AGENDA

1. LP Duality

2. Reusing Dual Information
   - Pruning
   - Bound tightening
   - Variable selection
   - IP bound
   - Dual branching

3. Computational Results
Consider the following LP (bounds are finite):

\[ z^* = \min c^T x \]
\[ L \leq Ax \leq U \]
\[ \ell \leq x \leq u \]

Assume \( z^* \) is finite. Any dual solution \( y \) satisfies:

\[ z^* \geq \sum_{i: y_i > 0} L_i y_i + \sum_{i: y_i < 0} U_i y_i + \sum_{i: c_i - A^T_i y > 0} \ell_i (c_i - A^T_i y) + \sum_{i: c_i - A^T_i y < 0} u_i (c_i - A^T_i y) \]

\[ = z_y(L, U, \ell, u) \]

There exists an optimal dual solution \( y^* \) that satisfies

\[ z^* = z_y^*(L, U, \ell, u) \]
The following are equivalent:

- The following system is infeasible:

\[
L \leq Ax \leq U \\
\ell \leq x \leq u
\]

- There is a dual ray \( r \) to prove it:

\[
\begin{align*}
z_r(L, U, \ell, u) &= \sum_{i: r_i > 0} L_i r_i + \sum_{i: r_i < 0} U_i r_i + \sum_{i: -A_i^T r > 0} \ell_i (-A_i^T r) + \sum_{i: -A_i^T r < 0} u_i (-A_i^T r) > 0
\end{align*}
\]

All bounds are finite, so \( r \) is unconstrained.
LP DUALITY

DUAL INFORMATION IN BRANCH AND BOUND

Standard usage
- Reduced cost fixing (local and root)
- Conflict analysis

New ways
- Keep the dual rays and solutions
- Prune nodes
- Reduced cost fixing globally
- Dual ray fixing (no incumbent or bound needed)
- Variable selection
- Get tree bounds
Using dual solution $y$:

- $y$ is globally valid and gives a lower bound (weak duality) at any other node
- Compute dual objective value $z_y(L, U, \ell, u) =$

\[
\sum_{i:y_i > 0} L_i y_i + \sum_{i:y_i < 0} U_i y_i + \sum_{i:c_i - A_i^T y > 0} \ell_i (c_i - A_i^T y) + \sum_{i:c_i - A_i^T y < 0} u_i (c_i - A_i^T y)
\]

- Let $\bar{z}$ be upper bound/incumbent objective value
- Prune node if $z_y(L, U, \ell, u) \geq \bar{z}$
Using dual ray \( r \):

- \( r \) can be used to detect infeasible nodes
- Compute ray objective value \( z_r(L, U, \ell, u) = \)

\[
\sum_{i:r_i>0} L_i r_i + \sum_{i:r_i<0} U_i r_i + \sum_{i:-A_i^T r > 0} \ell_i (-A_i^T r) + \sum_{i:-A_i^T r < 0} u_i (-A_i^T r)
\]

- Prune node if \( z_r(L, U, \ell, u) > 0 \)
- What if \( z_y < \bar{z} \) and \( z_r \leq 0 \)?
Using dual solution $y$: Reduced cost fixing

$$z_y(L, U, \ell, u) =$$

$$\sum_{i: y_i > 0} L_i y_i + \sum_{i: y_i < 0} U_i y_i + \sum_{i: c_i - A_i^T y > 0} \ell_i (c_i - A_i^T y) + \sum_{i: c_i - A_i^T y < 0} u_i (c_i - A_i^T y)$$

If $c_i - A_i^T y > 0$, then increasing $\ell_i$ also increases $z_y$

If there is $\ell_i < \ell'_i < u_i$ such that

$$z_y(L, U, \ell, u) + (\ell'_i - \ell_i) (c_i - A_i^T y) \geq \bar{z},$$

then we can tighten $u_i$ to $\ell'_i$ where

$$\ell'_i = \ell_i + \frac{\bar{z} - z(L, U, \ell, u)}{c_i - A_i^T y}$$

knowing that there are no better feasible solutions where $x_i > \ell'_i$. 
Using dual ray $r$: Dual ray fixing

$$z_r(L, U, \ell, u) =$$

$$\sum_{i: r_i > 0} L_i r_i + \sum_{i: r_i < 0} U_i r_i + \sum_{i: -A_i^T r > 0} \ell_i (-A_i^T r) + \sum_{i: -A_i^T r < 0} u_i (-A_i^T r)$$

If $-A_i^T r > 0$, then increasing $\ell_i$ also increases $z_r$

If there is $\ell_i < \ell'_i < u_i$ such that

$$z_r(L, U, \ell, u) + (\ell'_i - \ell_i)(-A_i^T r) > 0,$$

then we can tighten $u_i$ to $\ell'_i$ where

$$\ell'_i = \ell_i + \frac{-z_r(L, U, \ell, u)}{-A_i^T r}$$

knowing that there are no feasible solutions where $x_i > \ell'_i$

Note that we don’t need an upper bound on the objective
REUSING DUAL INFORMATION

ROW BOUND TIGHTENING

- Same as reduced cost fixing/dual ray fixing applied to row bounds

- For \( y_i > 0 \), \( U_i \) to \( L'_i \) where

\[
L'_i = L_i + \frac{\bar{z} - z_y(L, U, \ell, u)}{y_i}
\]

- For \( r_i > 0 \), \( U_i \) to \( L'_i \) where

\[
L'_i = L_i + \frac{-z_r(L, U, \ell, u)}{r_i}
\]
## REUSING DUAL INFORMATION

### DUAL SOLUTION CACHE VS CONFLICT ANALYSIS

<table>
<thead>
<tr>
<th>Conflict analysis</th>
<th>Dual cache</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extract clauses from dual rays and solutions</td>
<td>Keep the dual information</td>
</tr>
<tr>
<td>Discard the dual solution</td>
<td>Use it later to tighten bounds</td>
</tr>
<tr>
<td>Hard to implement</td>
<td>Easy to implement</td>
</tr>
<tr>
<td>Faster to apply</td>
<td>More general</td>
</tr>
<tr>
<td>Extraction takes time</td>
<td>Usage takes more time</td>
</tr>
<tr>
<td>No local cuts</td>
<td>Works with local cuts</td>
</tr>
</tbody>
</table>

Copyright 2014, SAS Institute Inc. All rights reserved.
Estimate child node objective values using dual solution $y$

Assume we branch on $x_i$ that takes a fractional value in current node LP: $x_i = f$

Let $Z_L ([\ell_i, [f]])$ and $Z_R ([f, u_i])$ be the LP objective values of left and right child nodes

- If $c_i - A_i^T y < 0$:
  \[
  Z_L \geq \sigma^i_L = z_y(L, U, \ell, u) + ([f] - u_i)(c_i - A_i^T y)
  \]

- If $c_i - A_i^T y > 0$:
  \[
  Z_R \geq \sigma^i_R = z_y(L, U, \ell, u) + ([f] - \ell_i)(c_i - A_i^T y)
  \]

Choose the variable that maximizes $\sigma^i_L$ or $\sigma^i_R$
Estimate how close the child nodes are to be infeasible using dual ray $r$

- If $-A^T_i r < 0$:
  $$\delta^i_L = z_r(L, U, \ell, u) + ([f] - u_i)(-A^T_i r)$$

- If $-A^T_i r > 0$:
  $$\delta^i_R = z_r(L, U, \ell, u) + ([f] - \ell_i)(-A^T_i r)$$

Choose the variable that maximizes $\delta^i_L$ or $\delta^i_R$
Branch on $x_i$. Left child $[\ell_i, \lfloor f \rfloor]$, right child $[\lceil f \rceil, u_i]$

Dual solutions: $y^1, \ldots, y^5$

LP bound: 
$$Z_R \geq \max_j \{z_{y_j}(L, U, \ell, u) + (\lfloor f \rfloor - \ell_i)(c_i - A^T_i y^j)\}$$

IP bound:

- $z_j$: the dual objective value of node $j$ with variable bounds of $x_i$ replaced with $[\lfloor f \rfloor, u_i]$
- $P_R \geq \max\{z_1, \min\{z_3, \max\{z_2, \min\{z_4, z_5\}\}\}\}$
Simulate branching with a pool of dual solutions and dual rays

- Select branching variable using dual information
- Evaluate child nodes
  - Prune if proven infeasible by a dual ray
  - Prune by bound by a dual solution
- Otherwise, select a child node and branch
(Yes, SAS has a full featured optimization suite.)

- 500 benchmark instances (internal + public)
- 1 hour time limit
- Variable selection: 15% speedup + 15 instances
- Node pruning + bound tightening: 7% speedup + 8 instances
- Helps a lot for feasibility instances
- Conflict analysis has become obsolete
BLOG  OUR NEW BLOG

http://blogs.sas.com/content/operations