Structure and matrix sparsity: Part 2
Interior point methods: Exploiting sparsity
Exploiting problem structure

Julian Hall

School of Mathematics
University of Edinburgh
jajhall@ed.ac.uk

25th June 2014

Structure and matrix sparsity: Overview

Part 2

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

Interior point methods: Exploiting sparsity

- Discuss structure and matrix sparsity for general LP problem

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

  Problem has n variables and $m < n$ constraints
- Interior point methods require solution of $(A\Theta A^T)\boldsymbol{x} = \boldsymbol{b}$

Features

- A has full rank and Θ is diagonal with positive entries
- $G = A\Theta A^T$ is **symmetric** and **positive definite** since

$$\boldsymbol{x}^T G\boldsymbol{x} = \boldsymbol{x}^T A\Theta A^T\boldsymbol{x} = \sum_{i=1}^{n} \theta_i [A^T\boldsymbol{x}]_i^2 \geq 0 \text{ with equality iff } A^T\boldsymbol{x} = \boldsymbol{0} \text{ iff } \boldsymbol{x} = \boldsymbol{0}$$

- Very large range of values in Θ so G is **ill-conditioned**

Interior point methods: Exploiting sparsity with direct methods

1. 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 $G = \sum_{k=1}^{n} \boldsymbol{a}_k\theta_k\boldsymbol{a}_k^T$ a single full column in A makes G full
2. 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
3. Solve $G\boldsymbol{x} = \boldsymbol{b}$ as

$$L\boldsymbol{y} = P\boldsymbol{b} \quad \text{then} \quad L^T\boldsymbol{z} = \boldsymbol{y} \quad \text{and} \quad \boldsymbol{x} = P^T\boldsymbol{z}$$

Interior point methods: Exploiting sparsity with iterative methods

- For some LP problems, memory required to form L (and the computation required) may be prohibitive
- Recent work has considered **iterative** methods for solving $G\boldsymbol{x} = \boldsymbol{b}$
- Referred to as **matrix-free** methods — Gondzio (2012)
- Based on the **conjugate gradient method**

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

1. $\boldsymbol{w} = G\boldsymbol{s}^{(k)}$
2. $\alpha^{(k)} = \boldsymbol{r}^{(k)T}\boldsymbol{s}^{(k)} / \boldsymbol{w}^T\boldsymbol{s}^{(k)}$
3. $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha^{(k)}\boldsymbol{s}^{(k)}$
4. $\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha^{(k)}\boldsymbol{w}$
5. If $\|\boldsymbol{r}^{(k+1)}\|_2 \leq \epsilon$ then stop
6. $\beta^{(k)} = \|\boldsymbol{r}^{(k+1)}\|_2^2 / \|\boldsymbol{r}^{(k)}\|_2^2$
7. $\boldsymbol{s}^{(k+1)} = \boldsymbol{r}^{(k+1)} + \beta^{(k)}\boldsymbol{s}^{(k)}$

- Key feature: G only appears in $\boldsymbol{w} = G\boldsymbol{s}^{(k)}$
- Form $\boldsymbol{w} = A(\Theta(A^T\boldsymbol{s}^{(k)}))$ as

$$\begin{aligned} \boldsymbol{z} &= A^T\boldsymbol{s}^{(k)} \\ \boldsymbol{y} &= \Theta\boldsymbol{z} \\ \boldsymbol{w} &= A\boldsymbol{y} \end{aligned}$$

- Reduces computation to operations on original sparse data

Interior point methods: Exploiting sparsity with iterative methods

Sounds good: does it work?

- Approximate solution of $G\boldsymbol{x} = \boldsymbol{b}$ 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 $G\boldsymbol{x} = \boldsymbol{b}$
  - Identify a matrix P so $\widetilde{G} = P^{-1}GP^{-T}$ has the desired spectral property
  - System becomes $\widetilde{G}(P^T\boldsymbol{x}) = P^{-1}\boldsymbol{b}$
  - Need to form $P^{-1}GP^{-T}\boldsymbol{z}$ 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

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

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

Row-linked block-angular form

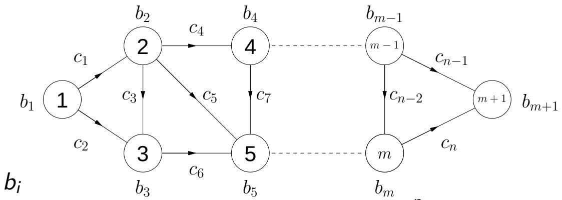
$$A = \begin{bmatrix} A_{00} & & & & \\ A_{10} & A_{11} & & & \\ A_{20} & & A_{22} & & \\ \vdots & & & \ddots & \\ A_{M0} & & & & A_{MM} \end{bmatrix}$$

Column-linked block-angular form

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

Exploiting network structure

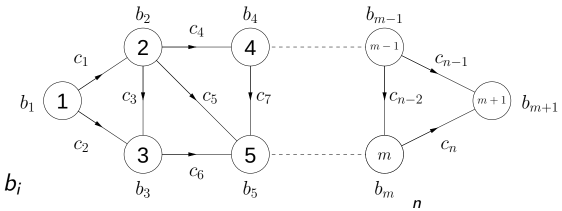
Classical network optimization problem is **minimum cost network flow**



- 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 $\sum_{i=1}^{m+1} b_j = 0$
  - Sum of all constraints is zero: remove constraint $m+1$

Exploiting network structure

Classical network optimization problem is **minimum cost network flow**



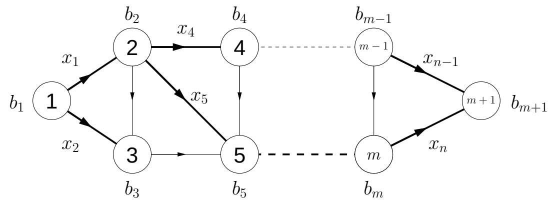
- Problem has
  - $m + 1$ nodes, each with supply b_i
  - n arcs, each with cost c_j and non-negative flow x_j so objective is $f = \sum_{j=1}^{n} c_j x_j$
- **Constraints:** Net flow into each node equals supply
- Problem is **sparse** LP

$$\text{maximize } f = \boldsymbol{c}^T\boldsymbol{x} \quad \text{subject to } A\boldsymbol{x} = \boldsymbol{b}, \quad \boldsymbol{x} \geq \boldsymbol{0}$$

Exploiting network structure

Basic solution has

- $n - m$ **nonbasic arcs** $\boldsymbol{x}_N = \boldsymbol{0}$
- m **basic arcs** $\boldsymbol{x}_B$



Basic arcs form a **spanning tree**

- Can solve $B\boldsymbol{x} = \boldsymbol{b}$ by traversing tree from leaves
- Corresponds to permuting $B\boldsymbol{x} = \boldsymbol{b}$ as $UQ\boldsymbol{x} = P\boldsymbol{b}$, 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

Exploiting network structure

- 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

Row-linked block-angular LP problems

$$\begin{array}{ll} \text{minimize} & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} & A\boldsymbol{x} = \boldsymbol{b} \quad \boldsymbol{x} \geq \boldsymbol{0} \end{array} \qquad A = \begin{bmatrix} A_{00} & A_{01} & A_{02} & \dots & A_{0N} \\ & A_{11} & & & \\ & & A_{22} & & \\ & & & \ddots & \\ & & & & A_{NN} \end{bmatrix}$$

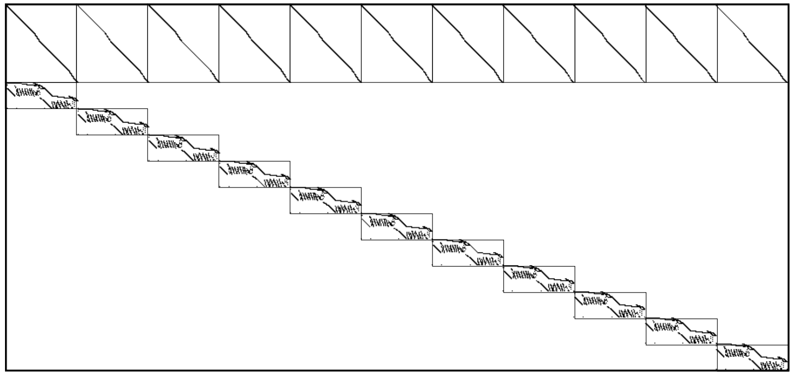
Structure:

- The **linking rows** are $\begin{bmatrix} A_{00} & A_{01} & \dots & A_{0N} \end{bmatrix}$
- The **diagonal blocks** are $\begin{bmatrix} A_{11} & A_{22} & \dots & A_{NN} \end{bmatrix}$
- Diagonal blocks can be many or few; dense or sparse

Origin:

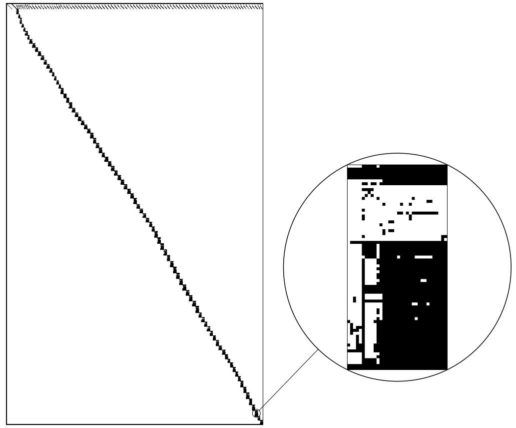
- Occur naturally in (eg) decentralised planning and multicommodity flow
- Linking rows (constraints) correspond to shared resources
- Without the linking constraints the problem would be N independent LPs

Example: pds-02: 3518 rows, 7535 columns, 11 diagonal blocks



Source: Patient Distribution System (multicommodity flow), Carolan *et al.* (1990)

Example: dcp1: 4950 rows, 3007 columns, 87 diagonal blocks



Source: Industrial, Hall (1997)

Row-linked block-angular structure: Dantzig-Wolfe decomposition

General row-linked block-angular LP is

$$\begin{array}{llllllll} \text{minimize} & \boldsymbol{c}_0^T\boldsymbol{x}_0 & + & \boldsymbol{c}_1^T\boldsymbol{x}_1 & + \;\dots\; + & \boldsymbol{c}_N^T\boldsymbol{x}_N & & \\ \text{subject to} & A_{00}\boldsymbol{x}_0 & + & A_{01}\boldsymbol{x}_0 & + \;\dots\; + & A_{0N}\boldsymbol{x}_N & = & \boldsymbol{b}_0 \\ & & & A_{11}\boldsymbol{x}_1 & & & = & \boldsymbol{b}_1 \\ & & & & \ddots & & & \vdots \\ & & & & & A_{NN}\boldsymbol{x}_N & = & \boldsymbol{b}_N \\ & \boldsymbol{x}_0 \geq \boldsymbol{0} & & \boldsymbol{x}_1 \geq \boldsymbol{0} & \dots & \boldsymbol{x}_N \geq \boldsymbol{0} & & \end{array}$$

Dantzig-Wolfe decomposition algorithm

- Feasible region K_i for each sub-problem is given by $A_{ii}\boldsymbol{x}_i = \boldsymbol{b}$, $\boldsymbol{x}_i \geq \boldsymbol{0}$
- **Key observation:** Any point in K_i is given by
$$\boldsymbol{x}_i = E_i\boldsymbol{\theta}_i \qquad \boldsymbol{e}^T\boldsymbol{\theta}_i = 1 \qquad \theta_i \geq \boldsymbol{0}$$
where E_i is the matrix of all p_i **extreme points** of K_i

Row-linked block-angular structure: Dantzig-Wolfe decomposition

- Substituting $\boldsymbol{x}_i = E_i\boldsymbol{\theta}_i$ in the original problem yields the **master problem**
$$\text{minimize } \boldsymbol{f}^T\boldsymbol{\theta} \quad \text{subject to } G\boldsymbol{\theta} = \boldsymbol{h}, \quad \boldsymbol{\theta} \geq \boldsymbol{0}$$
where
$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_2 \\ \vdots \\ \boldsymbol{\theta}_N \end{bmatrix}, \quad \boldsymbol{f} = \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \vdots \\ \boldsymbol{f}_N \end{bmatrix}, \quad \boldsymbol{h} = \begin{bmatrix} \boldsymbol{b}_0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \quad \text{and} \quad G = \begin{bmatrix} G_1 & G_2 & \dots & G_N \\ \boldsymbol{e}^T & & & \\ & \boldsymbol{e}^T & & \\ & & \ddots & \\ & & & \boldsymbol{e}^T \end{bmatrix}$$
for $G_i = A_{0i}E_i$ and $\boldsymbol{f}_i^T = \boldsymbol{c}_i^T E_i$.
- 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

$$\text{minimize } \boldsymbol{f}^T\boldsymbol{\theta} \quad \text{subject to } G\boldsymbol{\theta} = \boldsymbol{h}, \quad \boldsymbol{\theta} \geq \mathbf{0}$$

Consider forming the reduced costs $\widehat{\boldsymbol{f}}_N = \boldsymbol{f}_N - N^T G_B^{-T}\boldsymbol{f}_B$

- Basis matrix G_B is of dimension $m_0 + N$ so $\boldsymbol{\pi} = G_B^{-T}\boldsymbol{f}_B$ is formed cheaply
- However, cannot form $\widehat{\boldsymbol{f}}_N = \boldsymbol{f}_N - G_N^T\boldsymbol{\pi}$ since G_N cannot be known
- **Key trick:** Find the smallest reduced cost $\widehat{f}_j$ for sub-problem i by solving

$$\text{minimize } (\boldsymbol{c}_i - A_{0i}^T\boldsymbol{u})^T\boldsymbol{x}_i - v_i \quad \text{subject to } A_{ii}\boldsymbol{x}_i = \boldsymbol{b}_i, \quad \boldsymbol{x}_i \geq \mathbf{0}$$

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

Row-linked block-angular structure: Dantzig-Wolfe decomposition

- **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

Column-linked block-angular LP problems

$$\begin{array}{ll} \text{minimize} & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} & A\boldsymbol{x} = \boldsymbol{b} \quad \boldsymbol{x} \geq \mathbf{0} \end{array}$$

$$A = \begin{bmatrix} A_{00} & & & & \\ A_{10} & A_{11} & & & \\ A_{20} & & A_{22} & & \\ \vdots & & & \ddots & \\ A_{M0} & & & & A_{MM} \end{bmatrix}$$

Structure:

- The **linking columns** are $\begin{bmatrix} A_{00} \\ A_{01} \\ \vdots \\ A_{M0} \end{bmatrix}$
- The **diagonal blocks** are $\begin{bmatrix} A_{11} & A_{22} & \dots & A_{MM} \end{bmatrix}$
- 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

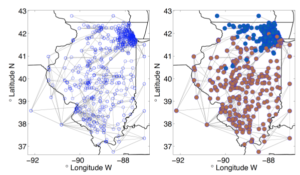
Exploiting column-linked block-angular structure: Example

Example: Stochastic wind energy generation

- An expected cost energy generation model is

$$\begin{array}{lrcll} \text{minimize} & \boldsymbol{c}_0^T\boldsymbol{x}_0 & + & c(\boldsymbol{x}) & \\ \text{subject to} & A_0\boldsymbol{x}_0 & & & = \boldsymbol{b}_0 \\ & T\boldsymbol{x}_0 & + & \boldsymbol{w}(\boldsymbol{x}) & = \boldsymbol{b} \end{array}$$

  where the values of the functions c and $\boldsymbol{w}$ depend on the stochastic behaviour of wind power generation
- Sample the stochastic behaviour to generate M discrete **scenarios**



Illinois power network (2010)

Exploiting column-linked block-angular structure: Example

- Sampling to generate M discrete scenarios yields the **stochastic LP**

$$\begin{array}{llllllll}
\text{minimize} & \boldsymbol{c}_0^T\boldsymbol{x}_0 & + & \boldsymbol{c}_1^T\boldsymbol{x}_1 & + \;\ldots\; + & \boldsymbol{c}_M^T\boldsymbol{x}_M & & \\
\text{subject to} & A_0\boldsymbol{x}_0 & & & & & = & \boldsymbol{b}_0 \\
 & T\boldsymbol{x}_0 & + & W_1\boldsymbol{x}_1 & & & = & \boldsymbol{b}_1 \\
 & \vdots & & & \ddots & & & \vdots \\
 & T\boldsymbol{x}_0 & & & + & W_M\boldsymbol{x}_M & = & \boldsymbol{b}_M \\
 & \boldsymbol{x}_0 \geq \boldsymbol{0} & & \boldsymbol{x}_1 \geq \boldsymbol{0} & \ldots & \boldsymbol{x}_M \geq \boldsymbol{0} & &
\end{array}$$

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

  This is *very* large scale optimization!

Exploiting column-linked block-angular structure: Example

Convenient to permute the LP thus:

$$\begin{array}{llllllll}
\text{minimize} & \boldsymbol{c}_1^T\boldsymbol{x}_1 & + \;\ldots\; + & \boldsymbol{c}_M^T\boldsymbol{x}_M & + & \boldsymbol{c}_0^T\boldsymbol{x}_0 & & \\
\text{subject to} & W_1\boldsymbol{x}_1 & & & + & T_1\boldsymbol{x}_0 & = & \boldsymbol{b}_1 \\
 & & \ddots & & \vdots & & & \vdots \\
 & & & W_M\boldsymbol{x}_M & + & T_M\boldsymbol{x}_0 & = & \boldsymbol{b}_M \\
 & & & & & A\boldsymbol{x}_0 & = & \boldsymbol{b}_0 \\
 & \boldsymbol{x}_1 \geq \boldsymbol{0} & \ldots & \boldsymbol{x}_M \geq \boldsymbol{0} & & \boldsymbol{x}_0 \geq \boldsymbol{0} & &
\end{array}$$

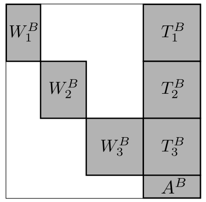
Exploiting column-linked block-angular structure: Example

- Inversion of the basis matrix B is key to revised simplex efficiency

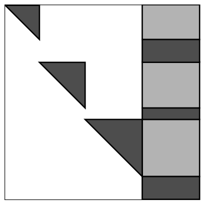
$$B = \begin{bmatrix} W_1^B & & & T_1^B \\ & \ddots & & \vdots \\ & & W_M^B & T_M^B \\ & & & A^B \end{bmatrix}$$

- W_i^B are columns corresponding to n_i^B basic variables in scenario i
- $\begin{bmatrix} T_1^B \\ \vdots \\ T_M^B \\ A^B \end{bmatrix}$ are columns corresponding to n_0^B basic first stage decisions

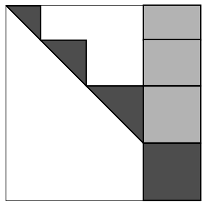
Exploiting column-linked block-angular structure: Basis matrix inversion



- 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^B and complete elimination

Exploiting column-linked block-angular structure: Basis matrix inversion

- Scope for parallelism
  - During inversion
    Since GE is applied independent to each $\left[W_i^B \mid T_i^B\right]$
  - 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 $N^T\boldsymbol{\pi}$
  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

Structure and matrix sparsity: Part 2: Summary

- Identified how IMP can exploit sparsity via iterative methods for $(A\Theta A^T)\boldsymbol{x} = \boldsymbol{b}$
- 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