Cost-Bounded Binary Decision Diagrams for 0-1 Programming

Tarik Hadzic
IT University of Copenhagen

John Hooker
Carnegie Mellon University

INFORMS 2007, Seattle
Motivation

• Perform **postoptimality analysis** for 0-1 programming.

• **Problem**: It is hard to reason about the entire solution space.

• **Solution**: Represent the set of near-optimal solutions as a Binary Decision Diagram.
  – This was done in previous work (H&H 2006).
Motivation

• Today’s focus: **Scalability**
  – How large do BDDs grow with problem size?
  – How can we minimize the growth?

• We introduce **sound** BDDs.
  – Much **smaller** than the full BDD.
  – Provide **exact** postoptimality analysis for near-optimal solutions.
Types of Analysis

• Characterization of optimal or near-optimal solutions.
  – How much freedom is there to alter solution without much sacrifice in solution quality?

• Sensitivity analysis.
  – Which problem data significantly affect the solution?

• Online what-if queries.
  – What if I fix certain variables?
Basic Idea

• Use **reduced ordered binary decision diagrams** (BDDs) as a compact representation of the set of feasible or near-optimal solutions.
  – We can extract information from BDDs in real time.
  – Although exponentially large in the worst case, BDDs can be compact for important constraints.
Binary Decision Diagrams

- A reduced ordered BDD for a constraint set is a compact representation of the branching tree for a given branching order.
  - If both branches from a node lead to identical subtrees, remove the node.
  - If two subtrees are identical, superimpose them.
Branching tree for 0-1 inequality

\[ 2x_0 + 3x_1 + 5x_2 + 5x_3 \geq 7 \]

\[ x_0 = 1 \quad \text{or} \quad x_0 = 0 \]
Branching tree for 0-1 inequality

\[ 2x_0 + 3x_1 + 5x_2 + 5x_3 \geq 7 \]

\[ x_0 = 1 \]

\[ x_0 = 0 \]

Reduced ordered BDD

If an edge skips a variable, both assignments are allowed (\( x_3 = 0 \) and \( x_3 = 1 \))
In practice, BDDs are generated **bottom-up**.

First construct a BDD for every constraint and then conjoin the BDDs.

as generated by software
(CLab, BuDDy)
The BDD for a knapsack constraint can be surprisingly small…

\[ 300x_0 + 300x_1 + 285x_2 + 285x_3 + 265x_4 + 265x_5 + 230x_6 + 230x_7 + 230x_8 + 190x_9 + 190x_{10} + 200x_{11} + 200x_{12} + 200x_{13} + 200x_{14} + 200x_{15} + 200x_{16} + 200x_{17} + 400x_{18} \geq 2700 \]

The 0-1 inequality has 117,520 minimal feasible solutions.

Or equivalently,

\[ 400x_0 + 200x_1 + 400x_2 + 400x_3 + 400x_4 + 400x_5 + 400x_6 + 400x_7 + 400x_8 + 200x_{10} + 200x_{11} + 400x_{12} + 400x_{13} + 400x_{14} + 400x_{15} + 400x_{16} + 200x_{17} + 400x_{18} \leq 2700 \]

But its reduced BDD has only 152 nodes…
• However, the BDD for multiple constraints can explode.
Optimization Over BDDs

- We want to solve a 0-1 programming model with an additively separable objective function

\[
\min \sum_{j=1}^{n} c_j(x_j) \\
g_i(x) \geq b_i, \quad i = 1, \ldots, m \\
x_j \in \{0, 1\}, \quad j = 1, \ldots, n
\]

Can be straightforwardly extended to general integer programming
Optimization Over BDDs

• We want to solve a 0-1 programming model with an additively separable objective function

\[ \min \sum_{j=1}^{n} c_j(x_j) \]
\[ g_i(x) \geq b_i, \quad i = 1, \ldots, m \]
\[ x_j \in \{0,1\}, \quad j = 1, \ldots, n \]

Can be straightforwardly extended to general integer programming

• If we represent the constraints \( g_i(x) \geq b_i \) as a BDD, then we can solve the problem by finding a shortest path in the BDD with appropriate edge lengths…
\[\min 2x_0 - 3x_1 + 4x_2 + 6x_3 \quad \text{subject to} \quad 2x_0 + 3x_1 + 5x_2 + 5x_3 \geq 7\]

Edge lengths reflect coefficients in the objective function.
\[
\min 2x_0 - 3x_1 + 4x_2 + 6x_3 \quad \text{subject to} \quad 2x_0 + 3x_1 + 5x_2 + 5x_3 \geq 7
\]

\[
2 \cdot 1 + \min_{x_1 \in \{0,1\}} \{-3x_1\} \rightarrow -1
\]

Edge lengths reflect coefficients in the objective function
min \(2x_0 - 3x_1 + 4x_2 + 6x_3\) \quad \text{subject to} \quad 2x_0 + 3x_1 + 5x_2 + 5x_3 \geq 7

Edge lengths reflect coefficients in the objective function.
We conduct postoptimality analysis by analyzing shortest and near-shortest paths.

Shortest path has length 1.

Optimal solution: 
\((x_0, x_1, x_2, x_3) = (0, 1, 1, 0)\)

Set to minimizing value.

\[\min 2x_0 - 3x_1 + 4x_2 + 6x_3 \text{ subject to } 2x_0 + 3x_1 + 5x_2 + 5x_3 \geq 7\]
Cost-Based Domain Analysis

• Consider again

\[
\min \sum_{j=1}^{n} c_j(x_j)
\]

\[
g_i(x) \geq b_i, \quad i = 1, \ldots, m
\]

\[
x_j \in \{0, 1\}, \quad j = 1, \ldots, n
\]

• What values can \(x_j\) take without forcing the objective function value above \(c_{opt} + \Delta\)?

Optimal value
Example: Network Reliability

• Minimize cost subject to a bound on reliability
  – System of 5 bridges:

\[
R = R_1 R_2 + (1 - R_2) R_3 R_4 + (1 - R_1) R_2 R_3 R_4 + R_1 (1 - R_2) (1 - R_3) R_4 R_5 + (1 - R_1) R_2 R_3 (1 - R_4) R_5
\]
The problem:

\[
\begin{align*}
\min \sum_{j} c_j x_j \\
R &\geq R_{\min} \\
R &= R_1 R_2 + (1 - R_2) R_3 R_4 + (1 - R_1) R_2 R_3 R_4 \\
&\quad + R_1 (1 - R_2) (1 - R_3) R_4 R_5 + (1 - R_1) R_2 R_3 (1 - R_4) R_5 \\
R_j &= 1 - (1 - r_j)^{x_j}, \text{ all } j \\
x_j &\in \{0, 1, 2, 3\}
\end{align*}
\]

Set \( R_{\min} = 60 \) in all examples
Cost-based domain analysis

308 nodes in BDD
1.1 seconds to compile BDD

\[ r = (0.9, 0.85, 0.8, 0.9, 0.95) \]
\[ c = (25, 35, 40, 10, 60) \]

<table>
<thead>
<tr>
<th>( c_{opt} + \Delta )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( x_4 )</th>
<th>( x_5 )</th>
<th>( R )</th>
</tr>
</thead>
<tbody>
<tr>
<td>50:</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>72</td>
</tr>
<tr>
<td>60:</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.2</td>
<td></td>
<td>79</td>
</tr>
<tr>
<td>85:</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>84</td>
</tr>
<tr>
<td>90:</td>
<td></td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td>86</td>
</tr>
<tr>
<td>95:</td>
<td>2</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>88</td>
</tr>
<tr>
<td>100:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>95</td>
</tr>
<tr>
<td>120:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>97</td>
</tr>
<tr>
<td>125:</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>155:</td>
<td></td>
<td>3</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>160:</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>170:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>180:</td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>230:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
Domain analysis with respect to $R$

Same BDD as before

<table>
<thead>
<tr>
<th>$R$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>99:</td>
<td>1,2</td>
<td>1,2</td>
<td>1,2</td>
<td>1,2</td>
<td>0,1,2,3</td>
</tr>
<tr>
<td>98:</td>
<td></td>
<td>0,3</td>
<td>0,3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>97:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0,3</td>
</tr>
<tr>
<td>95:</td>
<td>0,3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7 bridges
Cost-based domain analysis

1779 nodes in BDD
14.8 seconds to compile BDD

<table>
<thead>
<tr>
<th>( c_{opt} + \Delta )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( x_4 )</th>
<th>( x_5 )</th>
<th>( x_6 )</th>
<th>( x_7 )</th>
<th>( R )</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td>72.2</td>
</tr>
<tr>
<td>11:</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>13:</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td>82.9</td>
</tr>
<tr>
<td>14:</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:</td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
<td>84.6</td>
</tr>
<tr>
<td>18:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>3</td>
<td></td>
<td>95.2</td>
</tr>
<tr>
<td>19:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>20:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>97.2</td>
</tr>
<tr>
<td>23:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>99.2</td>
</tr>
<tr>
<td>34:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>99.4</td>
</tr>
<tr>
<td>40:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>99.6</td>
</tr>
<tr>
<td>43:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>99.7</td>
</tr>
<tr>
<td>47:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>99.8</td>
</tr>
<tr>
<td>54:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>99.9</td>
</tr>
</tbody>
</table>
Domain analysis with respect to $R$

Same BDD as before

<table>
<thead>
<tr>
<th>$R$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$x_6$</th>
<th>$x_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>99.9</td>
<td>2,3</td>
<td>0,1,2,3</td>
<td>1,2,3</td>
<td>0,1,2,3</td>
<td>0,1,2,3</td>
<td>2,3</td>
<td>2,3</td>
</tr>
<tr>
<td>99.8</td>
<td>1</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>99.5</td>
<td>0</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>99.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>97.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
12 bridges
Cost-based domain analysis

69,457 nodes in BDD
2933 seconds to compile BDD

c_{opt} + \Delta | x_1 | x_2 | x_3 | x_4 | x_5 | x_6 | x_7 | x_8 | x_9 | x_{10} | x_{11} | x_{12} | R
---|---|---|---|---|---|---|---|---|---|---|---|---|
180 | 1 | 0 | 2 | 3 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 80
185 | 3 | 2 | | | | | | | | | | | 82
190 | | | 3 | | | | | | | | | | 83
195 | | | | 1 | | | | | | | | | 86
200 | | | | | 1 | | | | | | | | 88
205 | | | | | | | | | | | | | 89
210 | 2 | | | | | | | | | | | | 90
215 | | | | | | | | | | | | | 91
220 | | | | | | | | | | | | | 92
225 | | | 1 | | | | | | | | | | 93
230 | 0 | 0 | 1,2 | | | | | | | | | | 94
235 | | | 1 | | | | | | | | | | 95
240 | 1 | 1 | 2 | 3 | 2 | 1 | | | | | | | 96
250 | 0 | | 0,1 | | | | | | | | | | 97
255 | | | | | | | | | | | | | 98
260 | 3 | | | | | | | | | | | | 99
265 | | | | | | | | | | | | | 100
270 | | 2 | 3 | 3 | | | | | | | | | 101
290 | | | 3 | | | | | | | | | | 102
300 | | 2 | | | | | | | | | | | 103
305 | | | 3 | | | | | | | | | | 104
310 | | | | 3 | | | | | | | | | 105
315 | | | 3 | | | | | | | | | | 106
340 | | | | | | | | | | | | | 107
360 | 3 | | | | | | | | | | | | 108
365 | | | | | | | | | | | | | 109
380 | | | | | | | | | | | | | 110
430 | | | | | | | | | | | | | 111
485 | | | | | | | | | | | | | 112
Domain analysis with respect to $R$

Same BDD as before

<table>
<thead>
<tr>
<th>$R$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$x_6$</th>
<th>$x_7$</th>
<th>$x_8$</th>
<th>$x_9$</th>
<th>$x_{10}$</th>
<th>$x_{11}$</th>
<th>$x_{12}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>99:</td>
<td>0.3</td>
<td>0.3</td>
<td>1.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>1.3</td>
<td>0.3</td>
<td>0.3</td>
<td>1.3</td>
<td>1.3</td>
<td>1.3</td>
</tr>
<tr>
<td>98:</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>96:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Reducing BDD Growth

• It suffices to use a BDD with the same near-optimal solutions as the original BDD. We assume $\Delta \leq \Delta_{\text{max}}$. 

$$B_{\Delta_{\text{max}}}$$
Reducing BDD Growth

• It suffices to use a BDD with the same near-optimal solutions as the original BDD. We assume $\Delta \leq \Delta_{\text{max}}$.

$$\text{Sol} = \begin{cases} \text{feasible solutions} \\ \end{cases}$$

$$\text{Sol}_{\Delta_{\text{max}}} = \begin{cases} \text{feasible solutions with value } \leq c_{\text{opt}} + \Delta_{\text{max}} \end{cases}$$

$$B = \text{original BDD}$$

$$B_{\Delta_{\text{max}}} = \text{represents } \text{Sol}_{\Delta_{\text{max}}}$$

$$B_{\Delta_{\text{max}}}$$
Reducing BDD Growth

• It suffices to use a BDD with the same near-optimal solutions as the original BDD. We assume $\Delta \leq \Delta_{\text{max}}$.

\[ Sol = \begin{cases} \text{feasible solutions} \end{cases} \]

\[ B = \text{original BDD} \]

\[ Sol_{\Delta_{\text{max}}} = \begin{cases} \text{feasible solutions with value } \leq c_{\text{opt}} + \Delta_{\text{max}} \end{cases} \]

\[ B_{\Delta_{\text{max}}} = \text{BDD that represents } Sol_{\Delta_{\text{max}}} \]

• Unfortunately, $B_{\Delta_{\text{max}}}$ can be exponentially larger than $B$.
  – Even though it represents a smaller set of solutions.
Reducing BDD Growth

• It suffices to use a BDD with the same near-optimal solutions as the original BDD. We assume $\Delta \leq \Delta_{\text{max}}$.

$$\text{Sol} = \left\{ \text{feasible solutions} \right\}$$

$$\text{Sol}_{\Delta_{\text{max}}} = \left\{ \text{feasible solutions with value} \leq c_{\text{opt}} + \Delta_{\text{max}} \right\}$$

$$B = \text{original BDD}$$

$$B_{\Delta_{\text{max}}} = \text{represents } \text{Sol}_{\Delta_{\text{max}}}$$

• We will construct a smaller BDD $B'(\Delta_{\text{max}})$ that is sound: $B'(\Delta_{\text{max}})_{\Delta_{\text{max}}} = B_{\Delta_{\text{max}}}$
  
  – It has the same near optimal solutions as $B$. 
Reducing BDD Growth

\[ \text{Sol} = \text{Solutions represented by } B \]

\[
\begin{align*}
\text{Value} & > c_{opt} + \Delta_{max} \\
\text{Value} & < c_{opt} + \Delta_{max}
\end{align*}
\]
Reducing BDD Growth

\[ Sol = \text{Solutions represented by } B \]

\[
\begin{align*}
\text{Value} & > c_{\text{opt}} + \Delta_{\text{max}} \\
\text{Value} & < c_{\text{opt}} + \Delta_{\text{max}}
\end{align*}
\]

\[ Sol_{\Delta_{\text{max}}} = \text{Solutions represented by } B_{\Delta_{\text{max}}} \]
Reducing BDD Growth

\[ \text{Sol} = \text{Solutions represented by } B \]

Solutions represented by \( B'(\Delta_{\text{max}}) \)

\[ \text{Value} > c_{\text{opt}} + \Delta_{\text{max}} \]
\[ \text{Value} < c_{\text{opt}} + \Delta_{\text{max}} \]
Pruning and Contracting

• We are unaware of a polytime exact method for constructing the \textit{smallest sound} BDD.

• We use two heuristic methods for generating \textit{small sound} BDDs during compilation:
  – Pruning edges
  – Contracting nodes
Pruning

- Delete all edges that belong only to paths longer than $c_{\text{opt}} + \Delta_{\text{max}}$. 

Edge that belongs only to long paths.
Pruning

- Delete all edges that belong only to paths longer than $c_{\text{opt}} + \Delta_{\text{max}}$. 
Pruning

• Delete all edges that belong only to paths longer than $c_{\text{opt}} + \Delta_{\text{max}}$. 

If another edge now belongs only to long paths
Pruning

• Delete all edges that belong only to paths longer than $c_{opt} + \Delta_{\text{max}}$.

Delete it, too.
Pruning

- Delete all edges that belong only to paths longer than $c_{\text{opt}} + \Delta_{\text{max}}$.

And simplify the BDD.
And simplify the BDD.

- Delete all edges that belong only to paths longer than $c_{opt} + \Delta_{max}$. 
Contracting

- Remove a node if this creates no new paths shorter than $C_{opt} + \Delta_{max}$.
Experimental Results

- We solve the 0-1 problem

\[
\min cx \\
Ax \geq b \\
x \in \{0,1\}^n
\]

\[b_i = \alpha \sum_j A_{ij}\]

\(A_{ij}\) drawn uniformly from [0, r]
### Experimental Results

- **20 variables, 5 constraints**
- **$c_{opt} = 101$, $c_{max} = 588$**

| $\Delta_{max}$ | $|B|$   | $|B_{\Delta_{max}}|$ | $|B'(\Delta_{max})|$ |
|---------------|--------|---------------------|---------------------|
| 0             | 8,566  | 20                  | 5                   |
| 40            | 742    | 524                 |                     |
| 80            | 4,388  | 3,456               |                     |
| 120           | 11,217 | 7,034               |                     |
| 200           | 16,285 | 8,563               |                     |
| 240           | 13,557 | 8,566               |                     |

*Sound BDD*
\[
\Delta_{\text{max}} / (c_{\text{max}} - c_{\text{opt}})
\]

\[
\text{Size of BDD}
\]

\[
\text{Exact} \quad \cdots \cdots \\
\text{Pruned and Contracted} \quad \cdots \cdots \\
\]

\[
|B_{\Delta_{\text{max}}}| \
\]

\[
|B'(\Delta_{\text{max}})| \
\]

\[
|\text{Sol}| = 556 \\
|\text{Sol}| = 72,896 \\
|\text{Sol}| = 449,102 \\
|\text{Sol}| = 929,260
\]
Experimental Results

- 30 variables, 6 constraints
- $c_{\text{opt}} = 36$, $c_{\text{max}} = 812$

| $\Delta_{\text{max}}$ | $|B|$       | $|B_{\Delta_{\text{max}}}|$ | $|B'(\Delta_{\text{max}})|$ |
|----------------------|------------|-----------------------------|-----------------------------|
| 0                    | 925,610    | 30                          | 10                          |
| 50                   | 3,428      | 2,006                        |                             |
| 150                  | 226,683    | 262,364                      |                             |
| 200                  | 674,285    | 568,863                      |                             |
| 250                  | 1,295,465  | 808,425                      |                             |
| 300                  | 1,755,378  | 905,602                      |                             |
Experimental Results

- 40 variables, 8 constraints
- $C_{opt} = 110$, $C_{max} = 1241$

| $\Delta_{max}$ | $|B|$ | $|B'_{\Delta_{max}}|$ |
|----------------|------|---------------------|
| 0              | ?    | 12                  |
| 15             | 40   | 1,143               |
| 35             | 402  | 3,003               |
| 70             | 11,040| 7,327              |
| 100            | 404,713| 223,008            |
| 140            | ?    | 52,123              |
Experimental Results

- 60 variables, 10 constraints
- $c_{\text{min}} = 67$, $c_{\text{max}} = 3179$

| $\Delta_{\text{max}}$ | $|B|$ | $|B_{\Delta_{\text{max}}}|$ | $|B'(\Delta_{\text{max}})|$ |
|-----------------------|------|---------------------|------------------|
| 0                     | ?    | 60                  | 7                |
| 50                    |      | 5,519               | 1,814            |
| 100                   | 111,401 |                    | 78,023          |
Experimental Results

Results when tightness $\alpha$ is gradually reduced
Experimental Results

- MIPLIB instances
- $\Delta_{\text{max}} = 0$ (BDD represents all optimal solutions)

| Instance | $|B_{\Delta_{\text{max}}}|$ | $|B'(\Delta_{\text{max}})|$ |
|----------|----------------|-------------------|
| lseu     | 99             | 19                |
| p0033    | 375            | 41                |
| p0201    | 310,420        | 737               |
| stein27  | 25,202         | 6,260             |
| stein45  | 5,102,257      | 1,765             |
Conclusions and Future Work

• Cost-bounded BDDs provide reasonable scalability for BDD-based postoptimality analysis in 0-1 linear programming.

Future work:

• Tests on nonlinear, nonconvex 0-1 problems
  – Nonlinearity, nonconvexity should not be a major factor.
• Extension to general integer problems.
  – Straightforward; a matter of implementation.
• Extension to MILP.
  – ??