Branching Rules Revisited

Alexander Martin

Darmstadt University of Technology

MIP 2006
“Workshop on Mixed Integer Programming”

Miami
June 5 – 8, 2006

joint work with Tobias Achterberg and Thorsten Koch
Introduction

Branching Goals

Primal Branching

Dual Branching

Computational Results

Conclusions
Mixed Integer Program (MIP)

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax \leq b \\
& \quad l \leq x \leq u \\
& \quad x \in \mathbb{Z}^{n-p} \times \mathbb{R}^p \\
\end{align*}
\]

with \( A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, c, l, u \in \mathbb{R}^n \) and \( p \in \{0, \ldots, n\} \).
**LP based branch-and-bound**

**Input:** A (MIP)

**Output:** An opt. solution $x^*$ or the message “(MIP) is infeasible”.

1. Initialize $S := \{P_{LP}\}$, where $P_{LP}$ is the relaxation of (MIP).
   Set $c^* := \infty$.

2. If $S = \emptyset$, exit (return $x^*$ or “(MIP) is infeasible”).

3. Choose a problem $Q \in S$ and delete it from $S$.

4. Solve the LP $c_Q = \min \{c^T x \mid x \in Q\}$ with opt. solution $\bar{x}$
   ($Q$ is possibly strengthened by cuts).

5. If $c_Q \geq c^*$, goto 2.

6. If $\bar{x}$ integer, set $c^* := c_Q$ and $x^* := \bar{x}$, and goto 2.

7. **Branching:** Split $Q$ into subproblems, add them to $S$ and
goto 3.
Branching

(a) Branching on trivial inequalities (= Branching on variables)
   - Land & Powel (1979)
   - Linderoth & Savelsbergh (1999)
   - ...

(b) Branching on non-trivial inequalities
   - Clochard & Naddef (1993)
   - Borndörfer, Ferreira, Martin (1998)
   - Naddef (2002)
   - ...

Branching = Branching on linear inequalities
Variable Selection

**Input:** Subproblem $Q$ with fractional LP solution $\bar{x}$.

**Output:** $i \in I$ with $\bar{x}_i \notin \mathbb{Z}$.

1. Let $C = \{i \in I \mid \bar{x}_i \notin \mathbb{Z}\}$ be the set of branching candidates.
2. For all candidates $i \in C$, calculate a score value $s_i \in \mathbb{R}$.
3. Return an index $i \in C$ with $s_i = \max_{j \in C} \{s_j\}$. 
**Straight Away Strategies**

**Most Infeasible**

Choose variable closest to 0.5, i.e.,

\[ s_i = 0.5 - |\bar{x}_i - \lfloor \bar{x}_i \rfloor - 0.5| \]

+ seems to have the most impact on the new LPs.
+ fast to compute
**Straight Away Strategies**

**Random**

Choose variable randomly, i.e.,

\[ s_i = \text{rand()} \]

The whole topic is anyway only about “reading tea leaves”!

<table>
<thead>
<tr>
<th>Strategy</th>
<th>B&amp;B nodes total</th>
<th>B&amp;B nodes geom.</th>
<th>time (sec) total</th>
<th>time (sec) geom.</th>
<th>fails</th>
</tr>
</thead>
<tbody>
<tr>
<td>random</td>
<td>21 246 277</td>
<td>275 789.4</td>
<td>46 202.2</td>
<td>923.6</td>
<td>11</td>
</tr>
<tr>
<td>most infeasible</td>
<td>19 421 589</td>
<td>262 368.9</td>
<td>48 037.5</td>
<td>938.0</td>
<td>11</td>
</tr>
</tbody>
</table>
**Straight Away Strategies**

**Random**

Choose variable randomly, i.e.,

\[ s_i = \text{rand}() \]

The whole topic is anyway only about “reading tea leaves”!

<table>
<thead>
<tr>
<th>Strategy</th>
<th>B&amp;B nodes total</th>
<th>B&amp;B nodes geom.</th>
<th>time (sec) total</th>
<th>time (sec) geom.</th>
<th>fails</th>
</tr>
</thead>
<tbody>
<tr>
<td>random</td>
<td>21 246 277</td>
<td>275 789.4</td>
<td>46 202.2</td>
<td>923.6</td>
<td>11</td>
</tr>
<tr>
<td>most infeasible</td>
<td>19 421 589</td>
<td>262 368.9</td>
<td>48 037.5</td>
<td>938.0</td>
<td>11</td>
</tr>
</tbody>
</table>
Goals of Branching

1. Improve primal bound
2. Improve dual bound
Primal Branching Rules

- Local Branching

\[
\sum |x_j^* - x_j| < \kappa \\
\sum |x_j^* - x_j| \geq \kappa
\]

- Relaxation Induced Neighborhood Search (RINS)
  Danna, Rothberg & Le Pape (2005)

- Guided Dives
  Danna, Rothberg & Le Pape (2005)
Dual Branching Rules

Measure the success in the increase of the objective function

\[ q = \max_{x \in Q} c^T x \]

\[ q^+ \approx \max_{x \in Q^+} c^T x \]

\[ q^- \approx \max_{x \in Q^-} c^T x \]

\[ x_j = 1 \]

\[ x_j = 0 \]

\[ s_i = \text{score}(q^-, q^+) := (1 - \mu) \cdot \min\{q^-, q^+\} + \mu \cdot \max\{q^-, q^+\}, \]

where \( \mu \) is some scaling factor, e.g. \( \mu = \frac{1}{6} \).
**Strong Branching**

1. Select $C' \subseteq \{ i \mid \bar{x}_i \notin \mathbb{Z} \}$

2. For each $i \in C'$
   
   (a) Temporally set $u_i = \lfloor \bar{x}_i \rfloor$
   
   (b) Perform $\gamma$ simplex iterations yielding obj. fct. value $c_i^-$
   
   (c) Temporally set $l_i = \lceil \bar{x}_i \rceil$
   
   (d) Perform $\gamma$ simplex iterations yielding obj. fct. value $c_i^+$
   
   (e) Set $s_i = \text{score}(c_i^-, c_i^+)$

3. Return an index $i \in C$ with $s_i = \max_{j \in C} \{ s_j \}$.

see CPLEX 7.5 and Applegate, Bixby, Chvátal, Cook (2003)

**Full Strong Branching**

(i) $\gamma = \infty$

(ii) $C' = \{ i \mid \bar{x}_i \notin \mathbb{Z} \}$
Strong Branching

1. Select $C' \subseteq \{ i \mid \bar{x}_i \notin \mathbb{Z} \}$
2. For each $i \in C'$
   (a) Temporally set $u_i = \lfloor \bar{x}_i \rfloor$
   (b) Perform $\gamma$ simplex iterations yielding obj. fct. value $c_i^-$
   (c) Temporally set $l_i = \lceil \bar{x}_i \rceil$
   (d) Perform $\gamma$ simplex iterations yielding obj. fct. value $c_i^+$
   (e) Set $s_i = \text{score}(c_i^-, c_i^+)$
3. Return an index $i \in C$ with $s_i = \max_{j \in C} \{ s_j \}$.

see CPLEX 7.5 and Applegate, Bixby, Chvátal, Cook (2003)

Full Strong Branching

(i) $\gamma = \infty$
(ii) $C' = \{ i \mid \bar{x}_i \notin \mathbb{Z} \}$
Pseudocost Branching

Keep a history of the success of the variables which has already been branched on, see Benichou et al (1971).

\[ s_{i}^{+} = \frac{(\bar{c}_{Q_{i}}^{+} - \bar{c}_{Q})}{([\bar{x}_{i}] - \bar{x}_{i})} \]
\[ \sigma_{i}^{+} = \sum_{i} s_{i}^{+} \]
\[ \eta_{i}^{+} = \text{number of these problems solved} \]
\[ \psi_{i}^{+} = \frac{\sigma_{i}^{+}}{\eta_{i}^{+}}. \]

Pseudocost branching

1. Let \( C = \{ i \in I \mid x_{i} \notin \mathbb{Z} \} \) be the set of candidates.
2. For all candidates \( i \in C \), use
\[ s_{i} = \text{score}( (\bar{x}_{i} - [\bar{x}_{i}]) \cdot \psi_{i}^{-}, ([\bar{x}_{i}] - \bar{x}_{i}) \cdot \psi_{i}^{+}) \]
3. Return an index \( i \in C \) with \( s_{i} = \max_{j \in C} \{s_{j}\} \).
Hybrid Strong/Pseudocost Branching

**Problem**

Uninitialized pseudocosts $\sigma_i^+ = \eta_i^+ = 0$ at the beginning.

**Solutions**

1. Initialize pseudocost values with strong branching values
   Linderoth & Savelsbergh (1999)

2. Use strong branching up to level $d$ in the B & B tree, use pseudocost branching from level $d + 1$ on.
   see, for instance, LINDO.
Hybrid Strong/Pseudocost Branching

Problem

Uninitialized pseudocosts \( \sigma_i^+ = \eta_i^+ = 0 \) at the beginning.

Solutions

1. Initialize pseudocost values with strong branching values
   Linderoth & Savelsbergh (1999)

2. Use strong branching up to level \( d \) in the B & B tree,
   use pseudocost branching from level \( d + 1 \) on.
   see, for instance, LINDO.
Two simple new ideas

- Use strong branching not only on variables with uninitialized pseudocosts, but also on variables with unreliable pseudocosts. The pseudocosts of variable $i$ are called unreliable, if

$$\min\{\eta_i^-, \eta_i^+\} < \eta_{\text{rel}},$$

with $\eta_{\text{rel}} \in \mathbb{N}$ being the reliability parameter.

- Select the set $C$ of candidates dynamically. Introduce a so-called look ahead parameter $\lambda$. If the best score does not change for $\lambda$ variables, then stop calling strong branching.
Reliability Branching

1. Let $C = \{i \in I \mid \bar{x}_i \notin \mathbb{Z}\}$ be the set of candidates.

2. Sort $C$ according to non-increasing pseudocosts.

   For all $i \in C$ with $\min\{\eta_i^-, \eta_i^+\} < \eta_{rel}$, do:

   (a) Perform $\gamma$ simplex iterations on $Q_i^-$ and $Q_i^+$.
       Let $\tilde{\Delta}_i^-$ and $\tilde{\Delta}_i^+$ be the objective gains.

   (b) Update the pseudocosts $\Psi_i^-$ and $\Psi_i^+$ with $\tilde{\Delta}_i^-$ and $\tilde{\Delta}_i^+$.

   (c) Update the score $s_i = \text{score}(\tilde{\Delta}_i^-, \tilde{\Delta}_i^+)$.

   (d) If the maximum score $s^* = \max_{j \in C}\{s_j\}$ has not changed for $\lambda$ consecutive score updates, goto 3.

3. Return an index $i \in C$ with $s_i = \max_{j \in C}\{s_j\}$. 
Branching Rule Classification

Reliability Branching

Hybrid Strong/Pseudocost Branching

Full Strong Branching

Lookahead

Depth

Pseudocost

Strong Branching
Test Set

Instances are taken from

- Miplib 2003, see http://miplib.zib.de
- Mittelmann 2003, see http://plato.asu.edu/bench.html

where CPLEX 9.0 needs

- at least 5000 B& B nodes
- at most 1 hour CPU time
  (on a 833 MHz Alpha with 4 MB Cache and 2 GB RAM)

These are 24 instances:

- aflow30a
- cap6000
- gesa2-o
- mas74
- mas76
- misc07
- pp08aCUTS
- qiu
- rout
- vpm2
- ran8x32
- ran10x26
- ran13x13
- mas284
- prod1
- bc1
- bienst1
- neos2
- swath1
- swath2
- neos7
- pk1
- neos3
- ran12x21
<table>
<thead>
<tr>
<th>Strategy</th>
<th>B&amp;B nodes total</th>
<th>B&amp;B nodes geom.</th>
<th>time (sec) total</th>
<th>time (sec) geom.</th>
<th>strong branchings total</th>
<th>strong branchings geom.</th>
<th>fails</th>
</tr>
</thead>
<tbody>
<tr>
<td>random</td>
<td>21 246 277</td>
<td>275 789.4</td>
<td>46 202.2</td>
<td>923.6</td>
<td>0</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>most infeasible</td>
<td>19 421 589</td>
<td>262 368.9</td>
<td>48 037.5</td>
<td>938.0</td>
<td>0</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>pseudocost</td>
<td>9 381 001</td>
<td>88 706.8</td>
<td>19 945.8</td>
<td>283.4</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>full strong</td>
<td>929 347</td>
<td>7 241.7</td>
<td>26 397.2</td>
<td>504.4</td>
<td>15 569 652</td>
<td>146 784.2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lookahead = 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>strong/pscost (5)</td>
<td>9 698 397</td>
<td>79 535.5</td>
<td>19 487.8</td>
<td>249.3</td>
<td>5 792</td>
<td>216.2</td>
<td>2</td>
</tr>
<tr>
<td>strong/pscost (10)</td>
<td>8 251 942</td>
<td>65 966.3</td>
<td>16 499.4</td>
<td>229.6</td>
<td>74 812</td>
<td>2 284.2</td>
<td>2</td>
</tr>
<tr>
<td>strong/pscost (15)</td>
<td>7 982 847</td>
<td>57 976.8</td>
<td>17 855.3</td>
<td>258.8</td>
<td>523 377</td>
<td>8 137.8</td>
<td>2</td>
</tr>
<tr>
<td>strong/pscost (20)</td>
<td>7 890 374</td>
<td>47 958.5</td>
<td>19 175.6</td>
<td>293.5</td>
<td>2 825 100</td>
<td>17 780.2</td>
<td>2</td>
</tr>
<tr>
<td>reliability (1)</td>
<td>9 000 334</td>
<td>70 013.6</td>
<td>17 199.6</td>
<td>216.4</td>
<td>39 126</td>
<td>374.8</td>
<td>1</td>
</tr>
<tr>
<td>reliability (4)</td>
<td>6 906 698</td>
<td>53 522.9</td>
<td>13 402.9</td>
<td>178.2</td>
<td>110 628</td>
<td>1 176.7</td>
<td>0</td>
</tr>
<tr>
<td>reliability (8)</td>
<td>7 937 968</td>
<td>48 772.8</td>
<td>11 132.7</td>
<td>170.5</td>
<td>117 643</td>
<td>1 850.3</td>
<td>0</td>
</tr>
<tr>
<td>reliability (16)</td>
<td>6 022 024</td>
<td>44 649.9</td>
<td>10 782.6</td>
<td>179.0</td>
<td>187 578</td>
<td>3 640.6</td>
<td>0</td>
</tr>
<tr>
<td>reliability (32)</td>
<td>7 940 797</td>
<td>39 655.2</td>
<td>11 103.0</td>
<td>184.2</td>
<td>253 014</td>
<td>5 837.8</td>
<td>0</td>
</tr>
<tr>
<td>strong branching</td>
<td>1 325 589</td>
<td>11 639.5</td>
<td>20 427.2</td>
<td>353.2</td>
<td>9 965 454</td>
<td>86 188.6</td>
<td>3</td>
</tr>
<tr>
<td>Strategy</td>
<td>B&amp;B nodes</td>
<td>time (sec)</td>
<td>strong branchings</td>
<td>fails</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------</td>
<td>----------------</td>
<td>-------------------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>total geom.</td>
<td>total geom.</td>
<td>total geom.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lookahead = 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>strong/pscost (5)</td>
<td>8 653 318</td>
<td>17 389.5</td>
<td>7 397</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>74 730.8</td>
<td>239.6</td>
<td>268.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>strong/pscost (10)</td>
<td>8 332 543</td>
<td>18 362.5</td>
<td>102 082</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>69 489.0</td>
<td>253.9</td>
<td>2 523.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>strong/pscost (15)</td>
<td>7 456 685</td>
<td>20 479.8</td>
<td>750 577</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>59 398.8</td>
<td>295.7</td>
<td>9 983.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>strong/pscost (20)</td>
<td>7 551 419</td>
<td>22 388.8</td>
<td>3 695 577</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>48 736.5</td>
<td>343.1</td>
<td>20 837.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reliability (1)</td>
<td>8 663 537</td>
<td>16 753.8</td>
<td>53 557</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>69 501.0</td>
<td>217.2</td>
<td>429.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reliability (4)</td>
<td>8 338 386</td>
<td>12 497.2</td>
<td>74 906</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>54 937.7</td>
<td>179.2</td>
<td>1 104.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reliability (8)</td>
<td>7 813 409</td>
<td>12 380.2</td>
<td>133 545</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>48 377.3</td>
<td>170.2</td>
<td>1 998.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reliability (16)</td>
<td>7 579 400</td>
<td>11 946.7</td>
<td>185 136</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>43 311.9</td>
<td>171.3</td>
<td>3 589.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reliability (32)</td>
<td>7 207 836</td>
<td>11 835.7</td>
<td>259 482</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>42 047.5</td>
<td>186.5</td>
<td>5 913.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>strong branching</td>
<td>1 294 569</td>
<td>25 619.4</td>
<td>12 651 504</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12 714.7</td>
<td>468.0</td>
<td>119 799.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strategy</td>
<td>B&amp;B nodes</td>
<td>time (sec)</td>
<td>strong branchings</td>
<td>fails</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>------------</td>
<td>--------------</td>
<td>--------------------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>total</td>
<td>total geom.</td>
<td>total geom.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lookahead = (\infty)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>strong/psc\textsuperscript{cost} (5)</td>
<td>8 498 292</td>
<td>18 116.9</td>
<td>14 675</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>71 817.4</td>
<td>229.3</td>
<td>489.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>strong/psc\textsuperscript{cost} (10)</td>
<td>9 247 636</td>
<td>20 472.1</td>
<td>154 458</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>70 125.8</td>
<td>276.2</td>
<td>3 870.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>strong/psc\textsuperscript{cost} (15)</td>
<td>6 670 440</td>
<td>19 907.4</td>
<td>890 187</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>56 926.6</td>
<td>312.2</td>
<td>13 127.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>strong/psc\textsuperscript{cost} (20)</td>
<td>7 627 640</td>
<td>23 538.5</td>
<td>3 842 516</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>48 547.0</td>
<td>373.6</td>
<td>26 557.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reliability (1)</td>
<td>7 747 290</td>
<td>15 825.8</td>
<td>48 162</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>72 159.1</td>
<td>220.0</td>
<td>408.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reliability (4)</td>
<td>9 068 723</td>
<td>14 258.1</td>
<td>78 625</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>58 886.4</td>
<td>195.8</td>
<td>1 096.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reliability (8)</td>
<td>8 551 045</td>
<td>13 563.0</td>
<td>135 541</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>54 118.3</td>
<td>189.6</td>
<td>2 042.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reliability (16)</td>
<td>6 567 432</td>
<td>12 766.4</td>
<td>196 220</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>49 839.9</td>
<td>191.6</td>
<td>3 601.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reliability (32)</td>
<td>7 502 942</td>
<td>12 393.6</td>
<td>281 822</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>41 636.1</td>
<td>192.5</td>
<td>6 000.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>strong branching</td>
<td>1 163 822</td>
<td>26 176.4</td>
<td>12 793 364</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11 355.7</td>
<td>500.3</td>
<td>127 737.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPLEX/SIP cuts</td>
<td>10 467 429</td>
<td>18 617.4</td>
<td>—</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>79 269.0</td>
<td>215.8</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Nodes versus Time
Time versus Strong Branchings

Lookahead=4

- Reliability Branching
- Hybrid Strong/Pseudocost Branching
- Strong Branching

Pseudocost

Depth

Strong Branchings

Reliability

Strong Branchings

1 10 100 1000 10000 100000

Time [s]
Nodes versus Strong Branching

Lookahead=4

- Pseudocost
- Hybrid Strong/Pseudocost Branching
- Reliability Branching
- Strong Branching
- Depth
- Reliability
Conclusions

Summary

- most infeasible as good as random
- *strong branching* is best with respect to number of nodes, but not with respect to time
- *reliability branching* outperforms *hybrid strong/pseudocost branching*
- Increasing $\eta_{rel}$ (or the depth $d$) decreases the number of nodes
- Currently best choice $\eta_{rel} = 8$ and $\lambda = 4$.

Open

- Bridge the gap to *full strong branching* without increasing the running time.
- Missing Theory !?
- Branching versus modeling with binary variables.