Lecture 3:
Basis updates, storage schemes, Dantzig-Wolfe Decomposition

Prof. Krishna R. Pattipati
Dept. of Electrical and Computer Engineering
University of Connecticut
Contact: krishna@engr.uconn.edu; (860) 486-2890

February 4, 2016

Outline

- Basis update methods
  - Product-form of the inverse (PFI)
  - Sequential LU update
  - Sequential QR update
- LP with upper and lower bound constraints
- Dantzig-Wolfe decomposition . . . a prelude to duality
- LP and Goal Programming
- Summary

Basis update methods

- Product form of the inverse (PFI)
  - Recall that replacing column l by $\underline{a}_k$ in the basis implies

$$\begin{aligned}\overline{B} &= B + \underline{a}_k \underline{e}_l^T - B\underline{e}_l \underline{e}_l^T \\ &= B + (\underline{a}_k - B\underline{e}_l)\underline{e}_l^T = B\left[I + (\underline{\alpha} - \underline{e}_l)\underline{e}_l^T\right]\end{aligned}$$

where $\underline{\alpha} = B^{-1}\underline{a}_k$

  - So,

$$\begin{aligned}\overline{B}^{-1} &= B^{-1} - \frac{B^{-1}(\underline{a}_k - B\underline{e}_l)\underline{e}_l^T B^{-1}}{1 + \underline{e}_l^T B^{-1}(\underline{a}_k - B\underline{e}_l)} \\ &= B^{-1} - \frac{(\underline{\alpha}_k - \underline{e}_l)\underline{e}_l^T B^{-1}}{1 + \underline{e}_l^T(\underline{\alpha}_k - \underline{e}_l)} \\ &= \left[I - \frac{(\underline{\alpha}_k - \underline{e}_l)\underline{e}_l^T}{\alpha_{lk}}\right]B^{-1} = EB^{-1}, E = I + (\underline{\eta} - \underline{e}_l)\underline{e}_l^T\end{aligned}$$

$$E = \begin{bmatrix} 1 & \cdots & \eta_1 & \cdots \\ & 1 & \eta_2 & \\ & \cdots & \eta_l & \cdots \\ & & \eta_m & 1 \end{bmatrix}$$

$$\text{where } \eta_i = \begin{cases} -\dfrac{\alpha_{ik}}{\alpha_{lk}}; i \neq l \\ \dfrac{1}{\alpha_{lk}};\quad i = l \end{cases}$$

and $E^{-1} = I + (\underline{\alpha} - \underline{e}_l)\underline{e}_l^T$

Sherman-Morrison-Woodbury formula:

$$\overline{A} = A + \underline{a}\underline{b}^T$$

$$\overline{A}^{-1} = A^{-1} - \frac{A^{-1}\underline{a}\underline{b}^T A^{-1}}{1 + \underline{b}^T A^{-1}\underline{a}}$$

Example:

$$B = I; l = 3; \underline{a}_k = \begin{bmatrix} 8 \\ 4 \\ 2 \end{bmatrix} = \underline{\alpha}_k$$

$$\underline{\eta} = \begin{bmatrix} -4 \\ -2 \\ 0.5 \end{bmatrix}$$

$$\overline{B}^{-1} = EB^{-1} = E = \begin{bmatrix} 1 & 0 & -4 \\ 0 & 1 & -2 \\ 0 & 0 & 0.5 \end{bmatrix}$$

Product form of the inverse (PFI)

- *Remarks:*
- E is entirely determined by $\underline{\eta}$ vector

$$E^{-1} = I + (\underline{\alpha} - \underline{e}_l)\underline{e}_l^T$$

$\Rightarrow$ Don't store E. Store only $\underline{\eta}$ and its column position l; record only non-zero entries of $\underline{\eta}$

- Since $B_0 = I$ in phase 1, at k^{th} iteration of simplex

$$B^{-1} = E_k E_{k-1} \cdots E_2 E_1 I_m$$

$\Rightarrow$ this is called product-form of the inverse (PFI), since B^{-1} is expressed as a product of elementary transformations

- Computation of simplex multipliers is called <u>BTRAN</u> (backward transformations)

$$\underline{\lambda}^T = \underline{c}_B^T B^{-1} = \underline{c}_B^T E_k E_{k-1} \cdots E_2 E_1$$

- Consider

$$\underline{c}_B^T E_k = \underline{c}_B^T + (\underline{c}_B^T \underline{\eta} - \underline{c}_B^T \underline{e}_l)\underline{e}_l^T$$

$\Rightarrow (\underline{c}_B)_l \leftarrow \underline{c}_B^T \underline{\eta}$

$\Rightarrow$ Only one element changes at each iteration

- so, BTRAN can be summarized as:

Do $i = k, k-1, \ldots, 1$

$(\underline{c}_B)_{i_l} \leftarrow \langle \underline{c}_B, \eta(i) \rangle$, i_l = column position at iteration l

end Do

FTRAN

- Computation of $\underline{\alpha}_j$ is called FTRAN (forward transformation)

$$\underline{\alpha}_j = B^{-1}\underline{a}_j \Rightarrow \underline{\alpha}_j = E_k E_{k-1} \cdots E_1 \underline{a}_j$$

- Consider

$$\begin{aligned} E_1 \underline{a}_j &= \underline{a}_j + (\underline{\eta} - \underline{e}_l)\underline{e}_l^T \underline{a}_j \\ \underline{\tilde{a}}_j &= \underline{a}_j + (\underline{\eta} - \underline{e}_l) a_{lj} \\ \tilde{a}_{ij} &= a_{ij} + a_{lj}\eta_i, i \neq l \\ \tilde{a}_{ij} &= \eta_l a_{lj}, i = l \end{aligned}$$

- So, FTRAN can be summarized as:

Do $i = 1, 2, \ldots, k$

$\underline{\tilde{a}}_j = \underline{\tilde{a}}_j + (\underline{\eta}(i) - \underline{e}_{l_i})\, a_{l_i j}$

end Do

- Must recompute from scratch a B^{-1} every K steps due to round-off problems and to reduce page faults
- Usually $K = m + 50$
- Also do reinversion whenever the residuals $\|A\underline{x} - \underline{b}\|_2 > 10^{-6}$ or $\|\underline{c}_B - B^T\underline{\lambda}\|_2 > 10^{-6}$
- Compute residuals in double precision

LU factorization of basis

- Finding LU factorization from scratch (e.g., for reinversion)
  - For simplicity assume that B = first m columns of A
  - $B = LU \Rightarrow$ want to determine $m^2 + m$ entries from m^2 entries

  $\Rightarrow$ Can fix either L = unit lower Δ_{lower} or U = upper Δ_{upper}

$$L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \vdots \\ \vdots & \cdots & \ddots & \vdots \\ l_{m1} & l_{m2} & \cdots & 1 \end{bmatrix}, U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ 0 & u_{22} & u_{23} & u_{2m} \\ \vdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & u_{mm} \end{bmatrix}$$

B as sum of outer products of $\underline{l}_k$ and $\underline{u}_k$

$$B = \begin{bmatrix} 1 & 1 \\ 2 & 7 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 5 \end{bmatrix} = LU$$

$$B = LU \Rightarrow$$

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \vdots \\ \vdots & \cdots & \ddots & \vdots \\ l_{m1} & l_{m2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ 0 & u_{22} & u_{23} & u_{2m} \\ \vdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & u_{mm} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \cdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mm} \end{bmatrix}$$

$$\Rightarrow B = \sum_{k=1}^{m} \underline{l}_k \underline{u}_k^T; \;\; \underline{l}_k = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ l_{k+1,k} \\ \vdots \\ l_{mk} \end{bmatrix}; \;\; \underline{u}_k^T = \begin{bmatrix} 0 & \cdots & u_{kk} & u_{k,k+1} & \cdots & u_{km} \end{bmatrix}$$

Column k of L and row k of U on pass k

- The decomposition is done in m passes
- On pass k, we get:
  - u_{kk}
  - column k of L
  - row k of U
- Initially, we start with the first column of B

  $a_{11} = u_{11} \Rightarrow u_{11} = a_{11}/l_{11} = a_{11} \Rightarrow l_{11} = a_{11}/u_{11}$

  $a_{21} = l_{21}u_{11} \Rightarrow l_{21} = a_{21}/u_{11}$

  $\vdots$

  $a_{m1} = l_{m1}u_{11} \Rightarrow l_{m1} = a_{m1}/u_{11}$
- Also, $a_{1j} = u_{1j}l_{11} \Rightarrow u_{1j} = a_{1j}$

  $\Rightarrow$ 1st row of U = 1st row of B
- Finished computing the 1st column of L and 1st row of U
- The sequence of computations is:
  - $u_{11} \rightarrow Diag(U)$; $\underline{l}_1 \rightarrow$ column of L; $\underline{u}_1^T \rightarrow$ remaining row of U;
- <u>Note</u>: a_{i1} and a_{1j} are used once <u>and</u> never again

  $\Rightarrow$ can overwrite $\underline{l}_1$ and $\underline{u}_1^T$ in the 1st column and row of B
- <u>Except for</u> l_{11}, which we know is 1 any way

$$l_{il} \leftarrow a_{il} / a_{11}, \quad i = 2, \ldots, m$$
$$u_{1j} \leftarrow a_{1j}, \quad j = 1, 2, \ldots, m$$

- Problem: what if $a_{11} = 0$?

<u>example</u> $\begin{bmatrix} 0 & 1 \\ 1 & 6 \end{bmatrix}$ <u>nonsingular</u>, but $a_{11} = 0$

Pivoting Idea

1. Compute $l_{i1}, \dots, l_{m1}$ except for division $\Rightarrow l_{i1}u_{11}$
2. Find the largest $|l_{i1}|$ relative to initial row i norm $\Rightarrow \frac{l_{i1}}{\sum_j |a_{ij}|} \forall i$
   - Assume that the maximum occurs in row $r_1 \Rightarrow r_\mathrm{i} = \arg\max_i \frac{l_{i1}}{\sum_j |a_{ij}|}$
3. Swap rows r_1 and 1 in B and L. Let $IP(1) = r_1$
   - What does it mean?
     - Multiply B by

$$P_1^{r1} = \begin{bmatrix} 0 & 0 & \cdots & 1 & 0 \\ 0 & 1 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & \cdots & 0 \end{bmatrix}$$ PERMUTATION MATRIX

On the left

Note: P_1^{r1} is symmetric and orthogonal $(P_1^{r1})^{-1} = P_1^{r1}$

4. Divide throughout by (new) $l_{11} \neq 0$ to get $l_{21}, \dots, l_{m1}$

$$\begin{bmatrix} 1 & 6 \\ 0 & 1 \end{bmatrix} = U$$

5. $u_{11} = l_{11}$ (new). In actuality, l_{11} replace a_{11}

Formalizing LU decomposition

- So, really have found the 1st LU factor, $\underline{l}_1\underline{u}_1^T$ of $P_1^{r_1}B = \tilde{B}$ and not B!!
  - Can we do it recursively? Is it useful? YES!!
- Consider the situation at column $k \geq 2$. Get column k of L and row k of U from column k and row k of $\tilde{B}$

$$\underbrace{P_{k-1}^{r_{k-1}}P_{k-2}^{r_{k-2}}\cdots P_1^{r_1}B}_{\tilde{B}} = \sum_{i=1}^{k-1} l_i u_i^T + l_k u_k^T + \text{ other terms}$$

$$\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{i1} & l_{i2} & \cdots & 1 & 0 \\ l_{m1} & l_{m2} & \cdots & \cdots & 1 \end{bmatrix}\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1k} & u_{1m} \\ 0 & u_{22} & \cdots & u_{2k} & u_{2m} \\ 0 & \cdots & \cdots & u_{kk} & u_{km} \\ 0 & \cdots & \cdots & 0 & u_{mm} \end{bmatrix}$$

- **Step 1**: For $i = k, \ldots, m$

$$\tilde{a}_{ik} = \sum_{p=1}^{k} l_{ip}u_{pk} \Rightarrow l_{ik}u_{kk} = \tilde{a}_{ik} - \sum_{p=1}^{k-1} l_{ip}u_{pk}$$

$$\tilde{l}_{ik} = \tilde{a}_{ik} - \sum_{p=1}^{k-1} l_{ip}u_{pk}$$

$$\text{if } k = m, \text{ set } u_{mm} = \tilde{l}_{mm} \text{ and DONE. } IP(m) = m$$

LU decomposition procedure

- **Step 2**: Find relative $\max_i |\tilde{l}_{ik}|$, r_k = row($r_k \geq k$)
- **Step 3**: Swap row k and row r_k in lower right $(m - k + 1)$ subblock of $\tilde{B}$ and L. Columns $\underline{l}_1, \ldots, l_k$ are Δ_{lower} since $r_k \geq k$
- **Step 4**: For $i = k + 1, \ldots, m$

$$\text{if } \tilde{l}_{kk} \neq 0,\ \tilde{l}_{ik} = l_{ik} / \tilde{l}_{kk}$$
$$\text{if } \tilde{l}_{kk} = 0,\ \text{then OK since } l_{ik} = 0$$

- **Step 5**: Set $u_{kk} = \tilde{l}_{kk}$ and $u_{kj} = \tilde{a}_{kj} - \sum_{p=1}^{k-1} l_{kp} u_{pj}$; $j = k+1, \ldots, m$; k^{th} row of U
- **Step 6**: Set $k = k + 1$ and go to step 1

MATLAB command: [L,U,p]= lu(B,'vector')
p is a row vector containing permutation information

Comments

- Don't need 3 matrices. All work can be done in place:

$$l_{ik} \leftarrow a_{ik}; i = k+1,\ldots,m$$

$$u_{kj} \leftarrow a_{kj}; j = k,\ldots,m$$

when done, $$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ l_{21} & u_{22} & \cdots & u_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ l_{m1} & l_{m2} & \cdots & u_{mm} \end{bmatrix}$$

and vector IP that summarizes the permutation matrices $P_k^{r_k}, k = 1, 2, \ldots m$

- det(*PB*) = det(*P*)det(*B*). So,

$$\det(B) = (-1)^{\#\,Pivots} \prod_{i=1}^{m} u_{ii}$$

- $P_k^{r_k}s$ are symmetric and orthogonal so,

$$B = P_1^{r_1} P_2^{r_2} \cdots P_m^{r_m} LU$$

- $B^{-1} = U^{-1} L^{-1} P_m^{r_m} \cdots P_1^{r_1}$
- Number of operations

$$\sum_{k=1}^{m} 2(k-1)(m-k+1) = \sum_{i=1}^{m} i(m-i) = \frac{m^3}{2} - \frac{m^3}{6} = \frac{m^3}{3}$$

- Pivoting is essential. Otherwise, the method can be unstable
- Accumulate all inner products in DOUBLE PRECISION

Forward Elimination and Backward Substitution

- Remaining step: solution of $B\underline{x} = \underline{b}$

$$PB\underline{x} = P\underline{b} \Rightarrow P\underline{b} = P_m^{r_m} P_{m-1}^{r_{m-1}} \cdots P_1^{r_1} \underline{b}$$

$\Rightarrow$ swap $b_1 \leftrightarrow b_{r_1}$, etc … can do in place

$\Rightarrow LU\underline{x} = \underline{\tilde{b}}$

- Solve
  - $L\underline{y} = \underline{\tilde{b}}$; via FORWARD ELIMINATION and
  - $U\underline{x} = \underline{y}$; via BACKWARD SUBSTITUTION

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{m1} & l_{m2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \vdots \\ \tilde{b}_m \end{bmatrix} \Rightarrow \begin{cases} y_1 = \tilde{b}_1 \\ y_2 = \tilde{b}_2 - l_{21} y_1 \\ \vdots \\ y_k = \tilde{b}_k - \sum_{j=1}^{k-1} l_{kj} y_j \end{cases}$$

  - $O(m^2 - m)/2$ operations

Forward Elimination and Backward Substitution

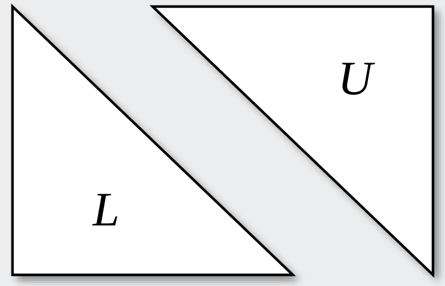
- can overwrite on $\tilde{b}_k$ with y_k

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ 0 & u_{22} & \cdots & u_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{mm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \Rightarrow \begin{cases} x_m = \dfrac{y_m}{u_{mm}} \\ x_{m-1} = \dfrac{y_{m-1} - u_{m-1,m} x_m}{u_{m-1,m-1}} \\ \vdots \\ x_k = \dfrac{\left(y_k - \sum_{i=m}^{m} l_{ki} x_i \right)}{u_{kk}} \end{cases}$$

- $O(m^2 + m/2)$ operations
- Total operations: $O(m^2) \Rightarrow$ Total = $O(m^3/3) + O(m^2)$
- But, in revised simplex, we drop a column and add a column at each iteration
  - Q: can we do it sequentially?
  - Q: can we do it faster? yes!!
- The sequential algorithm is due to Bartels and Golub, 1969 *CACM*, vol.12, no. 5, pp. 266-268

Sequential LU Update

$B =$



$\Rightarrow L^{-1}B = U$

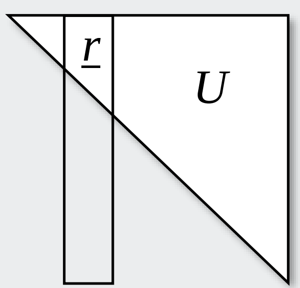
- Usually keep track of L^{-1} and U
- Recall

$$\begin{aligned}\overline{B} &= B + \text{col}\,\underline{a}_k - \text{col}\,\underline{a}_l,\ B = LU \\ &= B + \left(\underline{a}_k - B\underline{e}_l\right)\underline{e}_l^T\end{aligned}$$

  - So,

$$\begin{aligned}L^{-1}\overline{B} &= L^{-1}B + \left(L^{-1}\underline{a}_k - U\underline{e}_l\right)\underline{e}_l^T \\ &= U\left(I - \underline{e}_l\underline{e}_l^T\right) + \underline{r}\underline{e}_l^T,\ \underline{r} = L^{-1}\underline{a}_k\end{aligned}$$

- Pictorially, $\underline{r}$ replaces the lth column of U



$$\Rightarrow \begin{bmatrix} u_{11} & u_{12} & \cdots & r_1 & \cdots & u_{1m} \\ 0 & u_{22} & \cdots & r_2 & \cdots & u_{2m} \\ \vdots & 0 & \cdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & r_k & \cdots & u_{km} \\ 0 & \cdots & \cdots & r_m & \cdots & u_{mm} \end{bmatrix} = \tilde{U}$$

Key: Need to convert upper Hessenberg to upper triangular

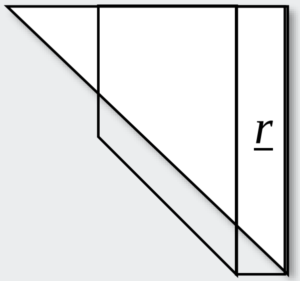
- Permute columns of $\tilde{U}$ so that
  - $\underline{r}$ is the last column $\Rightarrow \underline{r} \sim m$th column
  - Move other columns up by one place

    $l+1 \rightarrow 1$

    $m \rightarrow m-1$
  - The result is an **upper Hessenberg** matrix

$$H = \tilde{U}\left[\underline{e}_1 \underline{e}_2 \ldots \underline{e}_{l-1} \underline{e}_{l+1} \underline{e}_{l+2} \underline{e}_m \underline{e}_l\right] = \tilde{U}P$$

  - Don't physically have to permute, but can keep track of column positions … not worth the added complexity of coding
  - The result is:



Elementary Operation Matrix

- Now we want to reduce H to Δ_{upper}
- Use Gaussian elimination with pivoting
- Unwanted elements

  $u_{l+1,l+1}$ in column l

  u_{jj} in column $j-1$

  u_{mm} in column $m-1$

$\Rightarrow$ Note that these are original U elements

- Consider column l

$$\underline{u}_{l+1} = \begin{bmatrix} u_{1,l+1} \\ u_{2,l+1} \\ \vdots \\ u_{l,l+1} \\ u_{l+1,l+1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

And then we premultiply by

$$M_l \underline{u}_{l+1} = \begin{bmatrix} 1 &&&&&&& \\ & 1 &&&&&& \\ && 1 &&&&& \\ &&& 1 & 0 &&& \\ &&& m & 1 &&& \\ &&&&& 1 && \\ &&&&&& 1 & \\ &&&&&&& 1 \end{bmatrix} \begin{bmatrix} u_{1,l+1} \\ u_{2,l+1} \\ \vdots \\ u_{l,l+1} \\ u_{l+1,l+1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} u_{1,l+1} \\ u_{2,l+1} \\ \vdots \\ u_{l,l+1} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

where $m = \dfrac{-u_{l+1,l+1}}{u_{l,l+1}}$

Sequential LU Update

- When the subdiagonal element $u_{l+1,l+1} > u_{l,l+1}$ (the diagonal element of column l in H), then interchange rows l and $l+1 \Rightarrow$ interchange columns of M_l and $m \leftarrow \frac{1}{m}$

$$M_l \underline{u}_{l+1} = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 0 & 1 & & & \\ & & & 1 & m & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{bmatrix} \begin{bmatrix} u_{1,l+1} \\ u_{2,l+1} \\ \vdots \\ u_{l,l+1} \\ u_{l+1,l+1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} u_{1,l+1} \\ u_{2,l+1} \\ \vdots \\ u_{l,l+1} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

where $m = \frac{-u_{l,l+1}}{u_{l+1,l+1}}$

- Note:
  1. Must do the same for $\underline{b}$ vector
  2. Must consider effect of multiplying by M_l on all r columns l, $l+1,\ldots,m\text{-}1$

     Applying M $\overline{L}^{-1}\overline{B} = \overline{U}$, where $\overline{L}^{-1} = M_{m-1} \ldots M_l L^{-1}$

QR Decomposition

- QR decomposition methods
  - Householder
  - Gram-Schmidt, and
  - Givens orthogonalization methods

MATLAB command: [Q,R,p]= QR(B,'vector'). p is a row vector containing permutation information

- Key idea of all three methods
  - Find an $n \times n$ orthogonal matrix Q such that $Q^T B = R$ where R is $\triangle_{upper}$ $\Rightarrow Q = BR$
  - Do the same thing to $\underline{b} = Q^T \underline{b} = \tilde{b}$
  - Solve $\triangle_{upper}$ system of equations $R\underline{x} = \tilde{b}$
- <u>Givens transformations (rotations)</u>

$$\begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{bmatrix} \xrightarrow{(1,2)} \begin{bmatrix} x & x & x \\ 0 & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{bmatrix} \xrightarrow{(1,3)} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & x & x \\ x & x & x \\ x & x & x \end{bmatrix} \xrightarrow{(2,3)} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ x & x & x \\ x & x & x \end{bmatrix} \xrightarrow{(1,4)} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & x & x \\ x & x & x \end{bmatrix} \xrightarrow{(2,4)} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & x \\ x & x & x \end{bmatrix} \xrightarrow{(3,4)}$$

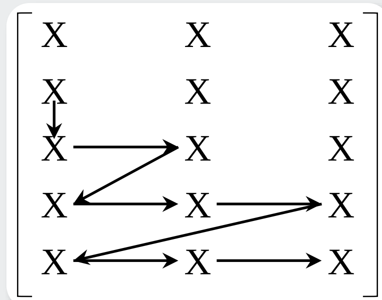
$$\begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & 0 \\ x & x & x \end{bmatrix} \xrightarrow{(1,5)} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & 0 \\ 0 & x & x \end{bmatrix} \xrightarrow{(2,5)} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & 0 \\ 0 & 0 & x \end{bmatrix} \xrightarrow{(3,5)} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Givens Transformations

$$J_{(3,5)}J_{(2,5)}J_{(1,5)}J_{(3,4)}J_{(2,4)}J_{(1,4)}J_{(2,3)}J_{(1,3)}J_{(1,2)}B = R$$

$$\Rightarrow Q^T = J_{(3,5)}J_{(2,5)}J_{(1,5)}J_{(3,4)}J_{(2,4)}J_{(1,4)}$$

- Zig-zag pattern of zeroed-out elements



- What are these Givens rotations?

$$J(i,k,\theta) = \begin{matrix} \\ \\ i \\ k \\ \\ \end{matrix}\begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & \cdots & c & s & 0 \\ 0 & \cdots & -s & c & 0 \\ 0 & \cdots & \cdots & \cdots & 1 \end{bmatrix} \begin{matrix} \\ c = \cos\theta \\ s = \sin\theta \\ \\ \\ \end{matrix}$$

$$\qquad\qquad i \quad k$$

$$= I + (\underline{v}_1 - \underline{e}_i)\underline{e}_i^T + (\underline{v}_2 - \underline{e}_k)\underline{e}_k^T$$

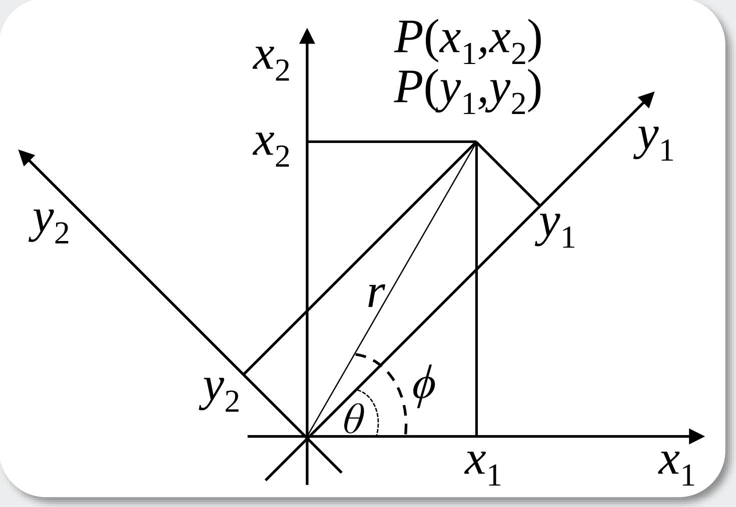
where $\underline{v}_1 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ c \\ \vdots \\ -s \\ \vdots \\ 0 \end{bmatrix}, \underline{v}_2 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ s \\ \vdots \\ c \\ \vdots \\ 0 \end{bmatrix},$ Note: $\underline{v}_1^T \underline{v}_2 = 0$

$\Rightarrow J(i,k,\theta)$ is a *rank two correction* to an identity matrix

Givens Transformations

- To motivate Givens rotations, consider the two-dimensional case:

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} cx_1 + sx_2 \\ -sx_1 + cx_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$



$$x_1 = r\cos\phi,\ x_2 = r\sin\phi$$
$$y_1 = r\cos(\phi-\theta) = r\cos\phi\cos\theta + r\sin\phi\sin\theta$$
$$y_2 = r\sin(\phi-\theta) = r\sin\phi\sin\theta - r\cos\phi\cos\theta$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}}_{J(1,2,\theta)\text{matrix}}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Rightarrow \begin{matrix}\text{rotation of } x_1 - x_2 \text{ axis} \\ \text{through an angle } \theta\end{matrix}$$

Also, $$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}}_{J(1,2,-\theta)\text{matrix}}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

- So we have the important result that

$$J^{-1}(1,2,\theta) = J^T(1,2,\theta) = J(1,2,-\theta)$$

Givens Transformation

- In general, $J(i,k,\theta)$ rotates $i-k$ coordinates by an angle θ in a counter-clockwise direction

$$J(i,k,\theta)\underline{x} = \underline{y}$$
$$\Rightarrow y_i = cx_i + sx_k, y_k = -sx_i + cx_k, y_j = x_j, \forall j \neq i,k$$

- Coming back to the general case, we can force $y_k \uparrow$ to 0 by letting

$$c = \frac{x_i}{\sqrt{x_i^2 + x_k^2}}, \; s = \frac{x_k}{\sqrt{x_i^2 + x_k^2}}$$

$\Rightarrow$ Any specified element can be *zeroed out* by approciate choice of c and s

$\Rightarrow$ Since the effect is local, the procedure is well-suited for parallel processing and ideal for revised simplex

- What is $J(i,k,\theta)B$ and $BJ(i,k,\theta)$

$J(i,k,\theta)B$ affects only rows i and k of B

$BJ(i,k,\theta)$ affects only columns i and k of B

} local effect

Givens Transformations

$\underline{J(i,k,\theta)B}$	$\underline{BJ(i,k,\theta)}$

For $j = 1, \ldots, m$ Do
$v = a_{ij}$
$\omega = a_{kj}$
$a_{ij} = cv + s\omega$
$a_{kj} = -sv + c\omega$
End Do
	For $l = 1, \ldots, m$ Do
	$v = a_{li}$
	$\omega = a_{lk}$
	$a_{li} = cv + s\omega$
	$a_{lk} = -sv + c\omega$
	End Do

Note: $J(i,k,\theta)B$ requires $O(2m)$ operations; $BJ(i,k,\theta)$ requires $O(2m)$ operations

- **Givens Orthogonalization Procedure**

For $k = 1, \ldots, m$ DO
  For $i = 1, \ldots, k-1$ DO
    Find c and s such that

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a_{ii} \\ a_{ki} \end{bmatrix} = \begin{bmatrix} x \\ 0 \end{bmatrix}$$

    $B \leftarrow J(i,k,\theta)B$
  End DO
End DO

- Number of operations: $\frac{4}{3}m^3$
- If want to solve $B\underline{x} = \underline{b}$, insert $\underline{b} \leftarrow J(i,k,\theta)B$ and solve $R\underline{x} = \underline{b}$

Sequential QR Updates

- Recall

$$\bar{B} = B + \text{col}\, \underline{a}_k - \text{col}\, \underline{a}_l,\ B = QR$$
$$= B + \left(\underline{a}_k - B\underline{e}_l\right)\underline{e}_l^T$$

  - So,

$$Q^T \bar{B} = Q^T B + \left(Q^T \underline{a}_k - R\underline{e}_l\right)\underline{e}_l^T$$
$$= R\left(I - \underline{e}_l \underline{e}_l^T\right) + \underline{r}\underline{e}_l^T,\ \underline{r} = Q^T \underline{a}_k$$

- As before, we move $\underline{r}$ to column m and move other columns to the left by one ($m \rightarrow (m-1)$, $(m-1) \rightarrow (m-2)$, etc.)
- The result is an upper Hessenberg matrix in columns $l, l+1, \ldots, m$
- Zero out unwanted subdiagonals $h_{l+1,l}, \ldots, h_{m,m-1}$

$$\Rightarrow J_{m-1}^T \ldots J_{l+1}^T J_l^T H = R_1$$
$$\Rightarrow Q = Q J_l J_{l+1} \ldots J_{m-1}$$

- Computational load: $O(m^2)$
- Do the same thing to rhs $\Rightarrow \underline{b}$ and $\underline{c}_b$

MATLAB command: [Q1,R1]= qrupdate(Q,R,u,v)
u and v are column vectors corresponding to rank one update

Storage Schemes

- Store matrix A column by column
- Record only nonzero entries a_{ij} and the corresponding row index i
- Usually < 5-10 entries per column ⇒ Density $\leq$ 5%
- Typical storage scheme for column j
  - # of nonzero row elements
  - Elements: a_{1j} a_{5j} a_{9j} a_{200j}
  - Locations: 1 5 9 200

MATLAB command: [L,U,p,q]= lu(B)
p is a row vector containing row permutation information
q is a column vector containing column permutation information
Performs the decomposition: PBQ=LU

LP with Upper bound Constraints on Variables

- Variables with upper bounds

$$\min \underline{c}^T \underline{x}$$
$$\text{s.t. } A\underline{x} = \underline{b}$$
$$\underline{0} \le \underline{x} \le \underline{h}$$

- Can convert to SLP: $(m + n) \times 2n$ matrix A
- Q: can we solve it without converting to SLP? Yes!!
- Solution is always an extended bfs
  - $(n - m)$ variables are at their lower bound (zero) or at their upper bound (h_i)
- Suppose we start with an extended bfs
  - A non-basic variable at its lower bound can only be increased

  $\Rightarrow p_j < 0$ to decrease cost
  - A non-basic variable at its upper bound can only be decreased

  $\Rightarrow p_j > 0$ to decrease cost

LP with Upper bound Constraints on Variables

- As x_{Nk} changes, one of two things happen:
  - A basic variable goes to one of its bounds ....(1)
  - The non-basic variable goes to its opposite bound ...(2)
- If (1) occurs, ok...change the basis
- If (2) occurs, basis does not change
- Optimality $\Rightarrow p_j \geq 0$ if $x_j = 0$

  $p_j \leq 0$ if $x_j = h_j$
- Modifications to revised simplex algorithm
  - Change steps 4 and 5 as follows:
    - ❖ Pick nonbasic variable x_{Nk} and compute $B^{-1}\underline{a}_k$
    - ❖ **Step 4:** Evaluate three numbers (called "bottlenecks")

      1. h_k 2. $\min_{i:\alpha_{ik}>0} \frac{\beta_i}{\alpha_{ik}}$ 3. $\min_{i:\alpha_{ik}<0} \frac{(\beta_i - h_i)}{\alpha_{ik}}$
    - ❖ **Step 5:**

1. If h_k is the smallest, $x_{Nk} \rightarrow$ opposite bound and the basis does not change. Replace x_{Nk} *by* (h_k- x'_{Nk}) throughout. Basically, $p_j \rightarrow -p_j$
2. Suppose l is the minimizing index in *2*. Then the lth basic variable returns to its lower bound ($\Rightarrow$ becomes non-basic)
3. If l is the minimizing index in *3*, then the lth basic variable goes to its opposite (upper) bound ($\Rightarrow$ becomes non-basic)

Example with Upper bound Constraints

$$\min\ f = -4x_1 - 2x_2 - 3x_3$$

$$\text{s.t}\quad 2x_1 + x_2 + x_3 \le 10$$

$$x_1 + 0.5x_2 + 0.5x_3 \le 6$$

$$2x_1 + 2x_2 + 4x_3 \le 20$$

$$0 \le x_1 \le 4$$

$$0 \le x_2 \le 3$$

$$0 \le x_3 \le 1$$

Iteration 1: slack variables as basis

$$\underline{x}_B = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 6 \\ 20 \end{bmatrix}; B = I = B^{-1};$$

$$\underline{\lambda}^T = [0\ \ 0\ \ 0]B^{-1} = [0\ \ 0\ \ 0]$$

$$\underline{p}^T = [-4\ -2\ -3\ 0\ 0\ 0]$$

Evaluate the three bottlenecks for x_1:

$(1)\, h_1 = 4; (2)\theta_2 = \min(5,6,10) = 5, (3)\theta_3 = \infty$

So, $x_1 \to h_1 = 4$; basis does not change!

Replace x_1 *by* $(4 - x_1^{'})$

$\Rightarrow$ change $\underline{b} \to \underline{b} - (B)_1 \underline{h}_1 = [2\ \ 2\ \ 12]^T$

(new slacks!) and negate column 1 of B

Iteration 2:

$$\underline{x}_B = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 12 \end{bmatrix}; B = I = B^{-1};$$

$$\underline{\lambda}^T = [0\ \ 0\ \ 0]B^{-1} = [0\ \ 0\ \ 0]$$

$\underline{p}^T = [4\ -2\ -3\ 0\ 0\ 0]$...*recall* $x_1 \to 4 - x_1^{'}$

Evaluate the three bottlenecks for x_3:

$(1)\, h_3 = 1; (2)\theta_2 = \min(2,4,3) = 2, (3)\theta_3 = \infty$

So, $x_3 \to h_3 = 1$; basis does not change!

Replace x_3 *by* $1 - x_3^{'}$

$\Rightarrow$ change $\underline{b} \to \underline{b} - (B)_3 \underline{h}_3 = \begin{bmatrix} 1 & \frac{3}{2} & 8 \end{bmatrix}^T$

(new slacks) and negate column 3 of B

Iteration 3:

$$\underline{x}_B = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 3/2 \\ 8 \end{bmatrix}; B = I = B^{-1};$$

$$\underline{\lambda}^T = [0\ \ 0\ \ 0]B^{-1} = [0\ \ 0\ \ 0]$$

$$\underline{p}^T = [4\ -2\ 3\ 0\ 0\ 0]$$

Evaluate the three bottlenecks for x_2:

$(1)\, h_2 = 3; (2)\theta_2 = \min(1,3,4) = 1, (3)\theta_3 = \infty$

So, $x_2 \to 1$; bring x_2 into basis; s_1 goes out.

Iteration 3: (continued)

$$\underline{x}_B = \begin{bmatrix} x_2 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 6 \end{bmatrix}; B = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix}; B^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}$$

$$\underline{\lambda}^T = [-2\ \ 0\ \ 0]B^{-1} = [-2\ \ 0\ \ 0]$$

$$\underline{p}^T = [8\ 0\ 5\ 2\ 0\ 0] \Rightarrow optimal$$

So, $x_1^{'} = 0 \Rightarrow x_1 = 4; x_2 = 1; x_3^{'} = 0 \Rightarrow x_3 = 1;$

$s_1 = 0; s_2 = 1; s_3 = 6; f^* = -21$

Decomposition Methods

- Large-scale LP problems are typically separable and/or have very few dependencies among (sets of) variables
- Consider

$$\min \underline{c}^T \underline{x} \quad \text{s.t. } A\underline{x} = \underline{b}, \quad \underline{x} \geq \underline{0}$$

- Suppose *A* has special "**block-angular**" structure

$$A = \begin{bmatrix} L_1 & L_2 & \cdots & L_r \\ A_1 & & \cdots & \\ & A_2 & \cdots & \\ & \cdots & & A_r \end{bmatrix}$$

- Define

$$m = \sum_{i=0}^{r} m_i, n = \sum_{i=1}^{r} n_i$$

Decomposition Methods

- Partition $\underline{c}$, $\underline{x}$, $\underline{b}$ conformally as:

$$\underline{x}^T = \left(\underline{x}_1^T, \underline{x}_2^T, \ldots, \underline{x}_r^T\right),\ \underline{b}^T = \left(\underline{b}_1^T, \underline{b}_2^T, \ldots, \underline{b}_r^T\right)$$

$$\underline{c}^T = \left(\underline{c}_1^T, \underline{c}_2^T, \ldots, \underline{c}_r^T\right)$$

  - Then, the problem is:

$$\begin{aligned} \min & \sum\nolimits_{i=1}^{r} \underline{c}_i^T \underline{x}_i \\ \text{s.t.} & \sum\nolimits_{i=1}^{r} L_i \underline{x}_i = \underline{b}_0 \\ & A_i \underline{x}_i = \underline{b}_i, i = 1, 2, \ldots, r \\ & \underline{x}_i \geq \underline{0} \end{aligned}$$

- r separable LPs linked by m_0 constraints
- Physical interpretation
  - Minimize total cost of operation of an r division firm
  - Division activities are constrained by $A_i\underline{x}_i = \underline{b}_i$
  - Overall resource constraint $\underline{b}_0$
- Suppose we use regular LP
  - Let $r = 100$, $m_i = 100$, $0 \leq i \leq r$
  - Need a basis matrix of $\approx 10^4 \times 10^4 \Rightarrow$ storage/computational problems

Dantzig-Wolfe Decomposition (DWD)

- To illustrate the idea of decomposition, consider r = 2 case

$$\begin{aligned}\min_{\underline{x}_1 \ge \underline{0}, \underline{x}_2 \ge \underline{0}} \ & \underline{c}_1^T \underline{x}_1 + \underline{c}_2^T \underline{x}_2 \\ \text{s.t.} \quad & L_1 \underline{x}_1 + L_2 \underline{x}_2 = \underline{b}_0 \\ & A_1 \underline{x}_1 = \underline{b}_1,\ A_2 \underline{x}_2 = \underline{b}_2\end{aligned}$$

- Consider subproblem 1:

$$\begin{aligned}\min_{\underline{x}_1} \ & \underline{c}_1^T \underline{x}_1 \\ \text{s.t.} \quad & A_1 \underline{x}_1 = \underline{b}_1 \\ & \underline{x}_1 \ge 0\end{aligned}$$

- Recall that the feasible set is a convex polytope and if bounded it is a convex polyhedron
- Then $\underline{x}_1 = \sum_{j=1}^{k_1} \delta_j \underline{y}_j$
  - $\{\underline{y}_j\}$ are the extreme points of the polyhedral set
  - For an unbounded convex polytope, add a nonnegative combination of extreme directions
  - $\delta_j \ge 0$, $\underline{e}^T \underline{\delta} = 1$
  - But, don't know δ_j!!!

DWD: Master LP

- Similarly, $\underline{x}_2 = \sum_{j=1}^{k_2} \mu_j \underline{z}_j$
  - $\{\underline{z}_j\}$ are the extreme points; $\mu_j \geq 0$, $\underline{e}^T\underline{\mu} = 1$
  - But, don't know μ_j!!!
- Original problem in terms of extreme points is called <u>Master LP</u>

$$\begin{aligned}
&\min_{\underline{x}_1 \underline{x}_2} \underline{c}_1^T \underline{x}_1 + \underline{c}_2^T \underline{x}_2 = \sum_{j=1}^{k_1} (\delta_j)(\underline{c}_1^T \underline{y}_j) + \sum_{j=1}^{k_2} (\mu_j)(\underline{c}_2^T \underline{z}_j) \\
&\text{s.t.} \quad \mu_j \geq 0, \delta_j \geq 0, \underline{e}^T \underline{\delta} = \underline{e}^T \underline{\mu} = 1 \\
&\qquad \sum_{j=1}^{k_1} \delta_j L_1 \underline{y}_j + \sum_{j=1}^{k_2} \mu_j L_2 \underline{z}_j = \underline{b}_0
\end{aligned}$$

- # of variables $k_1 n_1 + k_2 n_2 + k_1 + k_2$
- # of constraints: m_0+2 … for an r-division firm: m_0+r
- Need a basis matrix of dimension $(m_0 + 2) \times (m_0 + 2)$ only
- Thus, let $\underline{\alpha}^T = (\delta_1 \ldots \delta_{k_1}, \mu_1 \ldots \mu_{k_2})$

$$\underline{s}^T = (\underline{c}_1^T \underline{y}_1, \underline{c}_1^T \underline{y}_2, \ldots, \underline{c}_1^T \underline{y}_{k_1}, \underline{c}_2^T \underline{z}_1, \underline{c}_2^T \underline{z}_2, \ldots, \underline{c}_2^T \underline{z}_{k_2}), \quad \underline{q}_j = \begin{bmatrix} L_1 \underline{y}_j \\ 1 \\ 0 \end{bmatrix}, \quad \underline{q}_{k_1+j} = \begin{bmatrix} L_2 \underline{y}_j \\ 0 \\ 1 \end{bmatrix}$$

Solving Master LP

- In terms of $\underline{\alpha}$, the master LP is:

$$\min \underline{s}^T \underline{\alpha}$$
$$\text{s.t. } Q\underline{\alpha} = \begin{bmatrix} \underline{b}_0 \\ 1 \\ 1 \end{bmatrix}$$
$$\underline{\alpha} \geq 0$$

- Suppose we had a basis B and

$$\underline{\lambda}^T = \underline{s}_B^T B^{-1} = \left(\underline{\lambda}_0^T \lambda_{m_0+1} \lambda_{m_0+2} \right)$$

- Relative (reduced) cost vector $\underline{p} = \left(\underline{s}_N^T - \underline{\lambda}^T N \right)$

$$\Rightarrow p_j = s_j - \underline{\lambda}_0^T L_1 \underline{y}_j - \lambda_{m_0+1}; 1 \leq j \leq k_1$$
$$p_j = s_j - \underline{\lambda}_0^T L_2 \underline{z}_j - \lambda_{m_0+2}; k_1 + 1 \leq j \leq k_1 + k_2$$

- Want to find the minimum relative cost coefficient p_k
- For Bland's rule, all we need is a first negative reduced cost

⇒ Fortunately don't have to evaluate at each extreme point

Decomposition Framework

- We know $p^* = \min_{1 \le j \le k_1 + k_2} (p_j) = \min \left\{ \min_{1 \le j \le k_1} (p_j), \min_{k_1 + 1 \le j \le k_1 + k_2} (p_j) \right\} = \min\{p_1^*, p_2^*\}$
- Consider the 1st term

$$\min_{1 \le j \le k_1} (p_j) = \min_{1 \le j \le k_1} \left\{ s_j - \underline{\lambda}_0^T L_1 \underline{y}_j - \lambda_{m_0+1} \right\}$$

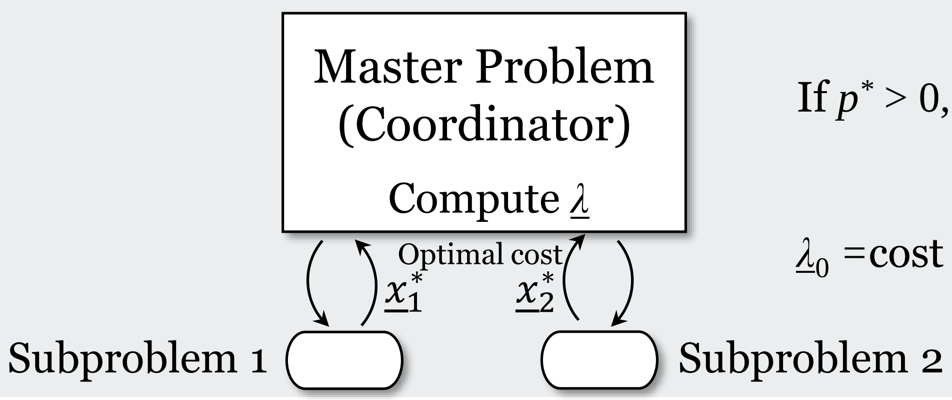
- But, this is equivalent to an LP

$$\begin{aligned} \min_{\underline{x}_1} & \left(\underline{c}_1^T - \underline{\lambda}_0^T L_1\right) \underline{x}_1 \\ \text{s.t.} \quad & A_1 \underline{x}_1 = \underline{b}_1 \\ & \underline{x}_1 \ge 0 \end{aligned}$$

and similarly,

$$\begin{aligned} \min_{\underline{x}_2} & \left(\underline{c}_2^T - \underline{\lambda}_0^T L_2\right) \underline{x}_2 \\ \text{s.t.} \quad & A_2 \underline{x}_2 = \underline{b}_2 \\ & \underline{x}_2 \ge 0 \end{aligned}$$

$\Rightarrow$ Finding reduced costs is equivalent to solving two LPs $\Rightarrow$ distributed problem solving



If $p^* > 0$, Stop

$\underline{\lambda}_0$ =cost of common resources

Steps of DWD Algorithm

Step 1: Calculate the current basic solution $\underline{x}_B$ and solve

$$\underline{\lambda}^T B = \underline{s}_B^T \text{ where } \underline{\lambda}^T = \left(\underline{\lambda}_0, \lambda_{m_0+1}, \lambda_{m_0+2}\right)$$

Step 2: Solve decoupled LPs

$$\begin{aligned} \min_{\underline{x}_i} & \left(\underline{c}_i^T - \underline{\lambda}_0^T L_i\right)\underline{x}_i = z_i \\ \text{s.t.} \quad & A_i \underline{x}_i = \underline{b}_i \\ & \underline{x}_i \geq 0 \end{aligned}$$

Step 3: Compute $p_i = z_i - \lambda_{m_0+i}$

Step 4: If all $p_i > 0 \Rightarrow$ optimal

Step 5: Else find minimal $p_i^* = p^*$

Suppose this corresponds to subproblem k

Compute $[L_k \underline{x}_k^* \quad 0 \quad \cdots \quad 1 \quad \cdots \quad 0]^T$

Step 6: Enter column into basis as in revised simplex

Q: do $\{\lambda_i\}$ have any meaning?
… Lagrangian dual & lecture 4

A Preview to Duality & DWD

$\text{Primal}:$

$$\min_{\{\underline{x}_i \geq \underline{0}\}} f = \sum_{i=1}^{r} \underline{c}_i^T \underline{x}_i$$

$$\text{s.t.} \quad \sum_{i=1}^{r} L_i \underline{x}_i = \underline{b}_0$$

$$A_i \underline{x}_i = \underline{b}_i, i = 1, 2, .., r$$

Master LP

$$\min_{\{\delta_{ji}\}} \sum_{i=1}^{r} \sum_{j=1}^{k_i} (\delta_{ji})(\underline{c}_i^T \underline{y}_{ji})$$

$$\text{s.t.} \quad \sum_{i=1}^{r} \sum_{j=1}^{k_i} \delta_{ji} L_i \underline{y}_{ji} = \underline{b}_0$$

$$\sum_{j=1}^{k_i} \delta_{ji} = 1; i = 1, 2, .., r$$

$$\delta_{ji} \geq 0, \ j = 1, 2, .., k_i; i = 1, 2, .., r$$

Lagrangian Dual of Master LP:

$$\max_{\underline{\lambda}_0, \{\lambda_{m_0+i}\}_{i=1}^r} q(\underline{\lambda}_0, \{\lambda_{m_0+i}\}_{i=1}^r) = \underline{\lambda}_0^T \underline{b}_0 + \sum_{i=1}^{r} \lambda_{m_0+i}$$

$$\text{s.t.} \quad \underline{\lambda}_0^T L_i \underline{y}_{ji} + \lambda_{m_0+i} \leq \underline{c}_i^T \underline{y}_{ji}; \ j = 1, 2, .., k_i; i = 1, 2, .., r$$

$$\text{If } \ p_i = \min_j \{p_{ij} = (\underline{c}_i^T - \underline{\lambda}_0^T L_i) \underline{y}_{ji} - \lambda_{m_0+i}; \ j = 1, 2, .., k_i\} > 0 \ \forall i = 1, 2, .., r \Rightarrow \textit{optimal}$$

$$\text{If } \ k = \arg\min_i p_i, \ \text{bring } [L_k \underline{y}_{j^*(k)k} \ldots\ldots\ldots 1 \ldots\ldots\ldots]^T \text{ into basis and do simplex step.}$$

column generation

Dual (lower) bound

$$p_i \leq (\underline{c}_i^T - \underline{\lambda}_0^T L_i) \underline{y}_{ji} - \lambda_{m_0+i} = z_i - \lambda_{m_0+i} \ \forall j = 1, 2, .., k_i$$

$$\Rightarrow \underline{\lambda}_0^T L_i \underline{y}_{j^*i} + \lambda_{m_0+i} + p_i \leq \underline{c}_i^T \underline{y}_{j^*(i)i} \Rightarrow \underline{\lambda}_0^T \underline{b}_0 + \sum_{i=1}^{r} \left[\lambda_{m_0+i} + p_i \right] = \underline{\lambda}_0^T \underline{b}_0 + \sum_{i=1}^{r} z_i \leq f^*$$

Dantzig-Wolfe Decomposition Example 1

$$\begin{aligned} \min\ & f = -x_1 - 2x_2 - 4x_3 - 3x_4 \\ \text{s.t.}\ & x_1 + x_2 + 2x_3 \le 4 \\ & x_2 + x_3 + x_4 \le 3 \\ & 2x_1 + x_2 \le 4 \\ & x_1 + x_2 \le 2 \\ & x_3 + x_4 \le 2 \\ & 3x_3 + 2x_4 \le 5 \end{aligned}$$

- Define

$$\underline{x}_1 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix};\ \underline{x}_2 = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$$

$$\underline{c}_1^T = [-1, -2];\ \underline{c}_2^T = [-4, -3]$$

$$L_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix};\ L_2 = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}$$

- Recall

$$\underline{x}_1 = \sum_{j=1}^{k_1} \delta_j \underline{y}_j \qquad \underline{x}_2 = \sum_{j=1}^{k_2} \mu_j \underline{z}_j$$

- So, master problem is

$$\begin{aligned} \min\ & \sum_{j=1}^{k_1} \delta_j \underline{c}_1^T \underline{y}_j + \sum_{j=1}^{k_2} \mu_j \underline{c}_2^T \underline{z}_j \\ \text{s.t.}\ & \sum_{j=1}^{k_1} \delta_j L_1 \underline{y}_j + \sum_{j=1}^{k_2} \mu_j L_2 \underline{z}_j = \underline{b}_0 \\ & \sum \delta_j = \sum \mu_j = 1 \end{aligned}$$

$\Rightarrow$ # of constraints = 2 coupling constraints + 2 convexity constraints = 4 $\Rightarrow$ basis matrix is 4×4

- Introduce slacks d_1 and d_2 in master problem

Dantzig-Wolfe Decomposition Example 1

- Iteration 0: Getting started

$$\underline{d} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}, B = I$$

$$\underline{x}_1 = 0,\ \underline{x}_2 = 0;\ \delta_1 = 1, \mu_1 = 1$$

$$\underline{\beta}^{(0)} = [4 \quad 3 \quad 1 \quad 1]^T$$

$$\underline{\lambda}^{(0)T} = [0 \quad 0 \quad 0 \quad 0] B^{-1} = \underline{0}$$

$$\underline{x}_B = \begin{bmatrix} d_1 \\ d_2 \\ \delta_1 \\ \mu_1 \end{bmatrix}; \underline{c}_B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- Iteration 1:
  - Subproblem solutions

$$\begin{aligned} &\min -x_1 - 2x_2 \\ &\text{s.t. } 2x_1 + x_2 \le 4 \\ &\quad\quad x_1 + x_2 \le 2 \end{aligned}$$

$$\begin{aligned} &\min\ -4x_3 - 3x_4 \\ &\text{s.t. } x_3 + x_4 \le 2 \\ &\quad\quad 3x_3 + 2x_4 \le 5 \end{aligned}$$

  - Optimal solutions of subproblems

$$\underline{x}_1^{(1)} = [0 \quad 2];\ \underline{x}_2^{(1)} = [1 \quad 1];\ z_1 = -4;\ z_2 = -7$$

$$-11 \le f^* \le 0$$

  - Relative cost coefficients

$$p_1^* = -4 + 0 = -4,\ p_2^* = -7 + 0 = -7$$

Dantzig-Wolfe Decomposition Example 1

- Master Iteration

  ⇒ Need to bring in solution corresponding to μ_2 into basis ⇒ column to enter basis is

$$\begin{bmatrix} L_2 \underline{x}_2^{(1)} \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \begin{pmatrix} 2 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 0 \\ 1 \end{bmatrix} = \underline{a}_k$$

$$B^{-1}\underline{a}_k = \begin{bmatrix} 2 \\ 2 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_{1k} \\ \alpha_{2k} \\ 0 \\ \alpha_{4k} \end{bmatrix} \Rightarrow \frac{\beta_i}{\alpha_{ik}} = \begin{bmatrix} 2 \\ 1.5 \\ x \\ 1 \end{bmatrix}$$

<u>Note:</u> remember division by non-zero α_{ik} only and minimum $\frac{\beta_i}{\alpha_{ik}}$ is the one that leaves basis

Dantzig-Wolfe Decomposition Example 1

⇒ Column 4 must go

$$\text{New } \underline{x}_B = \begin{bmatrix} d_1 \\ d_2 \\ \delta_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \underline{\beta}^{(1)}$$

<u>Note:</u> remember μ_2 (element 4) is replaced by $\mu_2 = 1$

$$B^{-1} = \begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

As a check, note that $\underline{\beta}^{(1)} = B^{-1}\underline{b} = [2 \;\; 1 \;\; 1 \;\; 1]^T$

$$\underline{\lambda}^{(1)T} = [0 \quad 0 \quad 0 \quad -7] B^{-1} = [0 \quad 0 \quad 0 \quad -7]$$

Dantzig-Wolfe Decomposition Example 1

- Iteration 2
  - $\underline{\lambda}_0^{(1)} = \underline{\lambda}_0^{(0)} \Rightarrow$ subproblem solutions do not change, but $p_1^* = -4$, $p_2^* = 0$
- Master Iteration

$-11 \leq f^* \leq -7$

$\Rightarrow$ Bring in column corresponding to subproblem 1

$$\begin{bmatrix} L_1 \underline{x}_1^{(2)} \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$

Q: Which one should go?

Remember $\underline{\beta}^{(1)} = B^{-1}\underline{b} = [2 \;\; 1 \;\; 1 \;\; 1]^T$

$$B^{-1}\underline{a}_k = \begin{bmatrix} 2 \\ 2 \\ 1 \\ 1 \end{bmatrix} \Rightarrow \frac{\beta_i}{\alpha_{ik}} = \begin{bmatrix} 2 \\ \frac{1}{2} \\ 1 \\ x \end{bmatrix}$$

Dantzig-Wolfe Decomposition Example 1

A: Column 2 must go!

$$\text{new } B^{-1} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & -1 \\ 0 & -\frac{1}{2} & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow \underline{\beta}^{(2)} = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \frac{1}{2}\begin{bmatrix} 2 \\ 2 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{2} \\ 0 \\ 0 \end{bmatrix} = B^{-1}\underline{b} = \begin{bmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{2} \\ 1 \end{bmatrix}$$

$$\Rightarrow \underline{\lambda}^{(2)T} = \begin{bmatrix} 0 & -4 & 0 & -7 \end{bmatrix} B^{-1} = \begin{bmatrix} 0 & -2 & 0 & -3 \end{bmatrix}$$

- Iteration 3:
  - Subproblems

$$\begin{aligned} \min & -x_1 \\ \text{s.t. } & 2x_1 + x_2 \le 4 \\ & x_1 + x_2 \le 2 \\ & x_1, x_2 \ge 0 \end{aligned} \qquad \begin{aligned} \min & -2x_3 - x_4 \\ \text{s.t. } & x_3 + x_4 \le 2 \\ & 3x_3 + 2x_4 \le 5 \\ & x_3, x_4 \ge 0 \end{aligned}$$

$$\Rightarrow \underline{x}_1^{(3)} = [2,0], \underline{x}_2^{(3)} = [1,1];\ p_1^* = -2,\ p_2^* = 0$$

⇒ Column to enter: solution corresponding to subproblem 1 at iteration 3

$$-6-2-3=-11 \le f^* \le -8.5$$

Dantzig-Wolfe Decomposition Example 1

$$\underline{a}_k = \begin{bmatrix} L_1 \underline{x}_1^{(3)} \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \end{bmatrix} \Rightarrow B^{-1}\underline{a}_k = \begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$\Rightarrow$ To determine column to go, recall:

$\underline{\beta}^{(2)} = [1 \; \frac{1}{2} \; \frac{1}{2} \; 1]$, so

$$\frac{\beta_i}{\alpha_{ik}} = \begin{bmatrix} \frac{1}{2} \\ x \\ \frac{1}{2} \\ x \end{bmatrix}$$

- Pick column 3 to go

$$\text{new } B^{-1} = \begin{bmatrix} 1 & 0 & -2 & -2 \\ 0 & \frac{1}{2} & 0 & -1 \\ 0 & -\frac{1}{2} & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} d_1 \\ \delta_2 \\ \delta_3 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{2} \\ 1 \end{bmatrix} - \frac{1}{2}\begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{2} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{2} \\ \frac{1}{2} \\ 1 \end{bmatrix} = \underline{\beta}^{(3)}$$

$$\underline{\lambda}^{(3)T} = \begin{bmatrix} 0 & -4 & -2 & -7 \end{bmatrix} B^{-1} = \begin{bmatrix} 0 & -1 & -2 & -5 \end{bmatrix}$$

Dantzig-Wolfe Decomposition Example 1

- Iteration 4
  - Subproblems

$$\begin{aligned} \min\ & -x_1 - x_2 \\ \text{s.t.}\ & 2x_1 + x_2 \le 4 \\ & x_1 + x_2 \le 2 \\ & x_1, x_2 \ge 0 \end{aligned} \qquad \begin{aligned} \min\ & -3x_3 - 2x_4 \\ \text{s.t.}\ & x_3 + x_4 \le 2 \\ & 3x_3 + 2x_4 \le 5 \\ & x_3, x_4 \ge 0 \end{aligned}$$

-3-2-5=-10≤$f^* \le$ -10

$$\Rightarrow \underline{x}_1^{(4)} = [1,1] \text{ or } [2,0], \underline{x}_2^{(4)} = [1,1]$$

$\Rightarrow$ Optimal value = 0 for each subproblem

$\Rightarrow$ Reduced costs: $p_1^* = 2$, $p_2^* = 5$ $\Rightarrow$ found an optimal solution for master problem

- Solution:

$$\frac{1}{2}\underline{x}_1^{(2)} + \frac{1}{2}\underline{x}_1^{(3)} + \underline{x}_2^{(2)} = \frac{1}{2}\begin{bmatrix}0\\2\\0\\0\end{bmatrix} + \frac{1}{2}\begin{bmatrix}2\\0\\0\\0\end{bmatrix} + \begin{bmatrix}0\\0\\1\\1\end{bmatrix} = \begin{bmatrix}1\\1\\1\\1\end{bmatrix}$$

optimal solution

Optimal cost = -10

Dantzig-Wolfe Decomposition Example 2

$$
\begin{aligned}
\min\ f &= -90x_1 - 80x_2 - 70x_3 - 60x_4 \\
\text{s.t}\quad & 3x_1 + x_2 \le 12 \\
& 2x_1 + x_2 \le 10 \\
& 3x_3 + 2x_4 \le 15 \\
& x_3 + x_4 \le 4 \\
& 8x_1 + 6x_2 + 7x_3 + 5x_4 \le 80 \\
& L_1 : (8 \quad 6),\ L_2 : (7 \quad 5)
\end{aligned}
$$

- Introduce slack s_1 into the master problem

- Iteration 0:

$$bfs = (80 \quad 1 \quad 1)$$

$$\delta_1 \to \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\mu_1 \to \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\lambda^{(0)^T} = c_B^T B^{-1} = (0 \ \ 0 \ \ 0) B^{-1} = 0$$

Dantzig-Wolfe Decomposition Example 2

- Subproblem 1:

$$\{[-90\ \ -80]-0[8\ \ 6]\}\begin{pmatrix}x_1\\x_2\end{pmatrix}$$

$$\text{s.t.}\ \begin{bmatrix}3 & 1\\2 & 1\end{bmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix}\le\begin{pmatrix}12\\10\end{pmatrix}\Rightarrow -90x_1-80x_2$$

$$\text{s.t.}\ 3x_1+x_2\le 12$$
$$2x_1+x_2\le 10$$

Opt. solution: $(x_1, x_2) = (0, 10)$

Opt. cost z_1 : –800

- Subproblem 2:

$$\{[-70\ \ -60]-0[7\ \ 5]\}\begin{pmatrix}x_3\\x_4\end{pmatrix}$$

$$\text{s.t.}\ 3x_3+2x_4\le 15$$
$$x_3+\ \ x_4\le 4$$
$$\Rightarrow \min -70x_3-60x_4$$
$$\text{s.t.}\ 3x_3+2x_4\le 15$$
$$x_3+\ \ x_4\le 4$$

Opt. solution: $(x_3, x_4) = (4, 0)$

Opt. cost z_2 : –280

$$p_1^* = -800, \quad p_2^* = -280$$

$$0\text{-}800\text{-}280=-1080 \le f^* \le 0$$

Dantzig-Wolfe Decomposition Example 2

$\Rightarrow$ Bring $\begin{pmatrix} L_1 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ 1 \\ 0 \end{pmatrix}$ into the basis

$$\Rightarrow \begin{pmatrix} \begin{pmatrix} 8 & 6 \end{pmatrix} \begin{pmatrix} 0 \\ 10 \end{pmatrix} \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 60 \\ 1 \\ 0 \end{pmatrix} = a_k$$

$$B^{-1}a_k = \begin{pmatrix} 60 \\ 1 \\ 0 \end{pmatrix}, bfs = \begin{pmatrix} 80 \\ 1 \\ 1 \end{pmatrix}$$

$$\Rightarrow \theta = 1$$

$\Rightarrow$ Remove column 2 (δ_1) from the basis

- Iteration 1

$$\text{New } bfs = \begin{pmatrix} 80-60 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 20 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} s_1 \\ \delta_2 \\ \mu_1 \end{pmatrix} \ldots -800$$

$$\text{New } B^{-1} = EB^{-1} = \begin{bmatrix} 1 & -60 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\lambda^{(1)^T} = c_B^T B^{-1} = \begin{pmatrix} 0 & -800 & 0 \end{pmatrix} \begin{pmatrix} 1 & -60 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -800 & 0 \end{pmatrix}$$

0-800-280=-1080 $\leq f^* \leq$ -800

Dantzig-Wolfe Decomposition Example 2

- Subproblem 1:

$$\begin{bmatrix}-90 & -80\end{bmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix}$$

$$\text{s.t.} \begin{bmatrix}3 & 1\\ 2 & 1\end{bmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} \le \begin{pmatrix}12\\ 10\end{pmatrix}$$

$$\Rightarrow -90x_1 - 80x_2$$

$$\text{s.t. } 3x_1 + x_2 \le 12$$

$$2x_1 + x_2 \le 10$$

Opt. solution: $(x_1, x_2) = (0, 10)$

Opt. cost: $z_1 = -800$

- Subproblem 2:

$$\left\{\begin{bmatrix}-70 & -60\end{bmatrix} - 0\begin{bmatrix}7 & 5\end{bmatrix}\right\}\begin{pmatrix}x_3\\ x_4\end{pmatrix}$$

$$\text{s.t. } 3x_3 + 2x_4 \le 15$$

$$x_3 + \quad x_4 \le 4$$

$$\Rightarrow \min -70x_3 - 60x_4$$

$$\text{s.t. } 3x_3 + 2x_4 \le 15$$

$$x_3 + \quad x_4 \le 4$$

Opt. solution: $(x_3, x_4) = (4, 0)$

Opt. cost: $z_2 = -280$

$$p_1^* = 0, \quad p_2^* = -280$$

Dantzig-Wolfe Decomposition Example 2

$\Rightarrow$ Bring μ_2 into the basis $\begin{pmatrix}4\\0\end{pmatrix}$

$$\Rightarrow \begin{pmatrix} \begin{pmatrix}7 & 5\end{pmatrix}\begin{pmatrix}4\\0\end{pmatrix} \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix}28\\0\\1\end{pmatrix} = a_k$$

$$B^{-1}a_k = \begin{pmatrix}1 & -60 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}28\\0\\1\end{pmatrix} = \begin{pmatrix}28\\0\\1\end{pmatrix}$$

$$bfs = \begin{pmatrix}20\\1\\1\end{pmatrix}\begin{matrix}s_1\\ \delta_2\\ \mu 1\end{matrix} \Rightarrow \theta = \frac{20}{28} = \frac{5}{7}$$

0-800-280=-1080 $\leq f^* \leq$ -800

$\Rightarrow \mu_2$ enters basis in column 1

- Iteration 2

$$\text{New } bfs = \begin{pmatrix}\frac{5}{7}\\ 1\\ \frac{2}{7}\end{pmatrix}\begin{matrix}\mu_2\\ \delta_2\\ \mu_1\end{matrix}$$

$$\text{New } B^{-1} = \begin{pmatrix}\frac{1}{28} & 0 & 0\\ 0 & 1 & 0\\ -\frac{1}{28} & 0 & 1\end{pmatrix}\begin{pmatrix}1 & -60 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix} = \begin{pmatrix}\frac{1}{28} & -\frac{60}{28} & 0\\ 0 & 1 & 0\\ -\frac{1}{28} & \frac{60}{28} & 1\end{pmatrix}$$

$$\lambda^{(2)^T} = c_B^T B^{-1} = \begin{pmatrix}-280 & -800 & 0\end{pmatrix}\begin{pmatrix}\frac{1}{28} & -\frac{60}{28} & 0\\ 0 & 1 & 0\\ -\frac{1}{28} & \frac{60}{28} & 1\end{pmatrix} = \begin{pmatrix}-10 & -200 & 0\end{pmatrix}$$

Dantzig-Wolfe Decomposition Example 2

- Subproblem 1:

$$\{[-90\ \ -80]+10[8\ \ 6]\}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\Rightarrow -10x_1 - 20x_2$$

$$\text{s.t. } 3x_1 + x_2 \le 12$$

$$2x_1 + x_2 \le 10$$

$$\Rightarrow x_1 = 0, x_2 = 10$$

Opt. cost: z_1= -200

- Subproblem 2:

$$\{[-70\ \ -60]+10[7\ \ 5]\}\begin{pmatrix} x_3 \\ x_4 \end{pmatrix}$$

$$\Rightarrow -10x_4$$

$$\text{s.t. } 3x_3 + 2x_4 \le 15$$

$$x_3 + \ \ x_4 \le 4$$

$$\Rightarrow x_3 = 0,\ x_4 = 4$$

Opt. cost: z_2=–40

$$p_1^* = 0,\ p_2^* = -40$$

$$-10*80-200-40=-1040 \le f^* \le -1000$$

Dantzig-Wolfe Decomposition Example 2

$\Rightarrow$ Bring $\begin{pmatrix} (7 \quad 5)\begin{pmatrix}0\\4\end{pmatrix} \\ 0 \\ 1 \end{pmatrix}$ into the basis $a_k = \begin{pmatrix}20\\0\\1\end{pmatrix}$

$$B^{-1}a_k = \begin{pmatrix} \frac{1}{28} & -\frac{60}{28} & 0 \\ 0 & 1 & 0 \\ -\frac{1}{28} & \frac{60}{28} & 1 \end{pmatrix}\begin{pmatrix}20\\0\\1\end{pmatrix} = \begin{pmatrix}\frac{20}{28}\\0\\\frac{8}{28}\end{pmatrix}$$

$$bfs = \begin{pmatrix}\frac{20}{28}\\1\\\frac{8}{28}\end{pmatrix} = \begin{pmatrix}\frac{5}{7}\\1\\\frac{2}{7}\end{pmatrix}\begin{matrix}\mu_2\\\delta_2\\\mu_1\end{matrix} \Rightarrow \theta = 1$$

$\Rightarrow$ Can enter in column 1 or column 3. Choose column 1

- Iteration 3

$$\text{New } B^{-1} = \begin{pmatrix} \frac{28}{20} & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{2}{5} & 0 & 1 \end{pmatrix}\begin{pmatrix} \frac{1}{28} & -\frac{60}{28} & 0 \\ 0 & 1 & 0 \\ -\frac{1}{28} & \frac{60}{28} & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{20} & -3 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{20} & 3 & 1 \end{pmatrix}$$

$$\lambda^{(3)^T} = c_B^T B^{-1} = \begin{pmatrix}-240 & -800 & 0\end{pmatrix}\begin{pmatrix} \frac{1}{20} & -3 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{20} & 3 & 1 \end{pmatrix} = \begin{pmatrix}-12 & -80 & 0\end{pmatrix}$$

$$\text{New } bfs = \begin{pmatrix}1\\1\\0\end{pmatrix}\begin{matrix}\mu_3\\\delta_2\\\mu_1\end{matrix}$$

$-12*80-80=-1040 \le f^* \le -1040$

Dantzig-Wolfe Decomposition Example 2

- Subproblem 1:

$$\{[-90\ \ -80]+12[8\ \ 6]\}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\Rightarrow 6x_1 - 8x_2$$

$$\text{s.t. } 3x_1 + x_2 \le 12$$

$$2x_1 + x_2 \le 10$$

$$\Rightarrow x_1 = 0, x_2 = 10$$

Opt. cost: $z_1 = -80$

- Subproblem 2:

$$\{[-70\ \ -60]+12[7\ \ 5]\}\begin{pmatrix} x_3 \\ x_4 \end{pmatrix}$$

$$\Rightarrow 14x_3$$

$$\text{s.t. } 3x_3 + 2x_4 \le 15$$

$$x_3 + \ x_4 \le 4$$

$$\Rightarrow x_3 = 0,\ x_4 = 0$$

Opt. cost: $z_2 = 0$

$$p_1^* = 720,\ p_2^* = 0$$

Optimal solution: $\begin{pmatrix} 0 \\ 10 \\ 0 \\ 4 \end{pmatrix}$

-12*80-80=-1040 $\le f^* \le$ -1040

Optimal cost: $-800 - 240 = -1040$

LP & Goal Programming

- Quite often, need to make decisions under multiple conflicting objectives (criteria, goals)
  - FIR filter pass and stop band constraints
  - Ship routing to minimize time, fuel and distance
- The achievement of one objective may require abandonment of another
- Typically, goals are represented as constraints, which may be infeasible
- One approach
  - Determine a set of "ideal" goals
  - Determine a metric in the objective space to measure the distance to the "ideal" goal
  - Minimize the distance to the "ideal" goals
- Two ways
  - Relative weights on different objectives (goals)
    - Makes sense if they have commensurate units (e.g., dollars)
  - Rank the goals and solve them in a **lexicographic** order
    - Optimize with regard to the 1st (most important) goal.
    - If there are multiple tie solutions, break them by optimizing with regard to the 2nd goal, etc.

Deviation variables on Goals

- Advertising problem (Whinston, 1995)

$x_1 =$ number of minutes of ads shown during football games ≥ 0

$x_2 =$ number of minutes of ads shown during comedy shows ≥ 0

Goal 1: $7x_1 + 3x_2 \geq 40M$ (High Income Men ...HIM)

Goal 2: $10x_1 + 5x_2 \geq 60M$ (Low Income People ...LIP)

Goal 3: $5x_1 + 4x_2 \geq 35M$ (High Income Women ...HIW)

Budget Constraint (in thousands):

$$100x_1 + 60x_2 \leq 600$$

- Goals are constraints
  - Use **slack** and **surplus** variables to measure "distance" to the ideal goal (RHS)

Goal 1: $7x_1 + 3x_2 + s_1^- - s_1^+ = 40M$ (HIM)

Goal 2: $10x_1 + 5x_2 + s_2^- - s_2^+ = 60M$ (LIP)

Goal 3: $5x_1 + 4x_2 + s_3^- - s_3^+ = 35M$ (HIW)

$s_i^- =$ Amount by which we are numerically under the i^{th} goal

$s_i^+ =$ Amount by which we are numerically over the i^{th} goal

  - Minimise distance to the ideal levels via either of two methods

Minimizing weighted deviation variables

- Relative weights on different goals

Goal 1 = 2*Goal 2 = 4* Goal 3

- Minimize weighted penalties for not meeting goals

$$\min\ f = 4s_1^- + 2s_2^- + s_3^-$$
$$s.t.\ 7x_1 + 3x_2 + s_1^- - s_1^+ = 40M \text{ (HIM constraint)}$$
$$10x_1 + 5x_2 + s_2^- - s_2^+ = 60M \text{ (LIP constraint)}$$
$$5x_1 + 4x_2 + s_3^- - s_3^+ = 35M \text{ (HIW constraint)}$$
$$100x_1 + 60x_2 \le 600$$
$$x_1 \ge 0; x_2 \ge 0; s_i^- \ge 0; s_i^+ \ge 0; i = 1,2,3$$

Optimal solution: $f^* = 5$

$x_1 = 6; x_2 = 0; s_1^- = 0; s_1^+ = 2;$

$s_2^- = s_2^+ = 0; s_3^- = 5; s_3^+ = 0$

Goals 1 and 2 are met.

Fail to meet Goal 3 by 5.

- What if budget constraint is a goal? Goal 4=0.02*Goal 3

$$\min\ f = 4s_1^- + 2s_2^- + s_3^- + 0.02s_4^+$$
$$s.t.\ 7x_1 + 3x_2 + s_1^- - s_1^+ = 40M \text{ (HIM constraint)}$$
$$10x_1 + 5x_2 + s_2^- - s_2^+ = 60M \text{ (LIP constraint)}$$
$$5x_1 + 4x_2 + s_3^- - s_3^+ = 35M \text{ (HIW constraint)}$$
$$100x_1 + 60x_2 + s_4^- - s_4^+ = 600 \text{ (Budget constraint)}$$
$$x_1 \ge 0; x_2 \ge 0; s_i^- \ge 0; s_i^+ \ge 0; i = 1,2,3$$

Optimal solution: $f^* = \frac{2}{3}$

$x_1 = 4\frac{1}{3}; x_2 = 3\frac{1}{3}; s_1^- = 0; s_1^+ = \frac{1}{3};$

$s_2^- = s_2^+ = 0; s_3^- = 5; s_3^+ = 0; s_4^- = 0; s_4^+ = 33\frac{1}{3}.$

Goals 1, 2 and 3 are met.

Need to spend $33,333.34 extra!

Lexicographic (Preemptive) Goal Programming-1

- Ranking goals is easier than specifying weights

Goal 1 $\succ$ Goal 2 $\succ$ Goal 3   Note: $\succ \Rightarrow$ preference ordering

- Solve a sequence of LP problems as follows:

LP1:

$$\min\ f_1 = s_1^-$$
$$s.t.\ 7x_1 + 3x_2 + s_1^- - s_1^+ = 40M \text{ (HIM constraint)}$$
~~$10x_1 + 5x_2 + s_2^- - s_2^+ = 60M$ (LIP constraint)~~

~~$5x_1 + 4x_2 + s_3^- - s_3^+ = 35M$ (HIW constraint)~~

$$100x_1 + 60x_2 + s_4 = 600 \text{ (Budget constraint)}$$
$$x_1 \geq 0; x_2 \geq 0; s_i^- \geq 0; s_i^+ \geq 0; i = 1,2,3, s_4 \geq 0$$

Optimal solution: $f_1^* = 0 \Rightarrow$ Goal 1 is met.

$$x_1 = \frac{40}{7}; x_2 = 0; s_1^- = 0; s_1^+ = 0;$$
$$s_2^- = \frac{20}{7}, s_2^+ = 0; s_3^- = \frac{45}{7}; s_3^+ = 0$$
$$s_4 = \frac{200}{7}$$

LP2:

$$\min\ f_2 = s_2^-$$
$$s.t.\quad s_1^- = 0$$
$$7x_1 + 3x_2 + s_1^- - s_1^+ = 40M \text{ (HIM constraint)}$$
$$10x_1 + 5x_2 + s_2^- - s_2^+ = 60M \text{ (LIP constraint)}$$
~~$5x_1 + 4x_2 + s_3^- - s_3^+ = 35M$ (HIW constraint)~~

$$100x_1 + 60x_2 + s_4 = 600 \text{ (Budget constraint)}$$
$$x_1 \geq 0; x_2 \geq 0; s_i^- \geq 0; s_i^+ \geq 0; i = 1,2,3; s_4 \geq 0$$

Optimal solution: $f_2^* = 0 \Rightarrow$ Goal 2 is met

$$x_1 = 6; x_2 = 0; s_1^- = 0; s_1^+ = 2;$$
$$s_2^- = s_2^+ = 0; s_3^- = 5; s_3^+ = 0; s_4 = 0.$$

Lexicographic (Preemptive) Goal Programming -2

- Solve a sequence of LP problems as follows:

LP3:

$$
\begin{aligned}
&\min\ f_3 = s_3^- \\
&s.t. \\
&\quad s_1^- = 0 \\
&\quad s_2^- = 0 \\
&\quad 7x_1 + 3x_2 + \cancel{s_1^-} - s_1^+ = 40M \text{ (HIM constraint)} \\
&\quad 10x_1 + 5x_2 + \cancel{s_2^-} - s_2^+ = 60M \text{ (LIP constraint)} \\
&\quad 5x_1 + 4x_2 + s_3^- - s_3^+ = 35M \text{ (HIW constraint)} \\
&\quad 100x_1 + 60x_2 + s_4 = 600 \text{ (Budget constraint)} \\
&\quad x_1 \geq 0; x_2 \geq 0; s_i^- \geq 0; s_i^+ \geq 0; i = 1, 2, 3, s_4 \geq 0
\end{aligned}
$$

Optimal solution: $f_3^* = 5 \Rightarrow$ Goal 3 is not met

$x_1 = 6; x_2 = 0; s_1^- = 0; s_1^+ = 2;$

$s_2^- = s_2^+ = 0; s_3^- = 5; s_3^+ = 0; s_4 = 0.$

- If budget constraint is a goal, use s_4^- and s_4^+ in the constraint
- The decision maker may want to do "what if" studies by reordering priorities

Generic Lexicographic Goal Programming

- Problem of minimizing p ranked objectives

$$L\min\{\underline{c}_1^T\underline{x}, \underline{c}_2^T\underline{x}, \ldots\ldots, \underline{c}_i^T\underline{x}, \ldots, \underline{c}_p^T\underline{x}\}$$
$$s.t. \quad A\underline{x} = \underline{b}$$
$$\underline{x} \geq \underline{0}$$

Lmin: Lexicographic ordering

- Approach

$$\min_{\underline{x}\geq\underline{0}} f_1 = \underline{c}_1^T\underline{x}$$
$$s.t. \quad A\underline{x} = \underline{b}$$

If unique solution, stop. Otherwise, continue

For $i = 2{:}p$

$$\min_{\underline{x}\geq\underline{0}} f_i = \underline{c}_i^T\underline{x} \text{ s.t. } A\underline{x} = \underline{b}; \underline{c}_j^T\underline{x} = f_j^*, j = 1, 2, .., i-1$$

If unique solution, break.

End

Summary

- Dantzig-Wolfe Decomposition is related to the cutting plane method or column generation method when the objective functions are nonlinear (see Bertsekas's Nonlinear Programming Book, Section 6.4)
- Summary
  - Basis update methods
    - Product-form of the inverse (PFI)
    - Sequential LU update
    - Sequential QR update
  - LP with upper and lower bound constraints
  - Dantzig-Wolfe decomposition
  - LP and Goal Programming