Interior point methods: Exploiting sparsity

Part 2
- Interior point methods: exploiting sparsity when solving \((A\Theta A^T)x = b\)
  - Using direct methods
  - Using iterative methods
- Exploiting problem structure
  - Network structure
  - Row-linked block angular problems
  - Column-linked block angular problems

Features
- \(A\) has full rank and \(\Theta\) is diagonal with positive entries
- \(G = A\Theta A^T\) is symmetric and positive definite since
  \[ x^T G x = x^T A\Theta A^T x = \sum_{i=1}^{n} \theta_i[A^T x]_i^2 \geq 0 \text{ with equality iff } A^T x = 0 \text{ iff } x = 0 \]
- Very large range of values in \(\Theta\) so \(G\) is ill-conditioned
Interior point methods: Exploiting sparsity with direct methods

- Form $G = A\Theta A^T$
  - Since $[G]_{ij} = \sum_{k=1}^{n} a_{ik} \theta_k a_{kj}$ sparsity will be lost in forming $G$
  - Since $G = \sum_{k=1}^{n} a_{ik} \theta_k a_{kj}^T$ a single full column in $A$ makes $G$ full

- Form Cholesky decomposition $LL^T = G$
  - $L$ is well defined without permutations
  - As with LU decomposition, pivoting for sparsity is valuable
  - Use special case of Markowitz to identify permutation $P$ so sparsity of $L$ in $LL^T = PGP^T$ is good

- Solve $Gx = b$ as
  $$Ly = Pb \quad \text{then} \quad L^Tz = y \quad \text{and} \quad x = P^Tz$$

Sounds good: does it work?
- Approximate solution of $Gx = b$ is obtained in a small number of iterations...
  - if eigenvalues of $G$ lie in a corresponding number of clusters
- Very rare for this to occur as a natural consequence of the class of LP
- Generally necessary to precondition the system $Gx = b$
  - Identify a matrix $P$ so $\tilde{G} = P^{-1}GP^{-T}$ has the desired spectral property
  - System becomes $\tilde{G}(P^T x) = P^{-1}b$
  - Need to form $P^{-1}GP^{-T}z$ so consider the cost of forming $P$ and operating with $P^{-1}$
  - $P$ is some approximation to the Cholesky matrix $L$ so that $P^{-1}GP^{-T} \approx I$
- In problems solved successfully via matrix-free IPM $P$ contains only a very few columns of $L$

Exploiting structure in convex optimization

- For some LP problems, memory required to form $L$ (and the computation required) may be prohibitive
- Recent work has considered iterative methods for solving $Gx = b$
- Referred to as matrix-free methods
- Based on the conjugate gradient method

For $s^{(1)} = b$ repeat, for $k = 1, 2, \ldots$

- $w = Gs^{(k)}$
- $\alpha^{(k)} = r^{(k)} T s^{(k)}/w^T s^{(k)}$
- $x^{(k+1)} = x^{(k)} + \alpha^{(k)} s^{(k)}$
- $r^{(k+1)} = r^{(k)} - \alpha^{(k)} w$
- If $\|r^{(k+1)}\| \leq \epsilon$ then stop
- $\beta^{(k)} = \|r^{(k+1)}\|^2/\|r^{(k)}\|^2$
- $s^{(k+1)} = r^{(k+1)} + \beta^{(k)} s^{(k)}$

- $y = \Theta z$
- $z = A^T s^{(k)}$
- $w = Ay$

- Key feature: $G$ only appears in $w = Gs^{(k)}$
- Form $w = A(\Theta(A^T s^{(k)}))$
- Reduces computation to operations on original sparse data
Exploiting structure in convex optimization

- Problem structure is generally manifested in the constraint matrix
- Many classes of structure, principally
  - Network structure
  - Block-angular structure

\[ A = \begin{bmatrix}
  A_{00} & A_{01} & A_{02} & \ldots & A_{0N} \\
  A_{11} & & & & \\
  & A_{22} & & & \\
  & & & & \\
  & & & A_{NN} & \\
\end{bmatrix}\]

Row-linked block-angular form

\[ A = \begin{bmatrix}
  A_{00} & A_{10} & A_{11} & & \\
  & A_{20} & & & \\
  & & & & \\
  & & & A_{M0} & \\
  & & & & \\
  & & & & A_{MM} \\
\end{bmatrix}\]

Column-linked block-angular form

- Structure can be explicit or hidden
- Exploit structure within standard algorithms or by using dedicated algorithms

Exploiting network structure

Classical network optimization problem is minimum cost network flow

- Problem has
  - \( m + 1 \) nodes, each with supply \( b_i \)
  - \( n \) arcs, each with cost \( c_j \) and non-negative flow \( x_j \) so objective is \( f = \sum_{j=1}^{n} c_j x_j \)

- Constraints: Net flow into each node equals supply
  - Each arc is from one node to another node
  - Each column of the constraint matrix has one +1 and one −1
  - Net supply to network is zero so \( \sum_{j=1}^{m} b_j = 0 \)
  - Sum of all constraints is zero: remove constraint \( m + 1 \)

Exploiting network structure

Classical network optimization problem is minimum cost network flow

- Problem has
  - \( m + 1 \) nodes, each with supply \( b_i \)
  - \( n \) arcs, each with cost \( c_j \) and non-negative flow \( x_j \) so objective is \( f = \sum_{j=1}^{n} c_j x_j \)

- Constraints: Net flow into each node equals supply
- Problem is sparse LP

\[
\text{maximize } f = c^T x \quad \text{subject to } Ax = b, \quad x \geq 0
\]
Exploiting network structure

Basic solution has
- \( n - m \) nonbasic arcs \( x_n = 0 \)
- \( m \) basic arcs \( x_B \)

Basic arcs form a **spanning tree**
- Can solve \( Bx = b \) by traversing tree from leaves
- Corresponds to permuting \( Bx = b \) as \( UQx = Pb \), where \( U \) is upper triangular
  - All leaves have Markowitz count of zero
  - Once pivoted, other leaves are created
Then solve via forward substitution
Triangularisation is guaranteed for all basic solutions

- Pure network optimization problems have specialised algorithms
- Many LPs have partial or hidden network structure due to underlying model
- Simplex basis matrix \( B \) is typically (almost) triangularisable
- Underlying LP is typically **hyper-sparse**
- Simplex may out-perform IPM significantly, even for very large problems

Exploiting row-linked block-angular structure

**Row-linked block-angular LP problems**

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = b \quad x \geq 0
\end{align*}
\]

where
\[
A = \begin{bmatrix}
A_{00} & A_{01} & A_{02} & \ldots & A_{0N} \\
A_{11} & A_{22} & & & \\
& & \ddots & & \\
& & & A_{NN}
\end{bmatrix}
\]

**Structure:**
- The **linking rows** are \([A_{00} \ A_{01} \ \ldots \ A_{0N}]\)
- The **diagonal blocks** are \([A_{11} \ A_{22} \ \ldots \ A_{NN}]\)
- Diagonal blocks can be many or few; dense or sparse

**Origin:**
- Occur naturally in (eg) decentralised planning and multicommodity flow
- Linking rows (constraints) correspond to shared resources
- Without the linking constraints the problem would be \( N \) independent LPs
Row-linked block-angular structure: Dantzig-Wolfe decomposition

General row-linked block-angular LP is

\[
\begin{align*}
\text{minimize} & \quad c_0^T x_0 + c_1^T x_1 + \ldots + c_N^T x_N \\
\text{subject to} & \quad A_{00} x_0 + A_{01} x_0 + \ldots + A_{0N} x_N = b_0 \\
& \quad A_{11} x_1 = b_1 \\
& \quad \vdots \\
& \quad A_{NN} x_N = b_N \\
x_0 \geq 0 & \quad x_1 \geq 0 & \ldots & \quad x_N \geq 0
\end{align*}
\]

Dantzig-Wolfe decomposition algorithm

- Feasible region \( K_i \) for each sub-problem is given by \( A_i x_i = b, x_i \geq 0 \)
- Key observation: Any point in \( K_i \) is given by
  \[ x_i = E_i \theta_i, \quad e^T \theta_i = 1 \quad \theta_i \geq 0 \]
  where \( E_i \) is the matrix of all \( p_i \) extreme points of \( K_i \)

Substituting \( x_i = E_i \theta_i \) in the original problem yields the master problem

\[
\begin{align*}
\text{minimize} & \quad f^T \theta \\
\text{subject to} & \quad G \theta = h, \quad \theta \geq 0
\end{align*}
\]

where

\[
\begin{align*}
\theta &= \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_N \end{bmatrix}, \quad f = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}, \quad h = \begin{bmatrix} b_0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \\
G &= \begin{bmatrix} G_1 & G_2 & \ldots & G_N \\ e^T & e^T & \ldots & e^T \end{bmatrix}
\end{align*}
\]

for \( G_i = A_{0i} E_i \) and \( f_i^T = c_i^T E_i \).

- Master problem has
  - Fewer equations: \( m_0 + N \)
  - Many more variables \( \sum_{i=1}^{N} p_i \)
Row-linked block-angular structure: Dantzig-Wolfe decomposition

Using the revised simplex method to solve

\[
\text{minimize } f^T \theta \quad \text{subject to } G\theta = h, \quad \theta \geq 0
\]

Consider forming the reduced costs \( \hat{f}_N = f_N - N^T G_B^{-T} f_B \)
- Basis matrix \( G_B \) is of dimension \( m_0 + N \) so \( \pi = G_B^{-T} f_B \) is formed cheaply
- However, cannot form \( \hat{f}_N = f_N - G^T N \pi \) since \( G_N \) cannot be known
- **Key trick:** Find the smallest reduced cost \( \hat{f}_j \) for each sub-problem by solving

\[
\text{minimize } (c_j - A_{0j}^T u) x_j - v_j \quad \text{subject to } A_{jj} x_j = b_j, \quad x_j \geq 0
\]

where \( \pi \) is partitioned into \( u \in \mathbb{R}^{m_0} \) and \( v \in \mathbb{R}^N \)
- Each yields an extreme point \( \zeta_j \) to add to the master problem
- Simplex iterations continue until optimality

**Pros:**
- Appealing reduction in problem size
- Immediate scope for parallelism when solving independent sub-problems

**Cons:**
- Uses “most negative reduced cost” rule
- Can build up large numbers of extreme points

**Summary:**
- Can be advantageous on “loosely coupled” problems
- Otherwise, solve BALP problems as single LPs if possible

Leads into **column generation methods** for classes of very large scale (unstructured) LPs

---

Exploiting column-linked block-angular structure

**Column-linked block-angular LP problems**

\[
\text{minimize } c^T x \quad \text{subject to } A x = b, \quad x \geq 0
\]

\[
A = \begin{bmatrix}
A_{00} & A_{01} & \cdots & A_{0M_0} \\
A_{10} & A_{11} & \cdots & A_{1M_1} \\
\vdots & \vdots & \ddots & \vdots \\
A_{M_0} & A_{M_1} & \cdots & A_{MM}
\end{bmatrix}
\]

**Structure:**
- The linking columns are
- The diagonal blocks are
- Diagonal blocks can be many or few; dense or sparse

**Origin:**
- Occur naturally in (eg) stochastic optimization with \( M \) scenarios
- Linking columns (variables) correspond to decisions affecting all scenarios
- Without the linking variables the problem would be \( M \) independent LPs
Exploiting column-linked block-angular structure: Example

Example: Stochastic wind energy generation

- An expected cost energy generation model is
  \[
  \text{minimize } \ c^T x_0 + c(x) \\
  \text{subject to } \ A_0 x_0 + w(x) = b_0 \\
  T x_0 + w(x) = b
  \]

  where the values of the functions \( c \) and \( w \) depend on the stochastic behaviour of wind power generation.

- Sample the stochastic behaviour to generate \( M \) discrete scenarios.

---

Sampling to generate \( M \) discrete scenarios yields the stochastic LP

\[
\begin{align*}
\text{minimize} & \quad c_0^T x_0 + c_1^T x_1 + \ldots + c_M^T x_M \\
\text{subject to} & \quad A_0 x_0 + \ldots + A_M x_M = b_0 \\
& \quad T x_0 + \ldots + W_1 x_1 = b_1 \\
& \quad \ldots \\
& \quad T x_0 + \ldots + W_M x_M = b_M \\
& \quad x_0 \geq 0 \quad x_1 \geq 0 \quad \ldots \quad x_M \geq 0
\end{align*}
\]

- The 12-hour Illinois model with 8,192 scenarios has
  - 463,113,276 variables
  - 486,899,712 constraints

  This is very large scale optimization!

---

Convenient to permute the LP thus:

\[
\begin{align*}
\text{minimize} & \quad c_1^T x_1 + \ldots + c_M^T x_M + c_0^T x_0 \\
\text{subject to} & \quad W_1^T x_1 + \ldots + W_M^T x_M + T_1^T x_0 = b_1 \\
& \quad \ldots \\
& \quad W_M^T x_M + T_M^T x_0 = b_M \\
& \quad x_0 \geq 0 \quad x_1 \geq 0 \quad \ldots \quad x_M \geq 0
\end{align*}
\]

- Inversion of the basis matrix \( B \) is key to revised simplex efficiency.

\[
B = \begin{bmatrix}
W_1^B & T_1^B \\
\vdots & \vdots \\
W_M^B & T_M^B \\
A^B
\end{bmatrix}
\]

- \( W_i^B \) are columns corresponding to \( n_i^B \) basic variables in scenario \( i \)
- \( T_i^B \) are columns corresponding to \( n_0^B \) basic first stage decisions
Exploiting column-linked block-angular structure: Basis matrix inversion

- Eliminate sub-diagonal entries in each $W_i^B$ (independently)
- Apply elimination operations to each $T_i^B$ (independently)
- Accumulate non-pivoted rows from the $W_i^B$ with $A^n$ and complete elimination

Scope for parallelism
- During inversion
  Since GE is applied independent to each $[W_i^B | T_j^T]$
- When solving systems involving $B$ and $B^T$
  Since they are independent subsystems linked by few equations and variables
- Also scope for parallelism when forming $N^T \pi$
  Since $N$ inherits structure from $A$

Exploiting column-linked block-angular structure: Results

- Solved stochastic LP problems with increasing number of scenarios
- The 12-hour Illinois model with 8,192 scenarios has
  - 463,113,276 variables
  - 486,899,712 constraints
- Solved using simplex implementation developed by H and Lubin (2013)
- Run on BlueGene/P at Argonne National Laboratory
- Possibly the largest “real” LP solved using the simplex method

Exploiting column-linked block-angular structure: Generally

- Interior point methods can also exploit column-linked block-angular structure
- The (Nested) Benders decomposition algorithm can be very efficient
- Identified how IMP can exploit sparsity via iterative methods for $(A\Theta A^T)x = b$
- Illustrated how matrix structure can be exploited for
  - Network LP problems: within the simplex method
  - Row-linked block-angular LP problems: using Dantzig-Wolfe decomposition
  - Column-linked block-angular LP problems: within the simplex method
- Only really scratched the surface of what is possible