Primal-Dual Pair of Linear Programs

Primal	Dual
$\min \quad c^T x$	$\max \quad b^T y$
$\text{s.t.} \quad Ax = b,$	$\text{s.t.} \quad A^T y + s = c,$
$x \geq 0;$	$s \geq 0.$

Lagrangian

$$L(x, y) = c^T x - y^T (Ax - b).$$

Optimality Conditions

$$\begin{aligned} Ax &= b, \\ A^T y + s &= c, \\ XSe &= 0, \quad (\text{ i.e., } x_j \cdot s_j = 0 \quad \forall j), \\ x &\geq 0, \\ s &\geq 0, \end{aligned}$$

where $X = diag\{x_1, \cdots, x_n\}$, $S = diag\{s_1, \cdots, s_n\}$ and $e = (1, 1, \cdots, 1) \in \mathcal{R}^n$.

Complementarity

Recall that the **Simplex Method** works with a partitioned formulation:

LP constraint matrix	$A = [B, N], \quad B$ is nonsingular
primal variables	$x = (x_B, x_N),$
reduced costs	$s = (s_B, s_N).$

The simplex method maintains the complementarity of primal and dual solutions:

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

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

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

School of Mathematics

Massively Parallel Implementation of Interior Point Methods for Very Large Scale Optimization

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

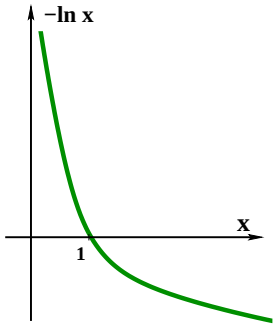
joint work with **Andreas Grothey**

Outline

- Optimality Conditions for LP
- Simplex Method vs Interior Point Method
- IPM Framework: LP, QP, NLP
- Features of Logarithmic Function
- From Sparse to Block-Sparse Problems
- Structured Very Large Optimization Problems
- Object-Oriented Parallel Solver
- Financial Planning Problems: Asset/Liability Models
- Conclusions

Logarithmic barrier $-\ln x_j$

"replaces" the inequality $x_j \geq 0$.



Observe that

$$\min \ e^{-\sum_{j=1}^{n} \ln x_j} \iff \max \prod_{j=1}^{n} x_j$$

The minimization of $-\sum_{j=1}^{n} \ln x_j$ is equivalent to the maximization of the product of distances from all hyperplanes defining the positive orthant: it prevents all x_j from approaching zero.

Use Logarithmic Barrier

Primal Problem

$$\begin{array}{rl} \min & c^T x \\ \text{s.t.} & Ax = b, \\ & x \geq 0; \end{array}$$

Dual Problem

$$\begin{array}{rl} \max & b^T y \\ \text{s.t.} & A^T y + s = c, \\ & s \geq 0. \end{array}$$

Primal Barrier Problem

$$\begin{array}{rl} \min & c^T x - \sum_{j=1}^{n} \ln x_j \\ \text{s.t.} & Ax = b, \end{array}$$

Dual Barrier Problem

$$\begin{array}{rl} \max & b^T y + \sum_{j=1}^{n} \ln s_j \\ \text{s.t.} & A^T y + s = c, \end{array}$$

What's wrong with the Simplex Method?

A **vertex** is defined by a set of n equations:

$$\begin{bmatrix} B & N \\ 0 & I_{n-m} \end{bmatrix} \begin{bmatrix} x_B \\ x_N \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

The linear program with m constraints and n variables ($n \geq m$) has at most

$$N_V = \begin{pmatrix} \mathrm{n} \\ \mathrm{m} \end{pmatrix} = \frac{n!}{m!(n-m)!}$$

vertices and the simplex method can make a non-polynomial number of iteratio to reach the optimality.

V. Klee and G. Minty's example LP: simplex method needs 2^n iterations

How good is the simplex algorithm,
in: Inequalities-III, O. Shisha, ed., Academic Press, 1972, 159–175.

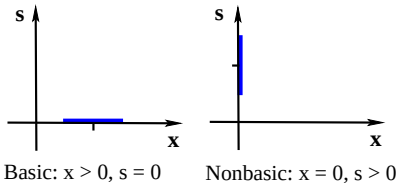
First Order Optimality Conditions

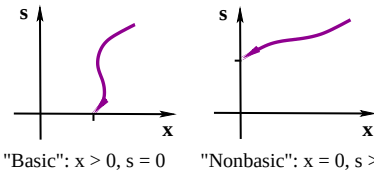
Simplex Method:

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

Interior Point Method:

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





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!

IPM for QP

$$\begin{array}{lcl} \min & c^Tx+\frac{1}{2}x^T\mathbf{Q}\,x & \rightarrow \\ \text{s.t.} & Ax=b, & \\ & x\geq 0. & \end{array} \quad \begin{array}{lcl} \min & c^Tx+\frac{1}{2}x^T\mathbf{Q}\,x-\mu\sum_{j=1}^n \ln x_j \\ \text{s.t.} & Ax=b, \end{array}$$

The first order conditions (for the barrier problem)

$$\begin{aligned} Ax &= b, \\ A^Ty+s-\mathbf{Q}x &= c, \\ XSe &= \mu e. \end{aligned}$$

Newton direction

$$\begin{bmatrix} A & 0 & 0 \\ -\mathbf{Q} & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} \xi_p \\ \xi_d \\ \xi_\mu \end{bmatrix} = \begin{bmatrix} b-Ax \\ c-A^Ty-s+\mathbf{Q}x \\ \mu e-XSe \end{bmatrix}.$$

Augmented system

$$\begin{bmatrix} -\mathbf{Q}-\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \xi_d - X^{-1}\xi_\mu \\ \xi_p \end{bmatrix}.$$

IPM for NLP

$$\begin{array}{lcl} \min & f(x) & \rightarrow \\ \text{s.t.} & g(x)+z=0 & \\ & z\geq 0. & \end{array} \quad \begin{array}{lcl} \min & f(x)-\mu\sum_{i=1}^m \ln z_i \\ \text{s.t.} & g(x)+z=0, \end{array}$$

Lagrangian: $L(x,y,z,\mu)=f(x)+y^T(g(x)+z)-\mu\sum_{i=1}^m \ln z_i.$

The first order conditions (for the barrier problem)

$$\begin{aligned} \nabla f(x)+\nabla g(x)^Ty &= 0, \\ g(x)+z &= 0, \\ YZe &= \mu e. \end{aligned}$$

Newton direction

$$\begin{bmatrix} Q(x,y) & A(x)^T & 0 \\ A(x) & 0 & I \\ 0 & Z & Y \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} -\nabla f(x)-A(x)^Ty \\ -g(x)-z \\ \mu e-YZe \end{bmatrix}.$$

Augmented system

$$\begin{bmatrix} Q(x,y) & A(x)^T \\ A(x) & -ZY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} -\nabla f(x)-A(x)^Ty \\ -g(x)-\mu Y^{-1}e \end{bmatrix} \quad \text{where} \quad \begin{aligned} A(x) &= \nabla g \\ Q(x,y) &= \nabla^2_{xx}L \end{aligned}$$

Primal Barrier Program:

$$\begin{array}{lcl} \min & c^Tx-\mu\sum_{j=1}^n \ln x_j \\ \text{s.t.} & Ax=b. \end{array}$$

Lagrangian: $L(x,y,\mu)=c^Tx-y^T(Ax-b)-\mu\sum_{j=1}^n \ln x_j,$

Stationarity: $\nabla_x L(x,y,\mu)=c-A^Ty-\mu X^{-1}e=0$

Denote: $s=\mu X^{-1}e,$ i.e. $XSe=\mu e.$

The **First Order Optimality Conditions** are:

$$\begin{aligned} Ax &= b, \\ A^Ty+s &= c, \\ XSe &= \mu e \\ (x,s) &> 0. \end{aligned}$$

Newton Method

The first order optimality conditions for the barrier problem form a large syste of nonlinear equations

$$F(x,y,s)=0,$$

where $F:\mathcal{R}^{2n+m}\mapsto\mathcal{R}^{2n+m}$ is an application defined as follows:

$$F(x,y,s)=\begin{bmatrix} Ax-b \\ A^Ty+s-c \\ XSe-\mu e \end{bmatrix}.$$

Actually, the first two terms of it are *linear*; only the last one, corresponding the complementarity condition, is *nonlinear*.

For a given point (x,y,s) we find the Newton direction $(\Delta x,\Delta y,\Delta s)$ by solvi the system of linear equations:

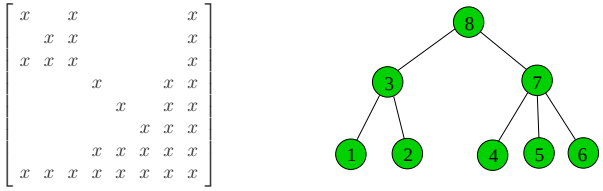
$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} b-Ax \\ c-A^Ty-s \\ \mu e-XSe \end{bmatrix}.$$

OOPS (Object Oriented Parallel Solver)

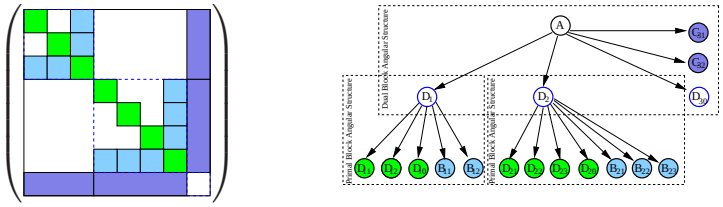
- Mantra: "Truly large scale problems are not only sparse but structured" (due to e.g. dynamics, uncertainty, spatial distribution etc.)
- Exploiting structure is key to building efficient IPMs for large problems:
  - Faster linear algebra
  - Reduced memory use
  - Possibility to exploit (massive) parallelism
  - **We assume that structure is known!** $\Rightarrow$ no automatic detection.
- OOPS currently solves LP/QP problems.
- Simple sequential-QP scheme solves nonlinear ALM models

OOPS: (Block) Elimination Trees:

Elimination tree orders rows/columns for elimination with minimum fill-in:



Elimination Tree can be extended to Block Elimination Tree



$\Rightarrow$ Organisation of linear algebra, Parallelism

Optimality Conditions:

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

Newton Direction:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} \xi_p \\ \xi_d \\ \xi_\mu \end{bmatrix}.$$

Linear Algebra involves an (ill-conditioned) scaling matrix $\Theta = XS^{-1}$.

Augmented System vs Normal Equations

LP	**QP**	**NLP**
$\begin{bmatrix} \Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ d \end{bmatrix}$	$\begin{bmatrix} Q+\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ d \end{bmatrix}$	$\begin{bmatrix} Q(x,y) & A(x)^T \\ A(x) & -ZY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ d \end{bmatrix}$
$(A\Theta A^T)\Delta y = g$	$(A(Q+\Theta^{-1})^{-1}A^T)\Delta y = g$	$(AQ^{-1}A^T + ZY^{-1})\Delta y = g$

Direct Methods: Symmetric LDL^T Factorization

Indefinite	**Quasidefinite**	**Positive Definite**
$H = \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$	$H = \begin{bmatrix} Q & A^T \\ A & -R \end{bmatrix}$	$H = AQ^{-1}A^T$
2×2 pivots needed	1×1 pivots (any sign)	1×1 pivots (positive)
$\begin{bmatrix} 0 & a \\ a & 0 \end{bmatrix}$ and $\begin{bmatrix} 0 & a \\ a & d \end{bmatrix}$	strongly factorizable	easy

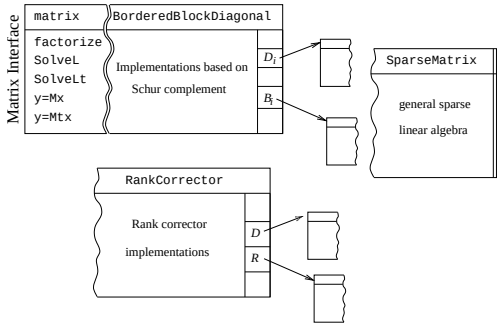
Vanderbei, *SIOPT* (1995): Symmetric QDFM's are strongly factorizable. For any quasidefinite matrix there exists a **Cholesky-like** factorization

$$\bar{H} = LDL^T,$$

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

OOPS: Object-oriented linear algebra implementation

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



⇒ Rebuild *block elimination tree* with matrix interface structures

Application: Asset and Liability Management Problem

- A set of assets $\mathcal{J} = \{1, ..., J\}$ is given (e.g. 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* (but distribution is known).

We have to make investment decisions: what to buy or sell, at which time stage

Objectives:

- maximize the final wealth
- minimize the associated risk

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

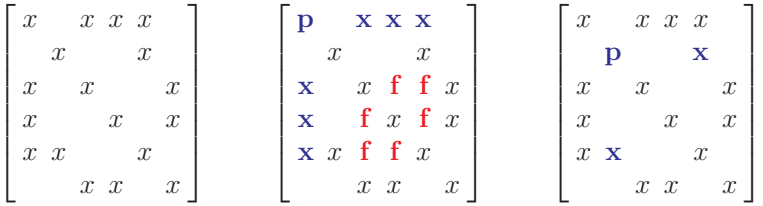
⇒ Stochastic Program:

- Can formulate deterministic equivalent problem
- standard QP, but huge

Minimum Degree Ordering

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

$H =$



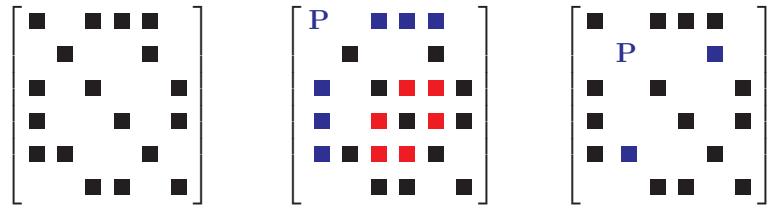
Minimum degree ordering:
choose a diagonal element corresponding to a row with the *min* number of nonzero
Permute rows and columns of H accordingly.

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 *min* numb
of blocks.
Permute block-rows and block-columns of H accordingly.

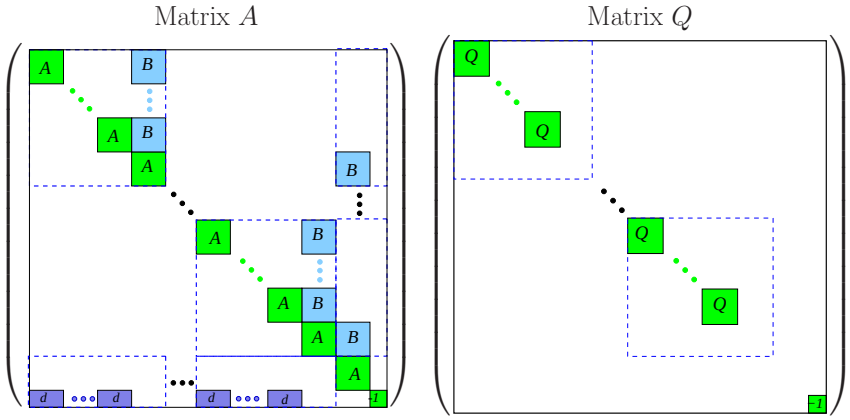
ALM: Largest Problem Attempted

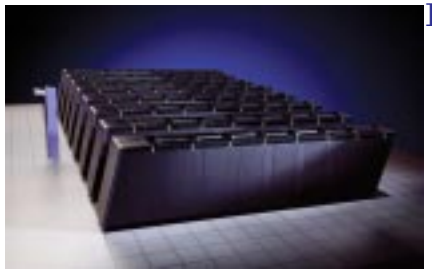
- Optimization of 21 assets (stock market indices) over 7 time stages.
- Using multistage stochastic programming
  Scenario tree geometry: 128-30-16-10-5-4 $\Rightarrow$ 16 million scenarios.
- Scenario Tree generated using geometric Brownian motion.
- $\Rightarrow$ 1.01 billion variables, 353 million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

Issues for Massive Parallelism

- Sparsity of multilevel linear Algebra
- Memory Management

ALM: Structure of matrices A and Q:





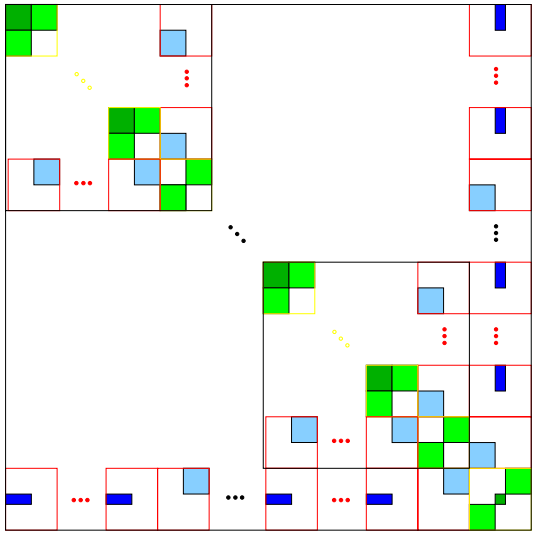
BlueGene (Edinburgh, Scotland)

- 2048 Processors
- 0.7GHz, 256Mb
- 4.7 TFlops
- #64 in `top500.org` list

HPCx (Daresbury, England)

- 1600 IBM Power-4 Processors
- 1.7GHz, 800Mb
- 6.2 TFlops
- #45 in `top500.org` list

ALM: Structure of Augmented System matrix:

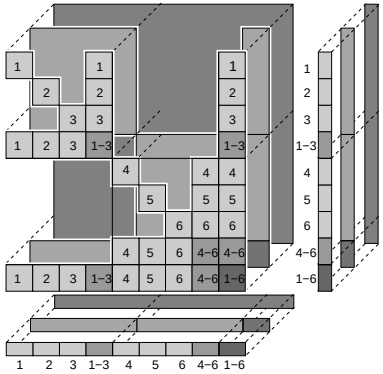


Memory Management

- Data for problem requires > 1GB of memory.
  ⇒ need to split information between processors
- To each node in block-elimination tree a set of processors is assigned
- Linear Algebra is implemented so that processors communicate when needed

Distribution of leading matrix blocks among processors implies

- Distribution of subordinate blocks
- Distribution of row/column vector contributions



Results (ALM: Mean-Variance QP formulation):

Problem	Stages	Blk	Assets	Scenarios	Constraints	Variables	iter	time	procs	machine
ALM8	7	128	6	12.831.873	64.159.366	153.982.477	42	3923	512	BlueGene
ALM9	7	64	14	6.415.937	96.239.056	269.469.355	39	4692	512	BlueGene
ALM10	7	128	13	12.831.873	179.646.223	500.443.048	45	6089	1024	BlueGene
ALM11	7	128	21	16.039.809	352.875.799	1.010.507.968	53	3020	1280	HPCx

The problem with

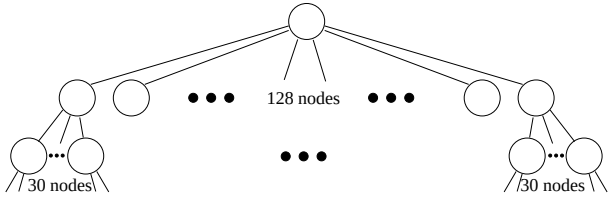
- **353 million of constraints**
- **1 billion of variables**

solved in 50 minutes using 1280 procs.

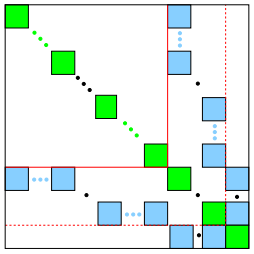
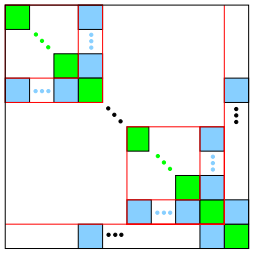
Sparsity of Linear Algebra I

- In ALM problems matrices up to ≈ 500.000 − 1.000.000 variables can l treated as unstructured sparse matrices
- Problem has: – 128 first level nodes with 10.000.000 variables each.
  – 3840 second level nodes with 350.000 variables each.

  ⇒ need to decompose problem at second level
  (with 1280 processors ⇒ 3 blocks per processor)



Sparsity of Linear Algebra II

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

Thank you for your attention!

Object-Oriented Parallel Solver (OOPS):

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

References:

- J. Gondzio and R. Sarkissian, *Parallel interior point solver for structured linear pr grams*, **Mathematical Programming** 96 (2003) pp 561–584.
- J. Gondzio and A. Grothey, *Reoptimization with the primal-dual interior point metho* **SIAM J. on Optimization** 13 (2003) pp 842-864.
- J. Gondzio and A. Grothey, *Parallel interior point solver for structured quadratic pr grams: application to financial planning problems*, Tech. Rep. MS-03-001, School Maths, University of Edinburgh, April 2003 (to appear in **Annals of OR**).
- J. Gondzio and A. Grothey, *Solving nonlinear portfolio optimization problems with t primal-dual interior point method*, Tech. Rep. MS-04-001, School of Maths, University Edinburgh, May 2004.
- J. Gondzio and A. Grothey, *Exploiting structure in parallel implementation of interi point methods for optimization*, Tech. Rep. MS-04-004, School of Maths, University Edinburgh, December 2004.

Conclusions:

- Interior Point Methods are the key optimization technique.
- The theory of IPMs is well understood.
- IPMs demonstrate spectacular efficiency.
- Today IPMs can solve problems of dimension $\mathbf{10^9}$.

IPMs are well-suited to exploit parallelism