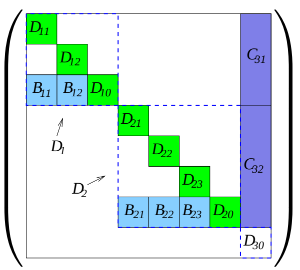
Assumptions:

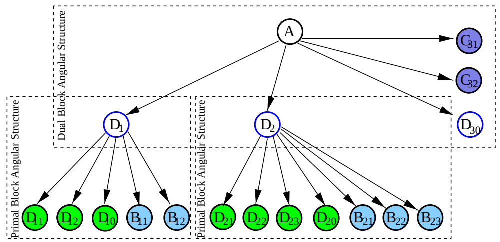
Q and A are block-structured

Linear Algebra of IPMs

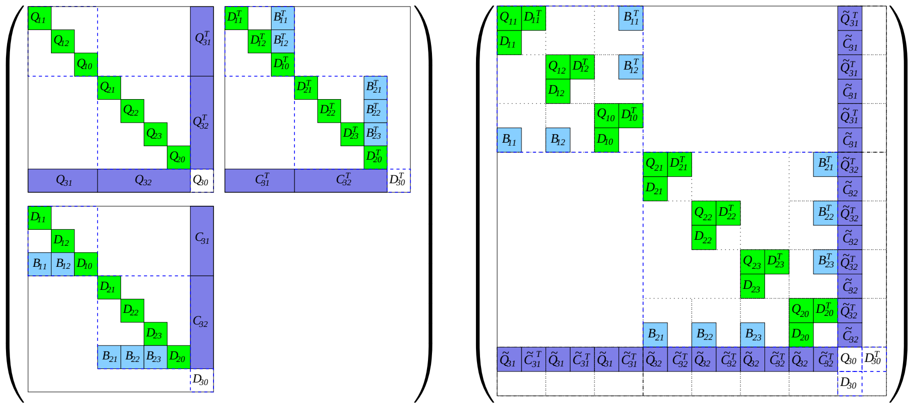
$$\underbrace{\begin{bmatrix} -Q-\Theta_P^{-1} & A^\top \\ A & \Theta_D \end{bmatrix}}_{\Phi\,(NLP)} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ d \end{bmatrix}$$

Tree representation of matrix A:





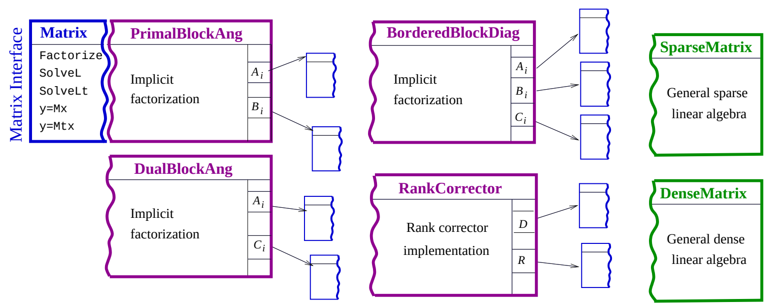
Structures of A and Q imply structure of Φ:



$$\begin{pmatrix} Q & A^\top \\ A & 0 \end{pmatrix} \qquad P\begin{pmatrix} Q & A^\top \\ A & 0 \end{pmatrix}P^{-1}$$

OOPS: Object-oriented linear algebra for IPM

- Every node in the *block elimination tree* has its own linear algebra implementation (depending on its type)
- Each implementation is a realisation of an abstract linear algebra interface.
- Different implementations are available for different structures



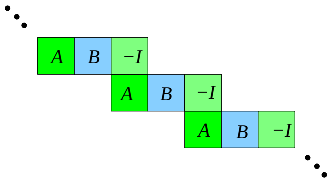
⇒ Rebuild *block elimination tree* with matrix interface structures

Structured Problems

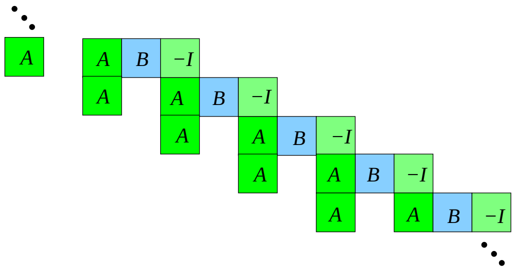
... are present everywhere.

Sources of Structure

Dynamics → Staircase structure



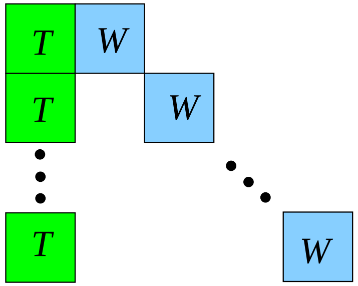
$x_{t+1} = A_t x_t + B_t u_t$



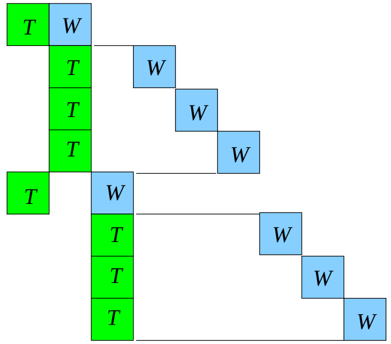
$x_{t+1} = A_t^{t+1} x_t + \ldots + A_{t-p}^{t+1} x_{t-p} + B_t u_t$

Sources of Structure

Uncertainty → Block-angular structure



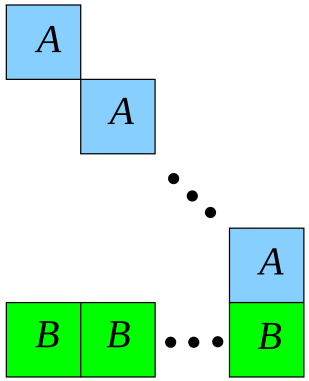
$T_i x^1 + W_i y_i = b_i$



$T_{l_t} x_{a(l_t)} + W_{l_t} x_{l_t} = b_{l_t}$

Sources of Structure

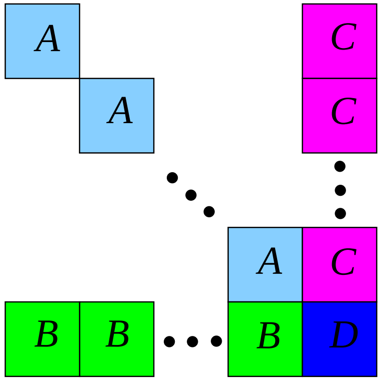
Common resource constraint
$\sum_{i=1}^{k} B_i x_i = b$ → **Dantzig-Wolfe structure**



Sources of Structure

Other types of **near-separability**

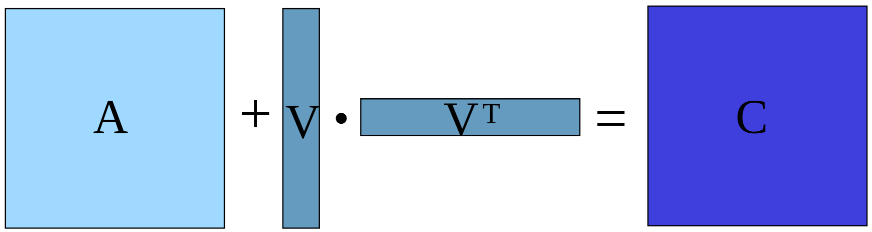
$\rightarrow$ **Row and column bordered block-diagonal structure**



Sources of Structure

(low) **rank-corrector**

$A + VV^T = C$



and networks, ODE- or PDE-discretizations, etc.

Applications:

- **financial planning problems (nonlinear risk measures)**
- **utility distribution planning**
- **data mining (nonlinear kernels in SVMs)**
- **PDE-constrained optimization**

Financial Planning Problems (ALM)

- A set of assets $\mathcal{J} = \{1, ..., J\}$ given (bonds, stock, real estate)
- At every stage $t = 0, ..., T-1$ we can buy or sell different assets
- The return of asset j at stage t is *uncertain*

Investment decisions: **what to buy or sell, at which time stage**

Objectives:

- maximize the final wealth
- minimize the associated risk

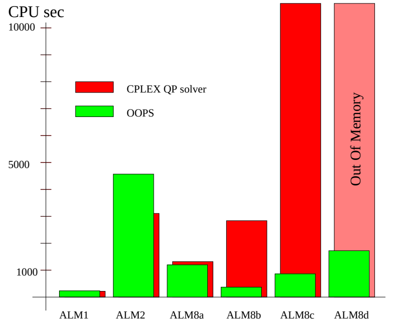
$\Rightarrow$ Mean Variance formulation: $\max \mathbb{E}(X) - \rho \mathrm{Var}(X)$

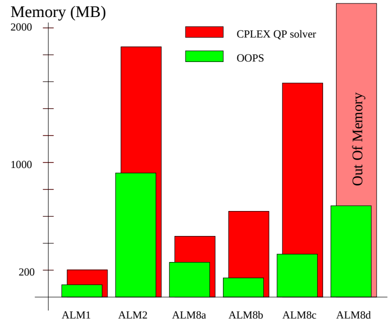
$\Rightarrow$ Stochastic Program: $\Rightarrow$ formulate deterministic equivalent

- standard QP, but huge
- extentions: **nonlinear risk measures** (log utility, skewness)

OOPS vs. CPLEX 7.0 (convexified QPs)

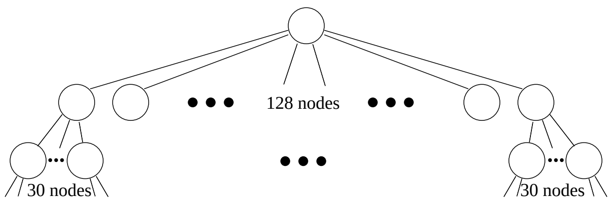
Pb	Stochastic Data			Dimensions		CPLEX 7.0			OOPS		
	Stgs	A	Nodes	Rows	Cols	time	iter	Mem	time	iter	Mem
2	6	5	111111	667K	1667K	3107	51	1859	4570	26	922
8a	4	50	1111	57K	167K	1317	29	452	1196	14	258
8b	3	50	1123	57K	168K	2838	31	637	368	16	142
8c	3	50	2552	130K	383K	10910	29	1590	860	16	319
8d	3	50	4971	254K	746K	51000*	30*	OoM	1723	17	678



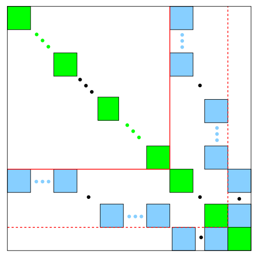
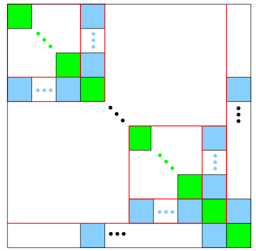


ALM: Largest Problem Attempted

- Optimization of 21 assets (stock market indices) 7 time stages.
- Using multistage stochastic programming
  Scenario tree geometry: 128-30-16-10-5-4 $\Rightarrow$ 16M scenarios.
- 3840 second level nodes with 350.000 variables each.
- Scenario Tree generated using geometric Brownian motion.
- $\Rightarrow$ 1.01 billion variables, 353 million constraints



Sparsity of Linear Algebra

-  $\Rightarrow$
  - $63 + 128 \times 63 = 8127$ columns for Schur-complement
  - Prohibitively expensive
-  $\Rightarrow$
  - Need facility to exploit nested structure
  - Need to be careful that Schur-complement calculations stay sparse on second level

Results (ALM: Mean-Variance QP formulation):

Prob	Stgs	Asts	Scen	Rows	Cols	iter	time	procs	machine
ALM8	7	6	13M	64M	154M	42	3923	512	BlueGene
ALM9	7	14	6M	96M	269M	39	4692	512	BlueGene
ALM10	7	13	12M	180M	500M	45	6089	1024	BlueGene
ALM11	7	21	16M	353M	1.011M	53	3020	1280	HPCx

The problem with

- **353 million of constraints**
- **1 billion of variables**

was solved in 50 minutes using 1280 procs.

Equation systems of dimension **1.363 billion** were solved with the direct (implicit) factorization.

$\longrightarrow$ One IPM iteration takes less than a minute.

Distribution Planning Prob: Deterministic Case

$$\min \sum_{t\in\mathcal{T}} (c_t^T x_t + p_t^T \phi_t) + \sum_{s\in\mathcal{S}} c_s^T \bar{x}_s + p_0^T \phi_0$$

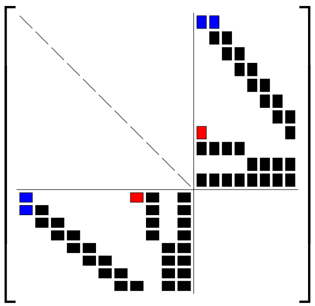
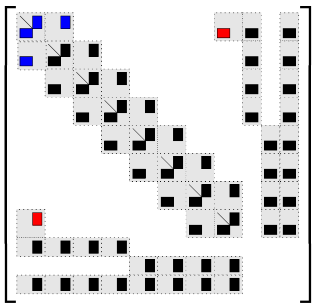
$$\text{s.t.}\quad A x_t + \sum_{\tau=1}^{\bar{\tau}} B^{(-\tau)} x_{t-\tau} + Q_s^T \phi_0 + Q_s^T \phi_t = d_t \qquad t\in\mathcal{T}$$

$$x_t \le \bar{x}_s \qquad t\in S(s), s\in\mathcal{S}$$

$$\left[\begin{array}{ccccccccccccc}
A & Q^T & & & & & & & B & & & & Q^T \\
I & & & & & & & & & & -I & & \\
B & & A & Q^T & & & & & & & & & Q^T \\
 & & I & & & & & & & & -I & & \\
 & & B & & \ddots & \ddots & & & & & & & \vdots \\
 & & & & \ddots & & & & & & & & \\
 & & & & \ddots & & A & Q^T & & & & & Q^T \\
 & & & & & & I & & & & & -I & \\
 & & & & & & B & & A & Q^T & & & Q^T \\
 & & & & & & & & I & & & -I &
\end{array}\right].$$

Deterministic Case (continued)

A cyclic dynamic structure with a "dense" column border block.
Apply the symmetric reordering to augmented system matrix H:
The 19 rows and columns are in the order:
$\{1, 12, 2, 13, 3, 14, 4, 15, 5, 16, 6, 17, 7, 18, 8, 19;\ 9, 10, 11\}$

$H=$  , $PHP^T=$ 

which is again of cyclic bordered structure.

Distribution Planning Prob: Stochastic Case

$$\min\ \mathbb{E}_\xi \left(\sum_{t\in\mathcal{T}} (c_t(\xi)^T x_t(\xi) + p_t(\xi)^T \phi_t(\xi)) + p_0(\xi)^T \phi_0(\xi) \right) + \sum_{s\in\mathcal{S}} c_s^T \bar{x}_s$$

$$\text{s.t.}\quad A x_t(\xi) + \sum_{\tau=1}^{\bar{\tau}} B^{(-\tau)} x_{t-\tau}(\xi) + Q_s^T \phi_0(\xi) + Q_s^T \phi_t(\xi) = d_t(\xi),\ t\in\mathcal{T}$$

$$x_t(\xi) \le \bar{x}_s,\ t\in S(s), s\in\mathcal{S},$$

where x_t, ϕ_t and ϕ_0 are recourse variables.
Assume that the distribution of ξ is discrete.

$$\min \sum_i \pi_i \left(\sum_{t\in\mathcal{T}} ({c_t^i}^T x_t^i + {p_t^i}^T \phi_t^i) + {p_0^i}^T \phi_0^i \right) + \sum_{s\in\mathcal{S}} c_s^T \bar{x}_s$$

$$\text{s.t.}\quad A x_t^i + \sum_{\tau=1}^{\bar{\tau}} B^{(-\tau)} x_{t-\tau}^i + Q_s^T \phi_0^i + Q_s^T \phi_t^i = d_t^i,\ t\in\mathcal{T},\ i\in\mathcal{I}$$

$$x_t^i \le \bar{x}_s,\ t\in S(s), s\in\mathcal{S},\ i\in\mathcal{I}.$$

Distribution Planning Problems

Prob	variables	constraints	periods	nodes	arcs	scenarios
D1Yl	850,324	484,355	365d	321	763	det
D1Yn	850,324	484,355	365d	321	763	det
D7Yn	5,880,190	3,390,485	2555d	321	763	det
S7	459,980	341,640	365d	7	10	36
S321	4,939,945	3,390,485	365d	321	763	7

Memory Requirements: CPLEX 9.1 vs. OOPS

Prob	Cplex 9.1				OOPS	
	time(s)	IPM iters	memory	$nz(LDL^T)$	memory	$nz(LDL^T)$
D1Yl	1448	60 (1e-4)	917MB	62 mln	388MB	8.9 mln
D1Yn	894	49 (1e-4)	808MB	49 mln	372MB	7.3 mln
D7Yn	-	-	OoM	**594** mln	3410MB	**54.7** mln
S7	161	162 (1e-3)	262MB	2.6 mln	184MB	1.5 mln
S321	-	-	OoM	**530** mln	2270MB	**45.3** mln

Performance of OOPS on large problems

Prob	div	time		memory	IPM iters	time/iter
D7Yn	7	921m	(2 proc)	3.5GB	77 (1e-4)	11.96m
	14	1161m	(2 proc)	3.4GB	79 (1e-4)	14.7m
	35	1260m	(2 proc)	3.4GB	84 (1e-4)	15.00m
S321	7	1223m		2.3GB	162 (1e-4)	7.5m
	14	1488m		2.2GB	166 (1e-4)	9.0m
	35	1318m		2.3GB	163 (1e-4)	8.0m

Parallel runs of OOPS

Prob	div	Procs	Speed-up
D7Yn	7	2	2.0
D7Yn	35	5	3.8
S321	7	7	3.9
S321	14	7	4.8

Support Vector Machines:

Formulated as the (dual) quadratic program:

$$\begin{array}{ll} \min & -e^T y + \frac{1}{2} y^T K y, \\ \text{s.t.} & d^T y = 0, \\ & 0 \le y \le \lambda e. \end{array}$$

Ferris & Munson, *SIOPT* 13 (2003) 783-804.

Kernel function $K(x,z) = \langle \phi(x), \phi(z) \rangle$,
where ϕ is a (nonlinear) mapping from X to feature space F
Matrix K: $K_{ij} = K(x_i, x_j)$

Linear Kernel $K(x,z) = x^T z.$
Polynomial Kernel $K(x,z) = (x^T z + 1)^d.$
Gaussian Kernel $K(x,z) = e^{-\gamma \|x-z\|^2}.$

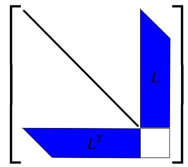
SVMs with Nonlinear Kernels:

K is very large and dense!
Approximate:

$$K \approx LL^T \quad \text{or} \quad K \approx LL^T + D$$

Introduce $v = L^T y$ and get a separable QP:

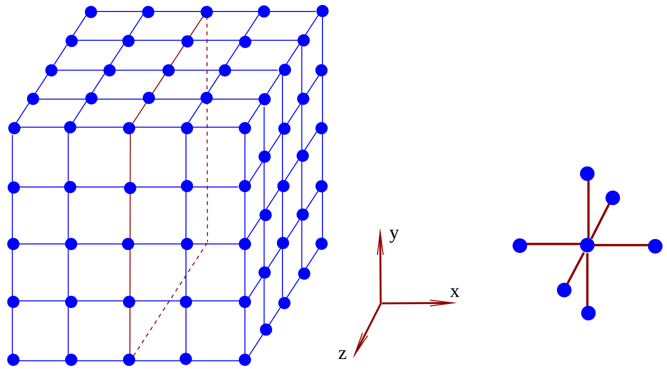
$$\begin{array}{ll} \min & -e^T y + \frac{1}{2} v^T v + \frac{1}{2} y^T D y, \\ \text{s.t.} & d^T y = 0, \\ & v - L^T y = 0, \\ & 0 \le y \le \lambda e. \end{array}$$

$H =$ 

Structure can be exploited in:

- **Linear Algebra of IPM**
  $\longrightarrow$(talk by **Kristian Woodsend** earlier today)

PDE-constrained problems



- grids may be irregular
- boundary conditions need to be taken into account

Domain decomposition

- **3D case:** n^3 **grid points**
- **"remove"** $\mathcal{O}(n^2)$ **points to split the grid into 2, 4, ... subsets each with** $n^3/2,\ n^3/4, \ldots$ **points**

$$\begin{pmatrix} G_1 & & S_1^\top \\ & G_2 & S_2^\top \\ S_1 & S_2 & S_0 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} G_1 & & S_1^\top & & & & S_{10}^\top \\ & G_2 & S_2^\top & & & & S_{20}^\top \\ S_1 & S_2 & S_{12} & & & & S_{xx}^\top \\ & & & G_3 & & S_3^\top & S_{30}^\top \\ & & & & G_4 & S_4^\top & S_{40}^\top \\ & & & S_3 & S_4 & S_{34} & S_{xx}^\top \\ S_{10} & S_{20} & S_{xx} & S_{30} & S_{40} & S_{xx} & S_{00} \end{pmatrix}$$

Conclusions:

Interior Point Methods

$\rightarrow$ **are well-suited to Large Scale Optimization**

Direct Methods

$\rightarrow$ **are well-suited to structure exploitation**

OOPS: Object-Oriented Parallel Solver

`http://www.maths.ed.ac.uk/~gondzio/parallel/solver.html`

$\Rightarrow$ problems of size $\mathbf{10^6}, \mathbf{10^7}, \mathbf{10^8}, \mathbf{10^9}, \ldots$
G. & Sarkissian, *MP* 96 (2003) 561-584.
G. & Grothey, *SIOPT* 13 (2003) 842-864.
G. & Grothey, *AOR* 152 (2007) 319-339.
G. & Grothey, *EJOR* 181 (2007) 1019-1029.