Postoptimality Analysis Using Multivalued Decision Diagrams

Tarik Hadzic
Cork Constraint Computation Centre
J. N. Hooker
Carnegie Mellon University

London School of Economics
June 2008

Postoptimality Analysis with MDDs

- **Goal:** Perform **postoptimality analysis** on discrete optimization problems.

Postoptimality Analysis with MDDs

- **Goal:** Perform **postoptimality analysis** on discrete optimization problems.

- **Difficulty**: It is hard to reason about the entire solution space.

Postoptimality Analysis with MDDs

- **Goal:** Perform **postoptimality analysis** on discrete optimization problems.

- **Difficulty**: It is hard to reason about the entire solution space.

- **Solution**: Use a **knowledge representation** approach.
  - Represent the set of near-optimal solutions as a **Multivalued Decision Diagram**.
  - MDD can be queried in real time.

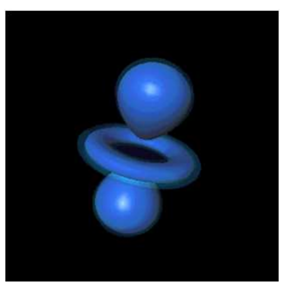
Why Postoptimality Analysis?

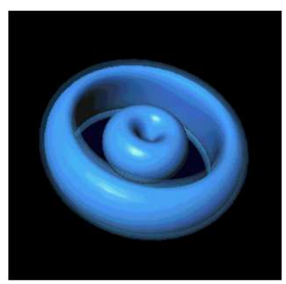
- A single optimal solution offers limited information.
- Postoptimality analysis can provide insight into model.
  - It takes a model seriously as a *model*.

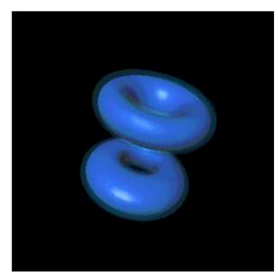
Why Postoptimality Analysis?

- A single optimal solution offers limited information.
- Postoptimality analysis can provide insight into model.
  - It takes a model seriously as a *model*.
  - Compare: models in the physical sciences.
    - Schrödinger's wave equation

$$E\,\psi(x) = -\frac{\hbar^2}{2m}\frac{d^2\psi(x)}{dx^2} + V(x)\psi(x)$$







Postoptimality Analysis

- Types of analysis:
  - Characterize the set of optimal or near-optimal solutions.
    - How much freedom to alter solution without much sacrifice?

Postoptimality Analysis

- Types of analysis:
  - Characterize the set of optimal or near-optimal solutions.
    - How much freedom to alter solution without much sacrifice?
  - Measure sensitivity of optimal value to problem data.
    - Which data really matter?

Postoptimality Analysis

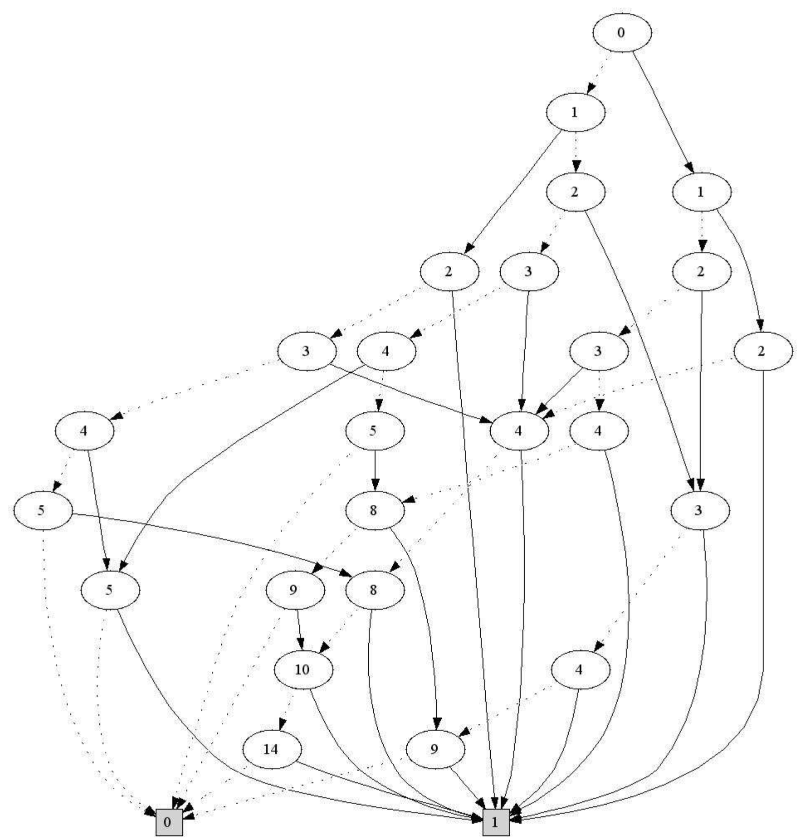
- Types of analysis:
  - Characterize the set of optimal or near-optimal solutions.
    - How much freedom to alter solution without much sacrifice?
  - Measure sensitivity of optimal value to problem data.
    - Which data really matter?
  - Do this online in response to what-if queries.
    - What if I fix certain variables?

Postoptimality Analysis

- Postoptimality analysis tends to be computationally intractable for discrete optimization.
  - Particularly real-time or interactive analysis.

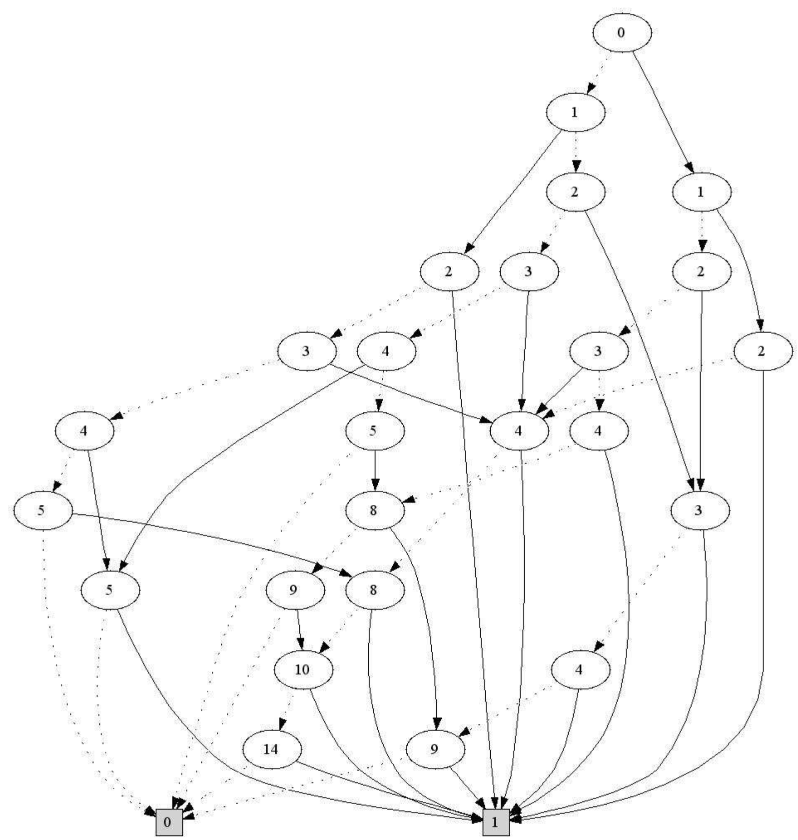
Analysis Using MDDs

- Proposal: Use **binary decision diagrams** (BDDs).
  - For binary variables.



Analysis Using MDDs

- Proposal: Use **binary decision diagrams** (BDDs)
  - For binary variables.
- Or **MDDs**
  - For multivalued variables
- Common applications:
  - VLSI verification & model checking
  - Configuration problems



Analysis Using MDDs

- MDD can grow exponentially with number of variables
  - …but can be surprisingly compact in important cases.
  - Size can be sensitive to variable ordering.

Analysis Using MDDs

- MDD can grow exponentially with number of variables
  - …but can be surprisingly compact in important cases.
  - Size can be sensitive to variable ordering.
- MDD doesn't care whether the problem is linear or nonlinear, convex or nonconvex.
  - Functions can take any form (polynomial, etc.).
  - But objective function must be separable.
    - Add new variables if necessary to accomplish this.

Outline

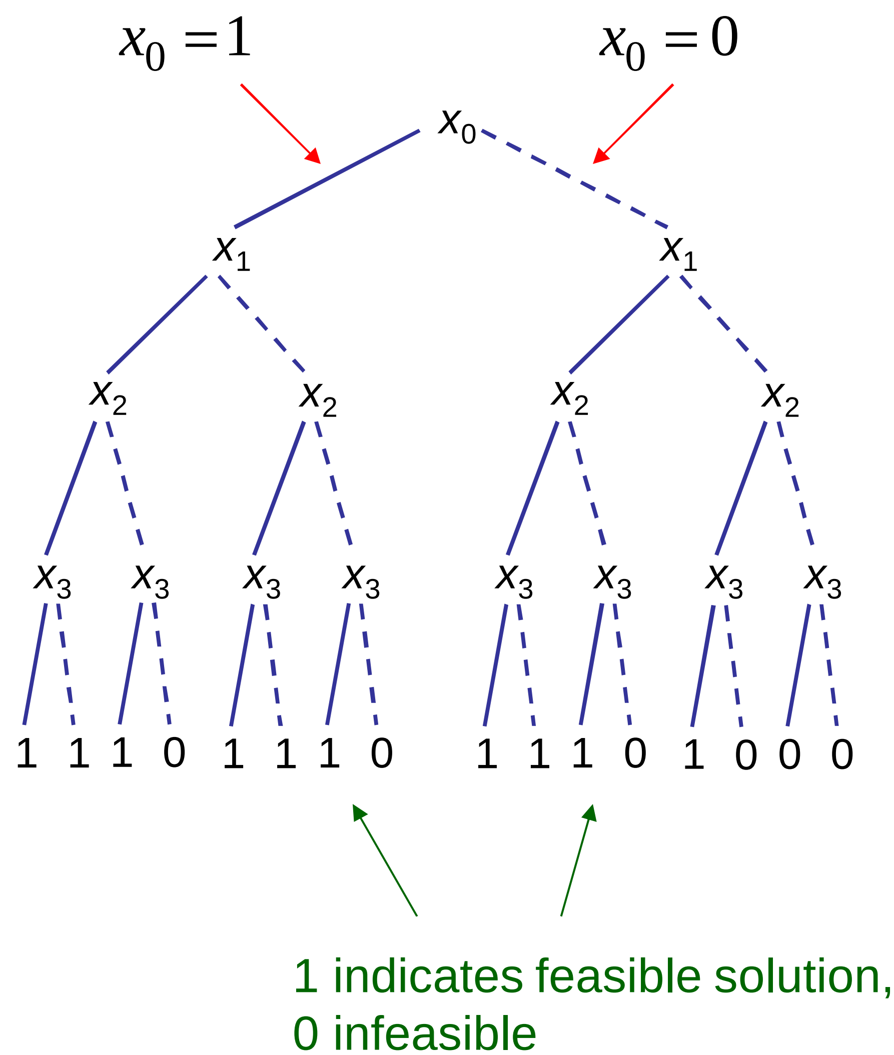
- Analysis using MDDs
- Example: capital budgeting problem
  - Complete MDD (all feasible solutions)
- Example: reliability networks
  - Complete MDD
- Cost-bounded BDDs
  - Represent near-optimal solutions only
  - Much smaller BDDs
  - Not yet implemented for MDDs
- Example: 0-1 linear programming

Analysis Using MDDs

- An MDD for a constraint set is a compact representation of the **branching tree**.
  - Superimpose isomorphic subtrees.
  - Remove unnecessary nodes.
- The constraint set has a unique **reduced** MDD.
  - For a given variable ordering.

Analysis Using MDDs

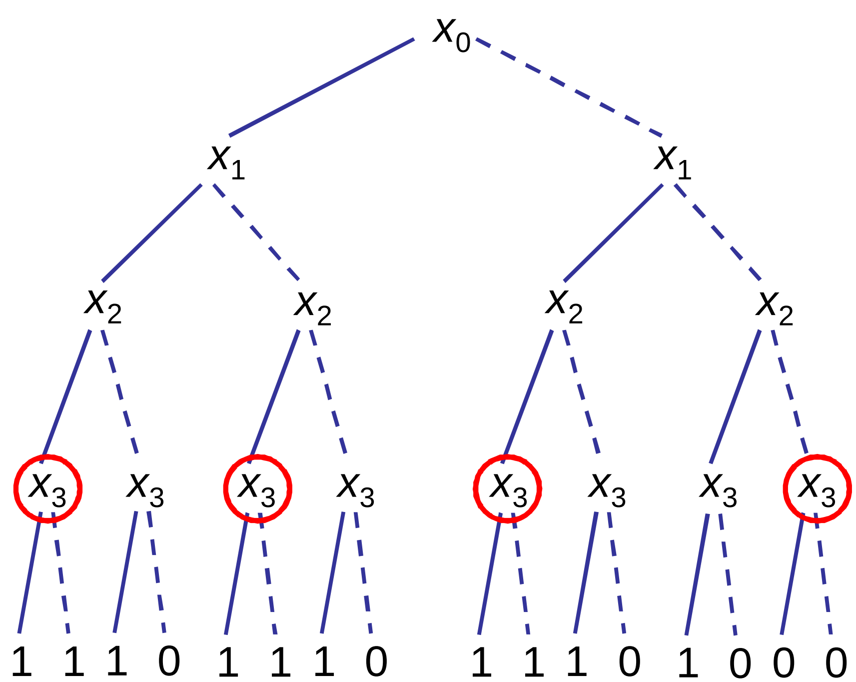
- Can find **optimal solution** by computing **shortest path** in MDD.
  - Do postoptimality analysis by analyzing MDD and its near-optimal paths.



Branching tree for 0-1 inequality

$$2x_0 + 3x_1 + 5x_2 + 5x_3 \geq 7$$

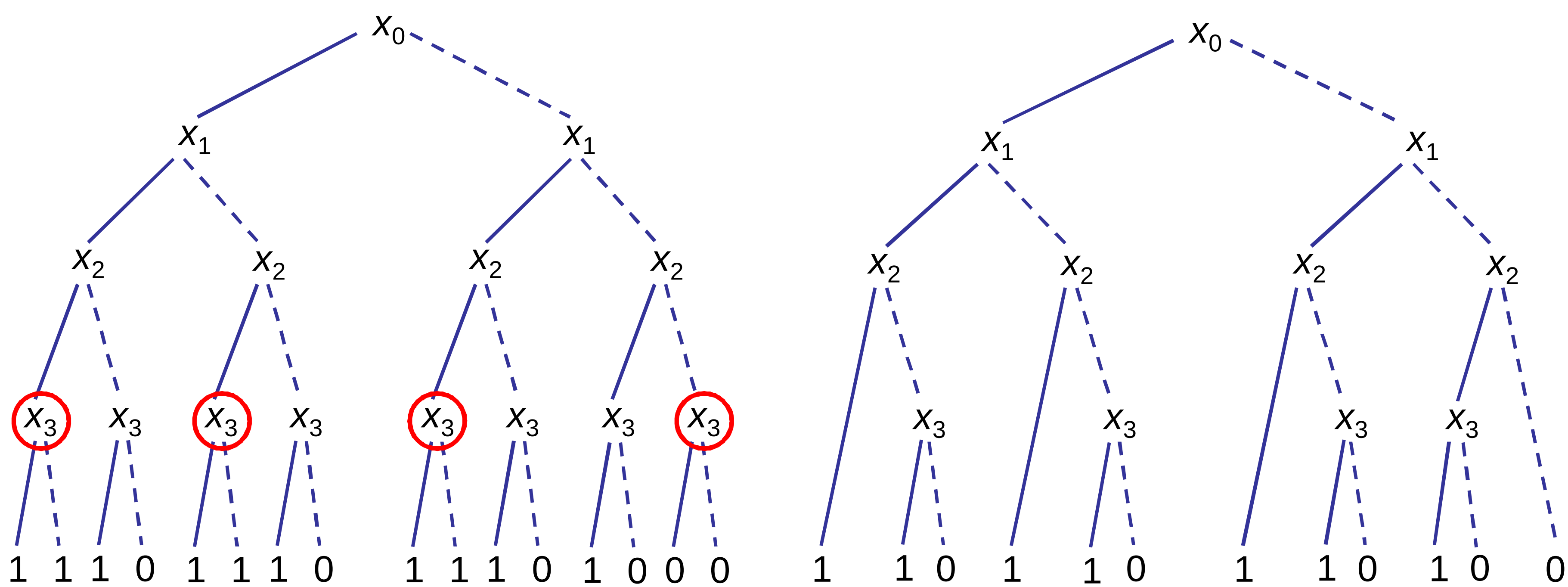
1



Branching tree for 0-1 inequality

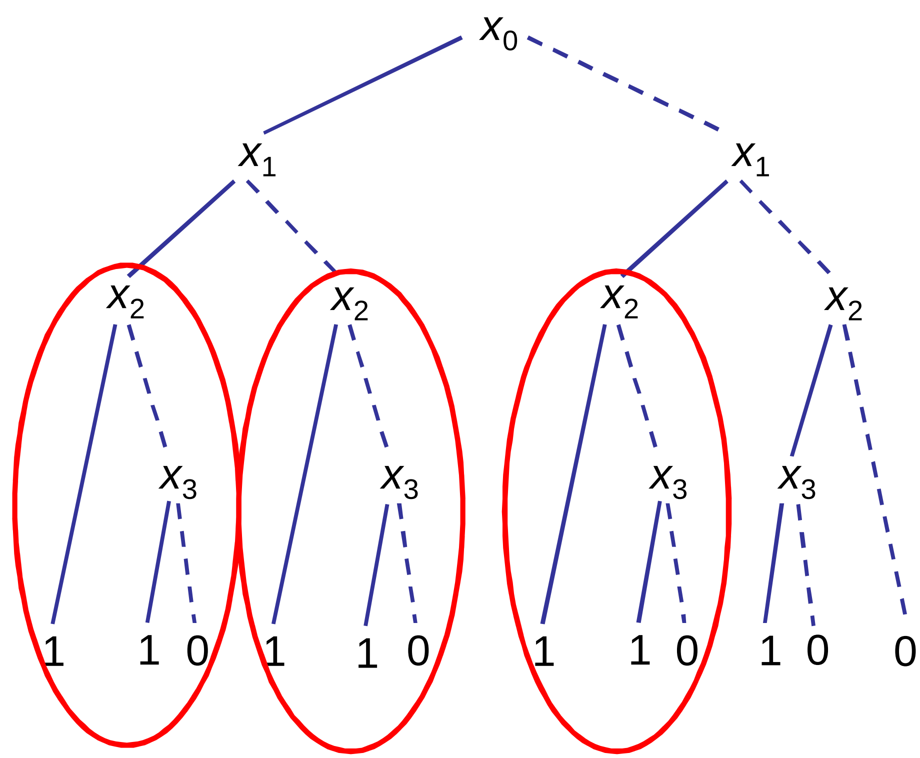
$$2x_0 + 3x_1 + 5x_2 + 5x_3 \geq 7$$

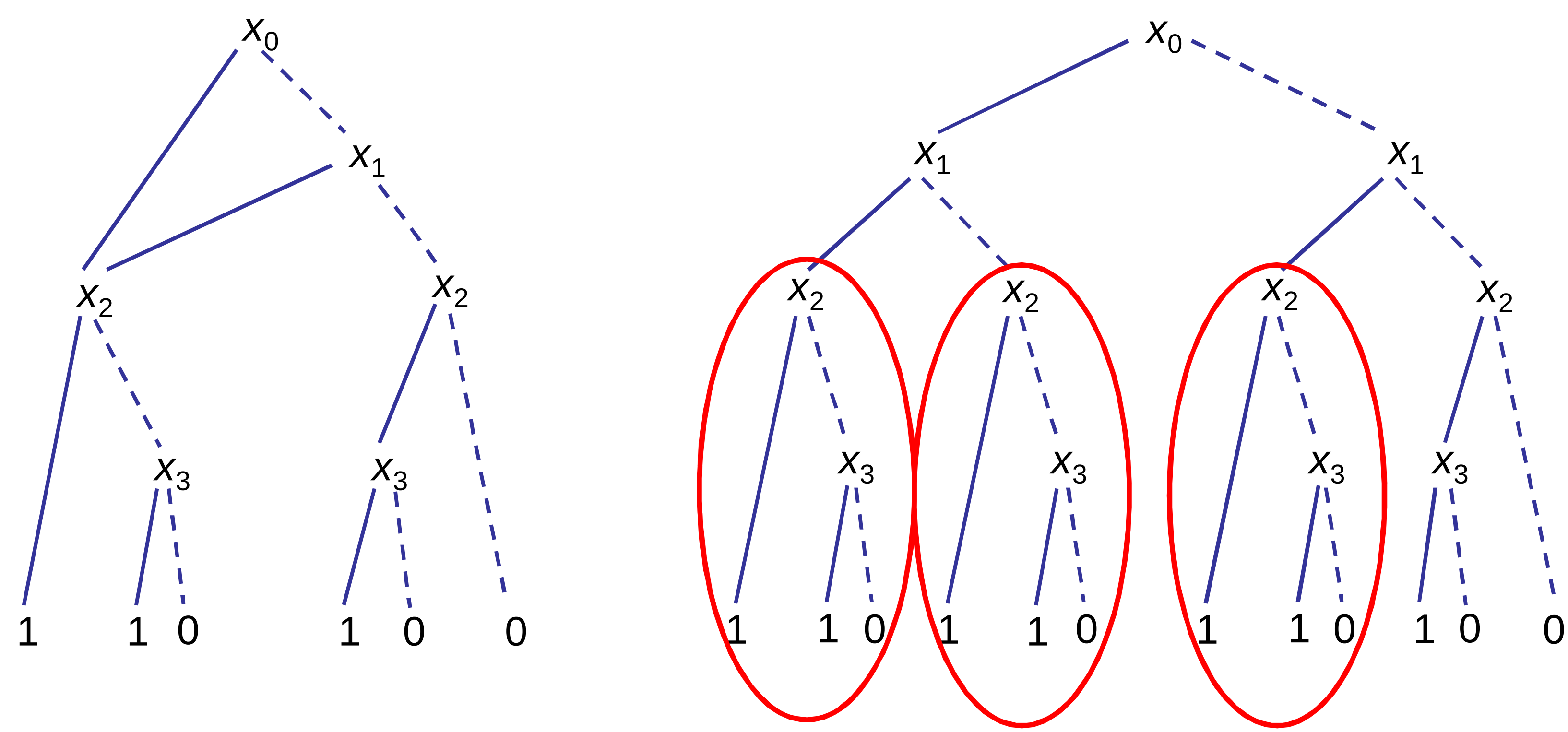
Remove redundant nodes…

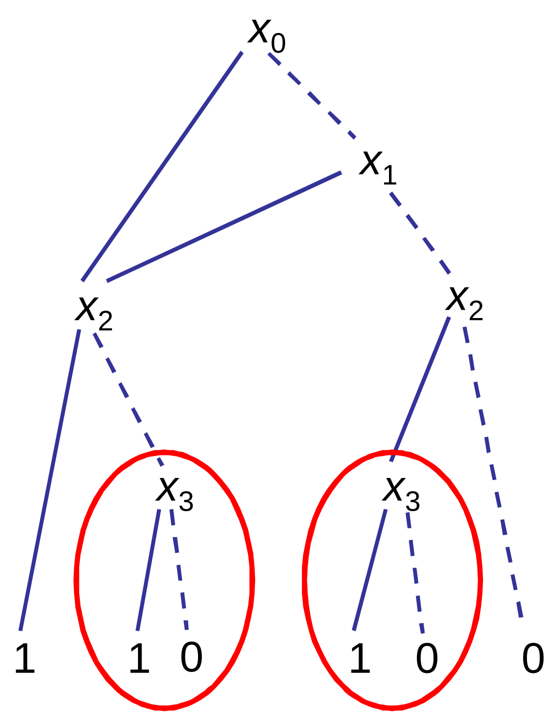


x0
x1 x1
x2 x2 x2 x2
x3 x3 x3 x3 x3 x3 x3 x3
1 1 1 0 1 1 1 0 1 1 1 0 1 0 0 0
x0
x1 x1
x2 x2 x2 x2
x3 x3 x3 x3
1 1 0 1 1 0 1 1 0 1 0 0

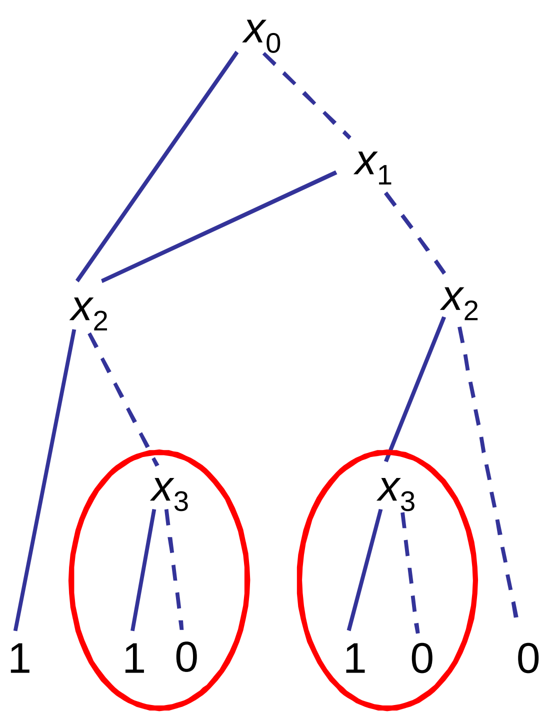
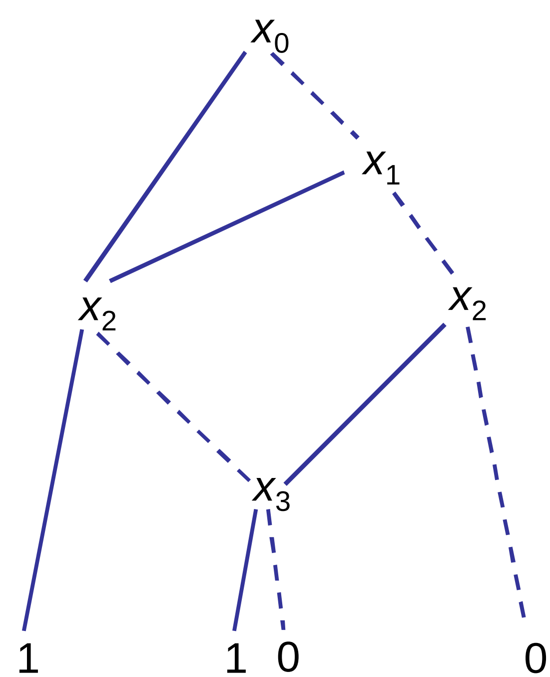
Superimpose identical subtrees…



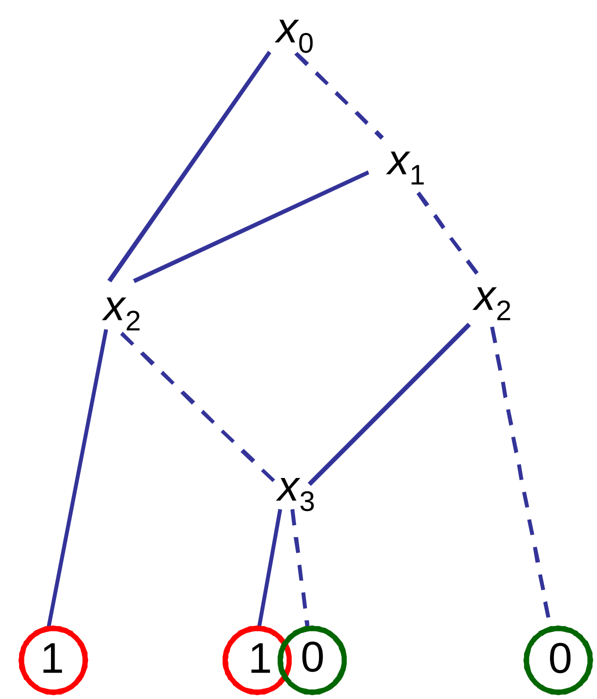


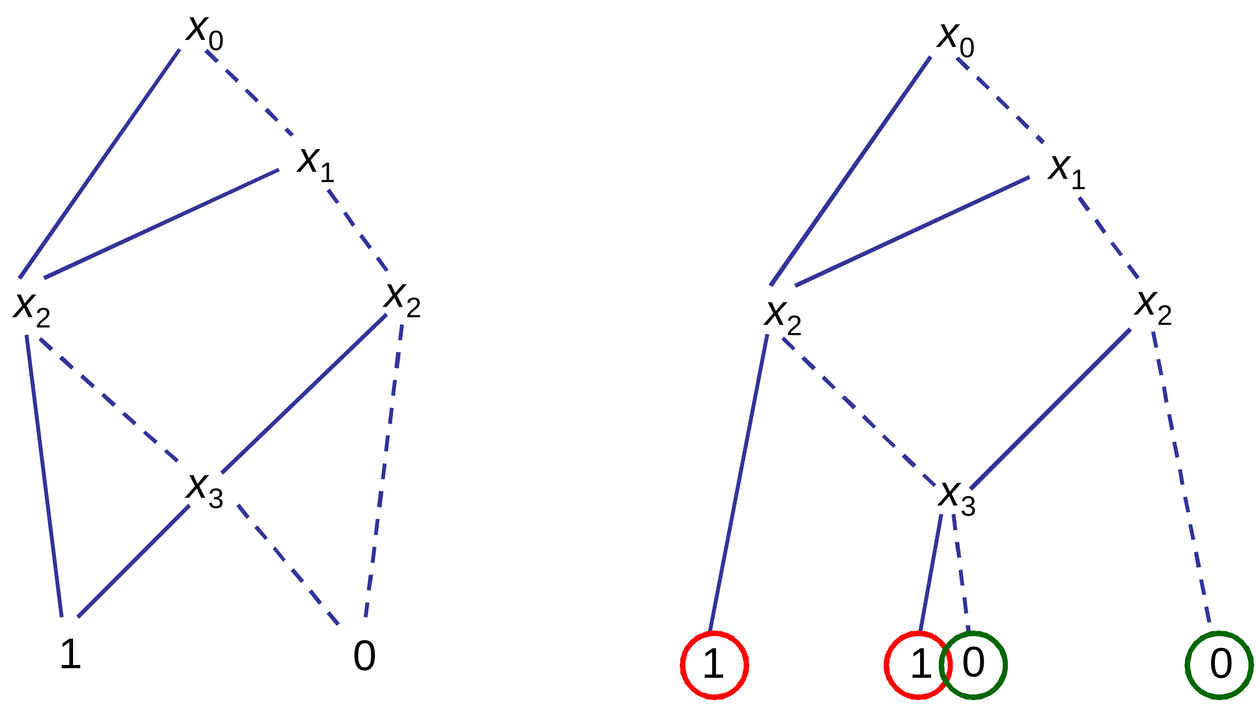


Superimpose identical subtrees…

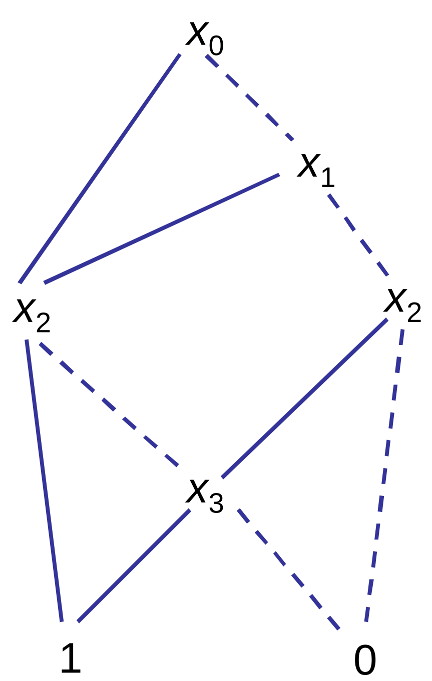

x_0
x_1
x_2
x_2
x_3
x_3
1
1
0
1
0
0

x_0
x_1
x_2
x_2
x_3
1
1
0
0

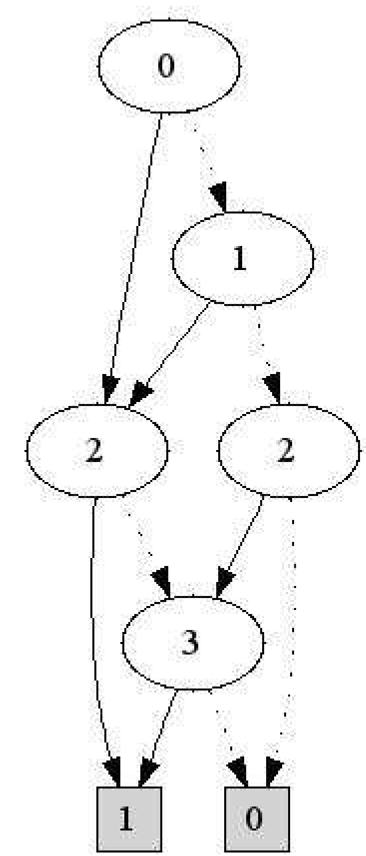
Superimpose identical leaf nodes…





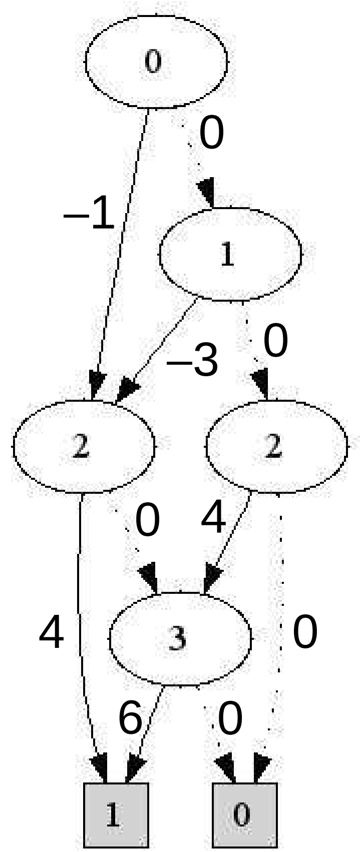
x_0
x_1
x_2
x_2
x_3
1
0
x_0
x_1
x_2
x_2
x_3
1
1
0
0





as generated by software

$\min 2x_0 - 3x_1 + 4x_2 + 6x_3$ subject to $2x_0 + 3x_1 + 5x_2 + 5x_3 \geq 7$



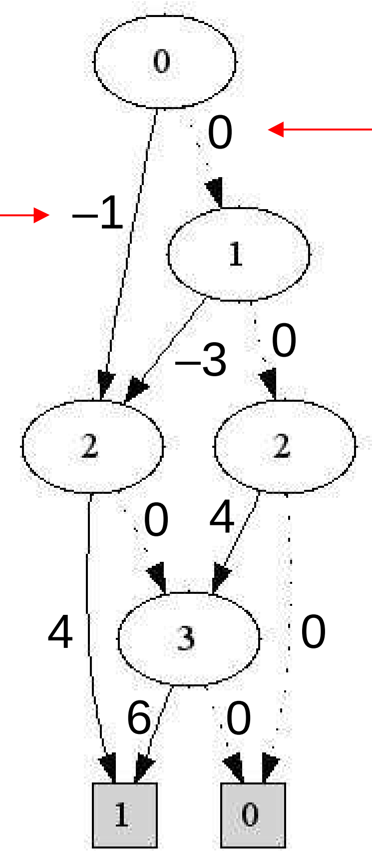
$2 \cdot 0$

Edge lengths reflect terms of objective function

$$\min 2x_0 - 3x_1 + 4x_2 + 6x_3 \quad \text{subject to} \quad 2x_0+3x_1+5x_2+5x_3 \geq 7$$

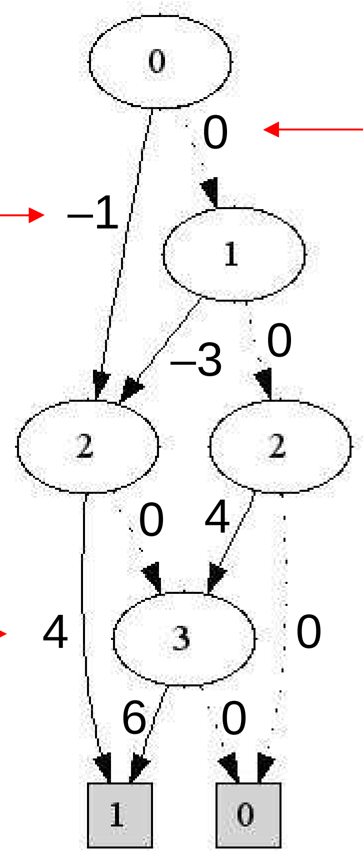
$2\cdot 1 + \min_{x_1\in\{0,1\}} \{-3x_1\}$

$2\cdot 0$



Edge lengths reflect terms of objective function

$\min 2x_0 - 3x_1 + 4x_2 + 6x_3$ subject to $2x_0+3x_1+5x_2+5x_3 \geq 7$



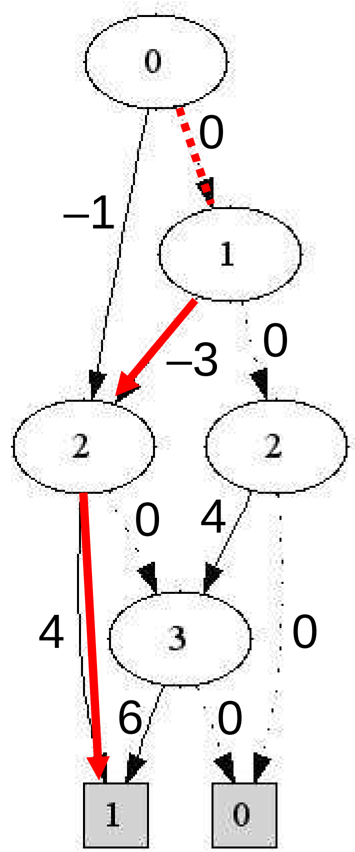
$2 \cdot 0$

$2 \cdot 1 + \min_{x_1 \in \{0,1\}} \{-3x_1\}$

$4 \cdot 1 + \min_{x_1 \in \{0,1\}} \{4x_3\}$

Edge lengths reflect terms of objective function

$$\min\ 2x_0 - 3x_1 + 4x_2 + 6x_3 \quad \text{subject to} \quad 2x_0 + 3x_1 + 5x_2 + 5x_3 \geq 7$$



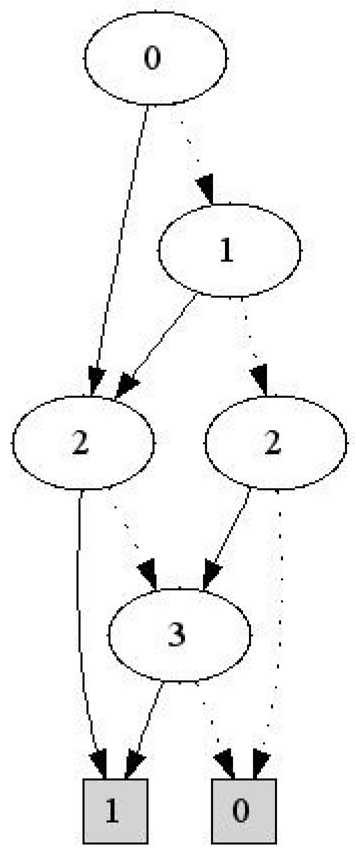
Shortest path has length 1

Optimal solution:

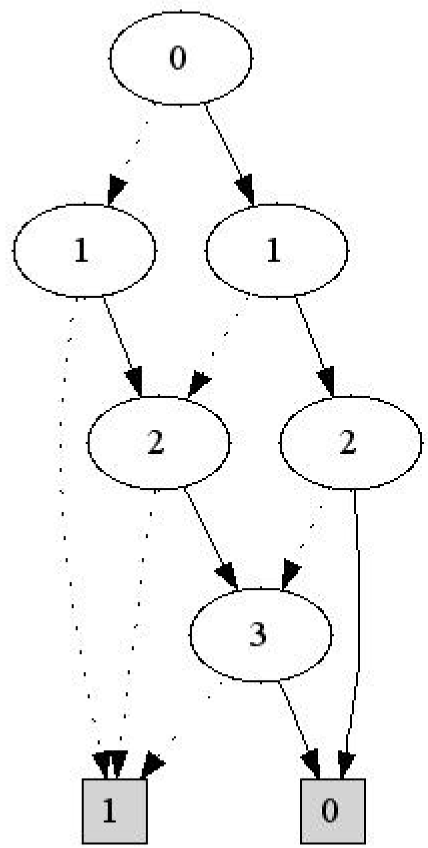
$$(x_0, x_1, x_2, x_3) = (0,1,1,0)$$

Set to minimizing value

Combine constraints by conjoining BDDs

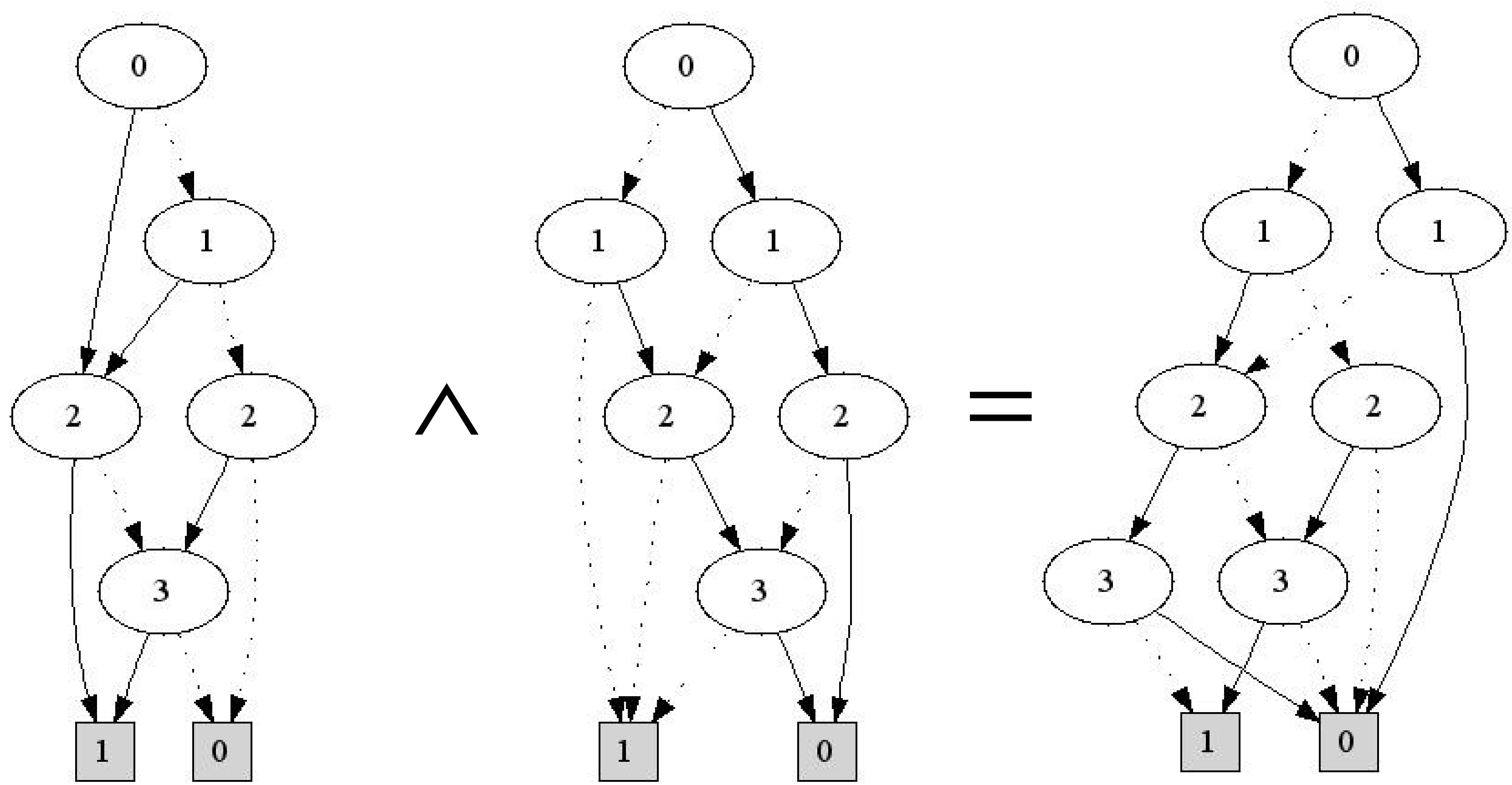


$\wedge$



$$2x_0 + 3x_1 + 5x_2 + 5x_3 \geq 7 \qquad x_0 + x_1 + x_2 + x_3 \leq 2.$$

Combine constraints by conjoining BDDs



$2x_0 + 3x_1 + 5x_2 + 5x_3 \geq 7$ $\quad$ $x_0 + x_1 + x_2 + x_3 \leq 2.$

Constructing MDDs

- Generating an MDD from scratch requires enumeration of search tree.
  - Intelligent caching to identify reduced form.
  - Known bound on optimal value can prune the search.

Constructing MDDs

- Generating an MDD from scratch requires enumeration of search tree.
  - Intelligent caching to identify reduced form.
  - Known bound on optimal value can prune the search.
- In practice, MDD for an expression is formed by combining MDDs of subexpressions.
  - For example, conjoining BDDs of individual knapsack constraints.
- We will use cost-based bounding.
  - Represent near-optimal solutions only.
  - Prune the MDD as it is constructed.

Constructing MDDs

- BDD for a knapsack constraint can be surprisingly small…

Constructing MDDs

- BDD for a knapsack constraint can be surprisingly small…

The 0-1 inequality

$$300x_0 + 300x_1 + 285x_2 + 285x_3 + 265x_4 + 265x_5 + 230x_6 + 230x_7 + 190x_8 + 200x_9 + \\ 400x_{10} + 200x_{11} + 400x_{12} + 200x_{13} + 400x_{14} + 200x_{15} + 400x_{16} + 200x_{17} + 400x_{18} \geq 2701$$

has 117,520 minimal feasible solutions

Constructing MDDs

- BDD for a knapsack constraint can be surprisingly small…

The 0-1 inequality

$$\begin{aligned}&300x_0 + 300x_1 + 285x_2 + 285x_3 + 265x_4 + 265x_5 + 230x_6 + 230x_7 + 190x_8 + 200x_9 + \\ &\quad 400x_{10} + 200x_{11} + 400x_{12} + 200x_{13} + 400x_{14} + 200x_{15} + 400x_{16} + 200x_{17} + 400x_{18} \le 2700\end{aligned}$$

has 117,520 minimal feasible solutions

Or equivalently,

$$\begin{aligned}&300x_0 + 300x_1 + 285x_2 + 285x_3 + 265x_4 + 265x_5 + 230x_6 + 230x_7 + 190x_8 + 200x_9 + \\ &\quad 400x_{10} + 200x_{11} + 400x_{12} + 200x_{13} + 400x_{14} + 200x_{15} + 400x_{16} + 200x_{17} + 400x_{18} \ge 2701\end{aligned}$$

has 117,520 minimal covers

Constructing MDDs

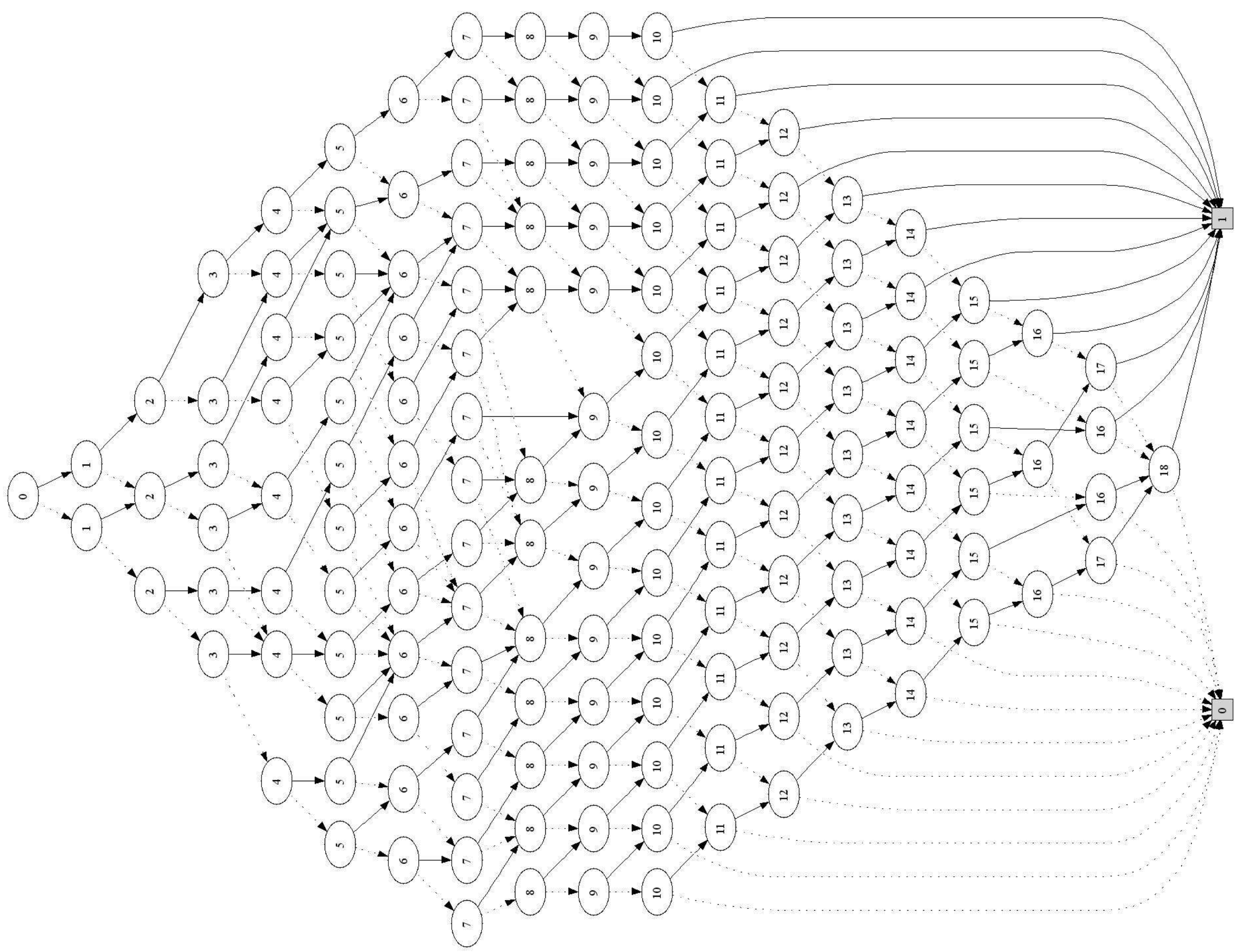
- BDD for a knapsack constraint can be surprisingly small…

The 0-1 inequality

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

has 117,520 minimal feasible solutions

But its reduced BDD has only 152 nodes…



Capital Budgeting

- Capital budgeting problem:

$$\max cx$$
$$ax \le b$$
$$x_j \in \{0,1,2,3\}$$

where

$c_j =$ return from facility j

$a_j =$ cost of facility j, $b =$ budget

$x_j =$ number of facilities of type j

Capital Budgeting

- Capital budgeting problem:

$$\max cx$$
$$ax \le b$$
$$x_j \in \{0,1,2,3\}$$

where

$c_j =$ return from facility j

$a_j =$ cost of facility j, $b =$ budget

$x_j =$ number of facilities of type j

Let

$c = (503, 493, 450, 410, 395, 389, 360, 331, 304, 299)$

$a = (249, 241, 219, 211, 194, 196, 177, 162, 150, 149)$

$b = 1800$

Capital Budgeting

- BDD has 1080 nodes.
  - MDD is converted to BDD.

- Cost-based domain analysis:
  - Let $Sol(\Delta) = \{x \mid cx \geq c_{opt} - \Delta\}$ be set of solutions within Δ of the optimal value c_{opt}.

    Let $x_j(\Delta)$ be projection of $Sol(\Delta)$ onto x_j
  - Observe how $x_j(\Delta)$ grows as Δ increases.
    - » Gives an idea of how much freedom there is to adjust the solution without much sacrifice.

$x_1(0)$

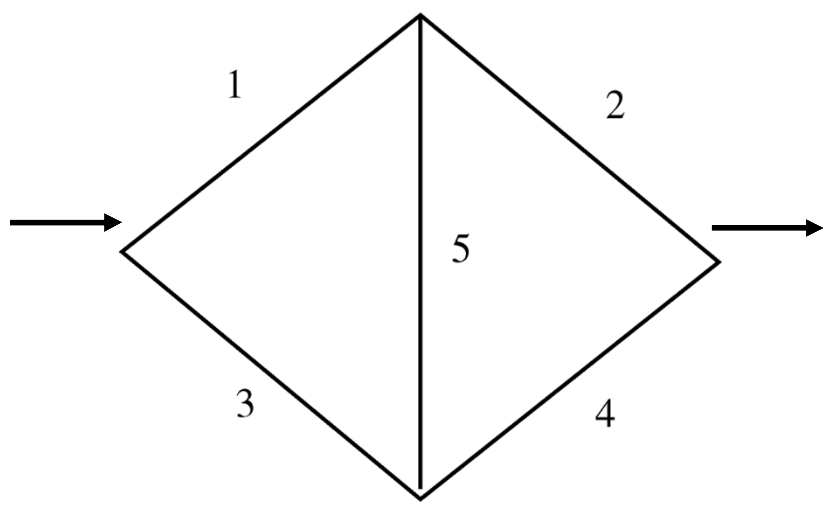
$c_{opt} - \Delta$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
3678:	0	3	2	0	0	0	1	1	2	0
3677:			1		1			3	0	
3676:							0		1	
3673:	1,2	0,1,2	3		3		2,3	2	3	1
3672:					2			0		
3669:										2
3668:			0							
3666:						1				
3664:										3
3658:				1						
3657:	3					2				
3646:						3				
3633:				2						
3616:				3						

$x_1(0)$ $x_1(5) = \{0,1,2\}$

$c_{opt} - \Delta$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
3678:	0	3	2	0	0	0	1	1	2	0
3677:			1		1			3	0	
3676:							0		1	
3673:	1,2	0,1,2	3		3		2,3	2	3	1
3672:					2			0		
3669:										2
3668:			0							
3666:						1				
3664:										3
3658:				1						
3657:	3					2				
3646:						3				
3633:				2						
3616:				3						

Network Reliability

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



$$R = R_1R_2 + (1-R_2)R_3R_4 + (1-R_1)R_2R_3R_4 + R_1(1-R_2)(1-R_3)R_4R_5 + (1-R_1)R_2R_3(1-R_4)R_5$$

The problem:

$$\min \sum_j c_j x_j$$

Number of links j

$$R \geq R_{\min}$$

$$R = R_1R_2 + (1-R_2)R_3R_4 + (1-R_1)R_2R_3R_4 + R_1(1-R_2)(1-R_3)R_4R_5 + (1-R_1)R_2R_3(1-R_4)R_5$$

$$R_j = 1-(1-r_j)^{x_j}, \quad \text{all } j$$

$$x_j \in \{0,1,2,3\}$$

Set $R_{\min}$ = 60 in all examples

$r = (0.9, 0.85, 0.8, 0.9, 0.95)$

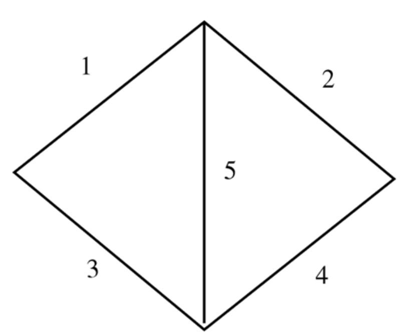
$c = (25, 35, 40, 10, 60)$

Cost-based domain analysis

$c_{opt} + \Delta$	x_1	x_2	x_3	x_4	x_5	R
50:	0	0	1	1	0	72
60:	1	1	0	0,2		79
85:	2					84
90:			2	3		86
95:		2			1	88
100:						95
120:						97
125:	3					
155:		3			2	
160:						98
170:						99
180:			3			
230:					3	

89 nodes in MDD

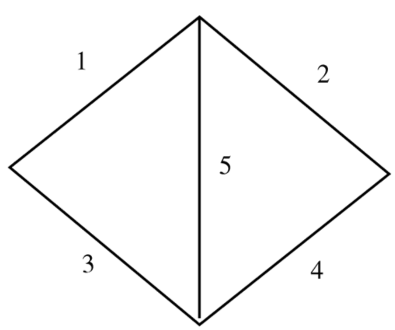
1.2 seconds to compile MDD



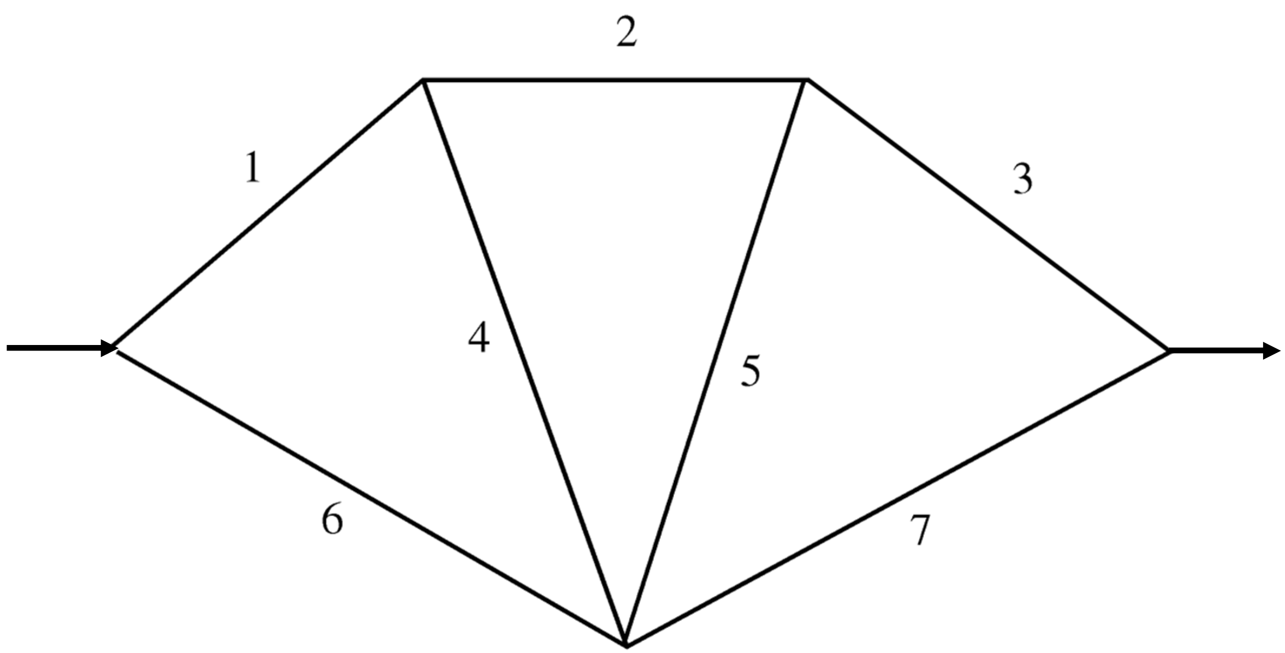
Domain analysis with respect to *R*

R	x_1	x_2	x_3	x_4	x_5
99:	1,2	1,2	1,2	1,2	0,1,2,3
98:		0,3	0,3		
97:				0,3	
95:	0,3				

Same MDD as before



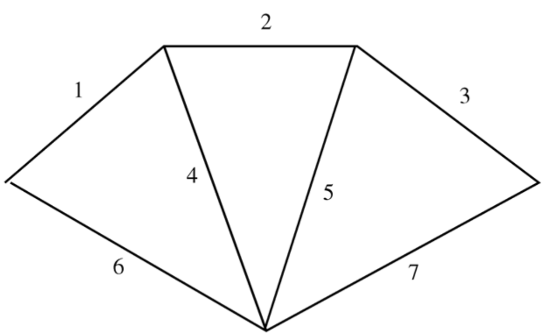
7 bridges



Cost-based
domain analysis

3126 nodes in MDD

9.0 seconds
to compile MDD

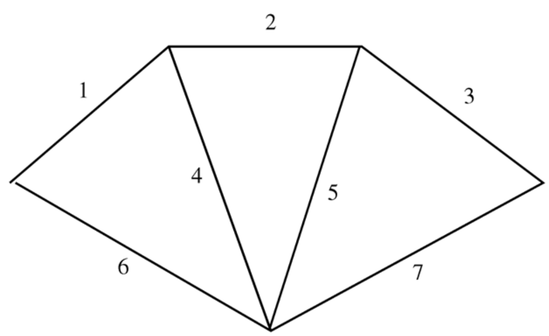


$c_{opt} + \Delta$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	R
9:	0	0	0	0	0	1	1	72.2
11:			1		1		0	
12:	1			1		0		
13:		1				2		82.9
14:					2		2	
15:			2	2				
16:	2							
17:					3	3		84.6
18:		2		3				95.2
19:			3				3	
20:	3							
22:								97.2
23:		3						
27:								99.2
34:								99.4
40:								99.6
43:								99.7
47:								99.8
54:								99.9

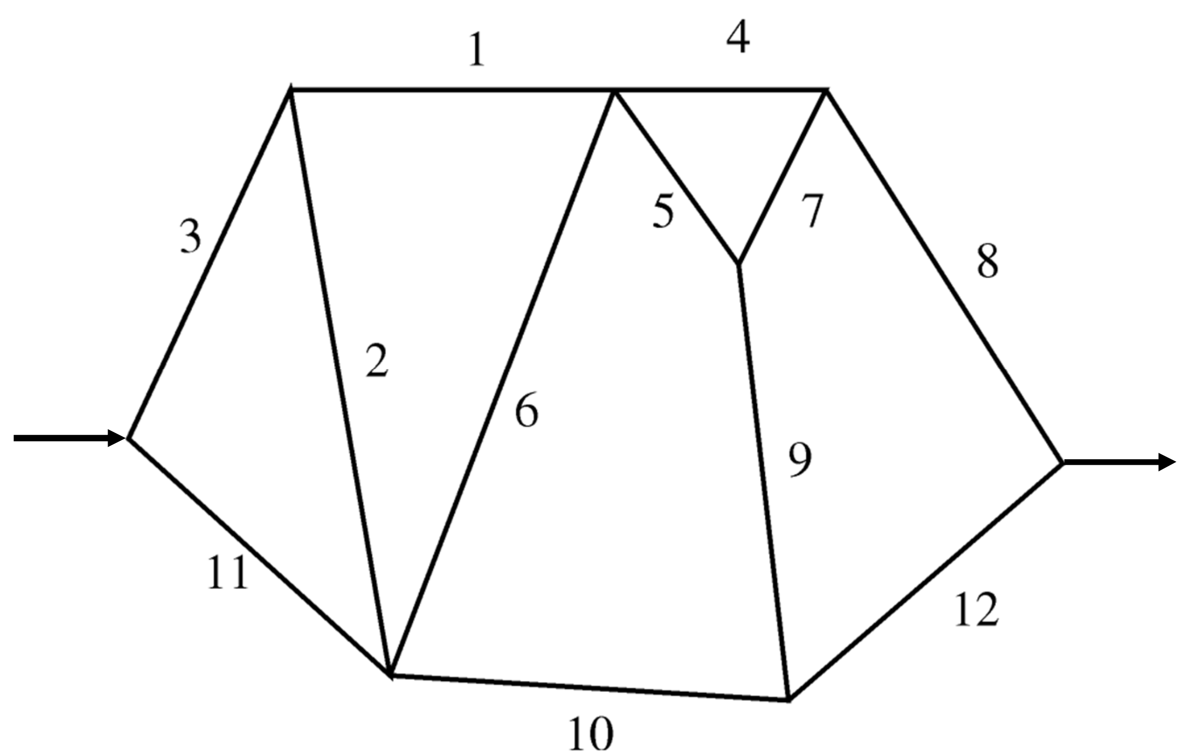
Domain analysis with respect to *R*

R	x_1	x_2	x_3	x_4	x_5	x_6	x_7
99.9	2,3	0,1,2,3	1,2,3	0,1,2,3	0,1,2,3	2,3	2,3
99.8	1						1
99.5	0		0			1	
99.1							0
97.2						0	

Same BDD as before



12 bridges

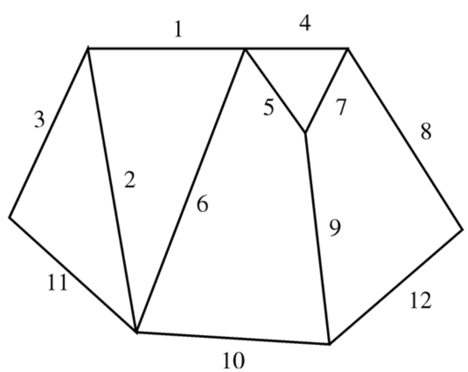


Cost-based domain analysis

36,301 nodes in MDD

1980 seconds to compile MDD

1.8 seconds query time to build this table

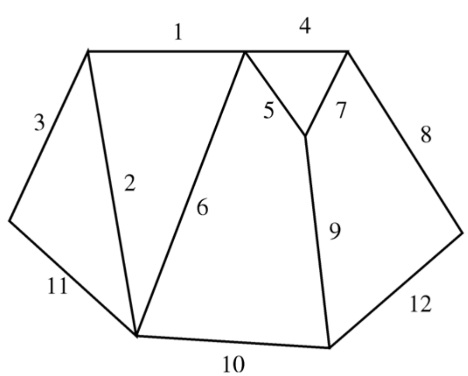


$c_{opt}+\Delta$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	R
180	1	0	2	3	0	0	0	2	0	0	0	0	80
185			3	2									82
190								3					
195													83
200										1			
205													86
210	2						1						
215													88
220										2			
225					1				1				
230	0		0			1,2					1		
235			1										
240		1		1			2			3	2	1	
250				0				0,1				2	
255													91
260	3								2				
265													93
270					2	3	3						
290											3		
300		2											
305									3				
310												3	
315					3								94
340													95
360		3											
365													96
380													97
430													98
485													99

Domain analysis with respect to R

R	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
99:	0..3	0..3	1..3	0..3	0..3	0..3	0..3	1..3	0..3	0..3	1..3	1..3
98:			0					0				0
96:											0	

Same MDD as before



Reducing MDD Growth

- **Original** MDD
  - Represents entire feasible set
  - Can be very large.

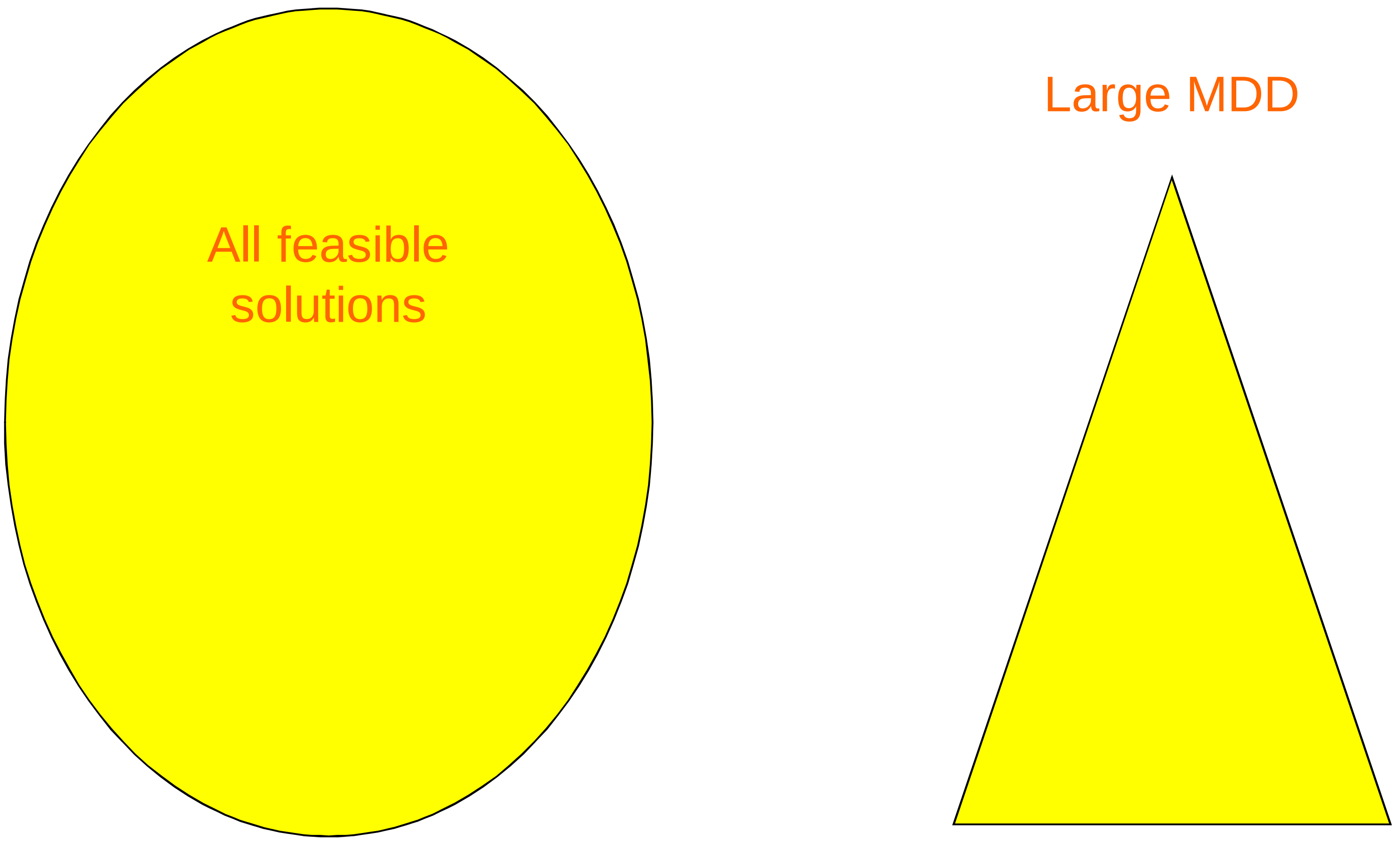
Reducing MDD Growth

- **Original** MDD
  - Represents entire feasible set
  - Can be very large.
- **Near-optimal** MDD
  - Exactly represents solutions with Δ_{max} of optimum.
  - Can be even larger than the original MDD!

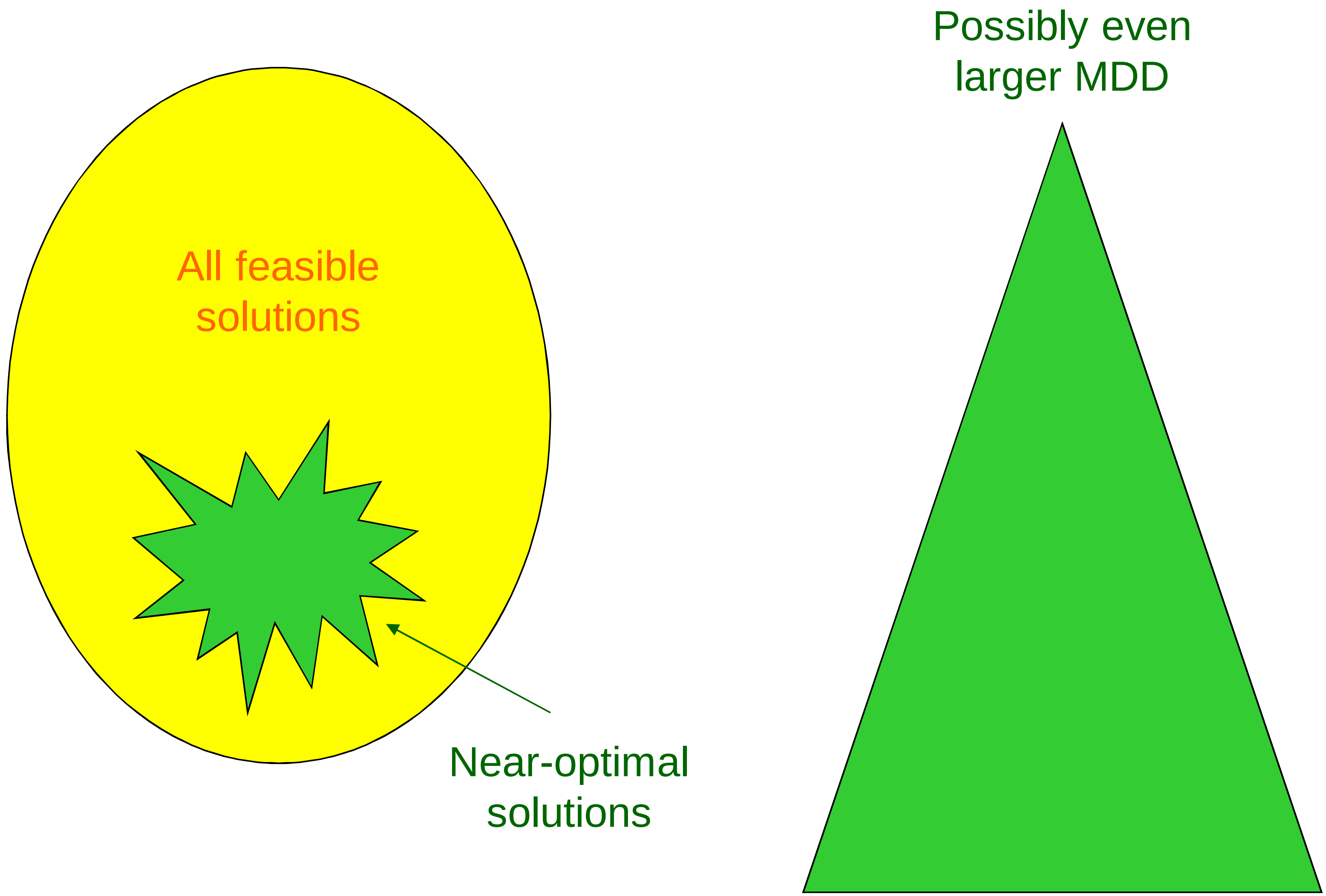
Reducing MDD Growth

- **Original** MDD
  - Represents entire feasible set
  - Can be very large.
- **Near-optimal** MDD
  - Exactly represents solutions with Δ_{max} of optimum.
  - Can be even larger than the original MDD!
- We introduce a **sound** MDD.
  - Includes all near-optimal solutions, plus some others.
  - Much smaller than near-optimal MDD.
  - Constructed by pruning and contraction.

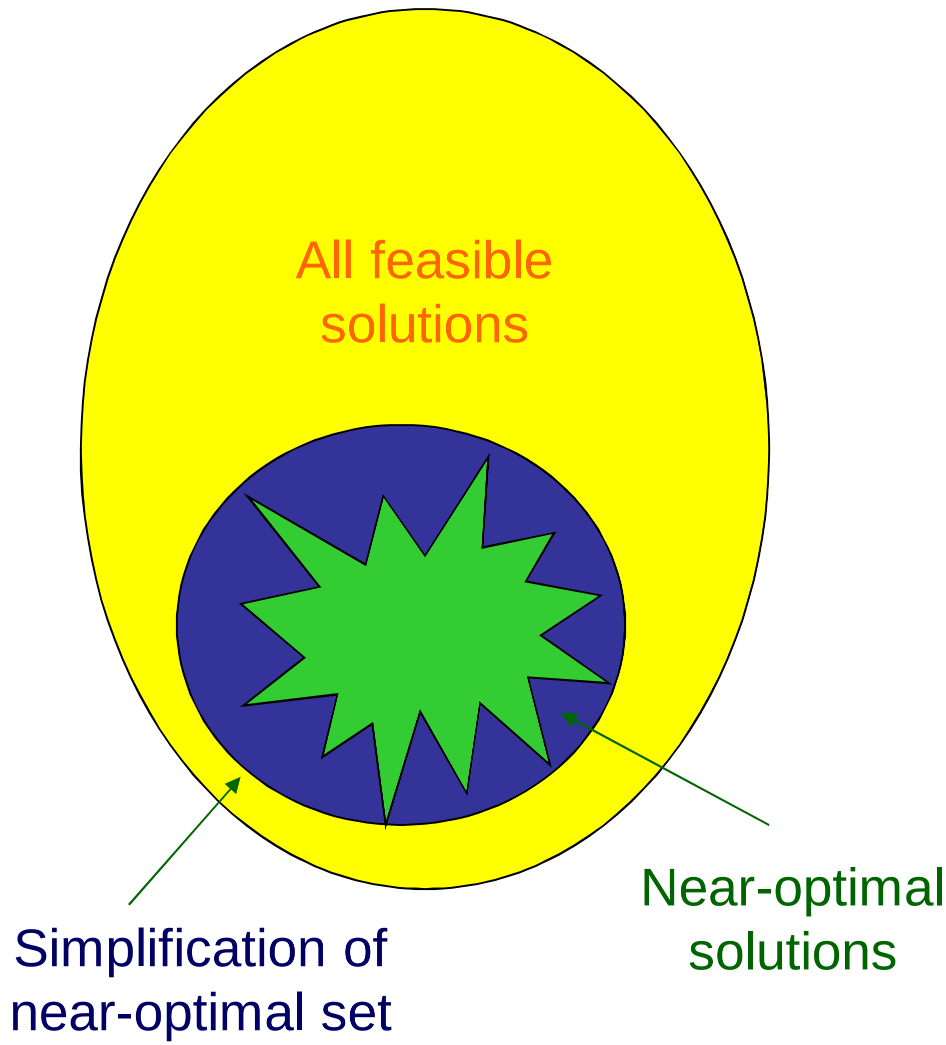
Reducing MDD Growth

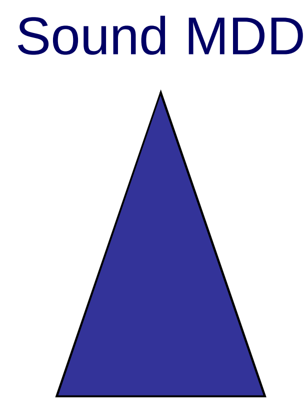


Reducing MDD Growth



Reducing MDD Growth



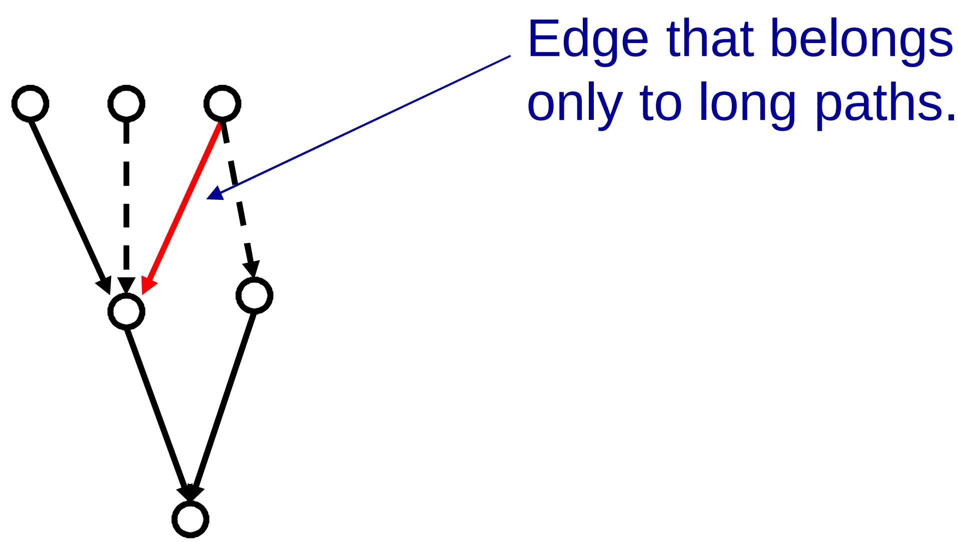


Pruning and Contracting

- We are unaware of a polytime exact method for constructing the **smallest sound** MDD.
- We use two heuristic methods for generating **small sound** MDDs during compilation:
  - Pruning edges
  - Contracting nodes

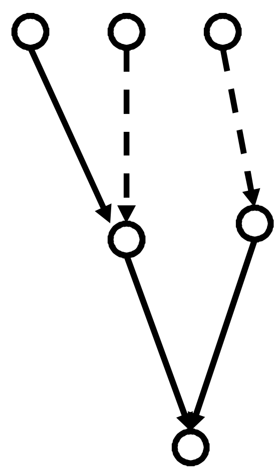
Pruning

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



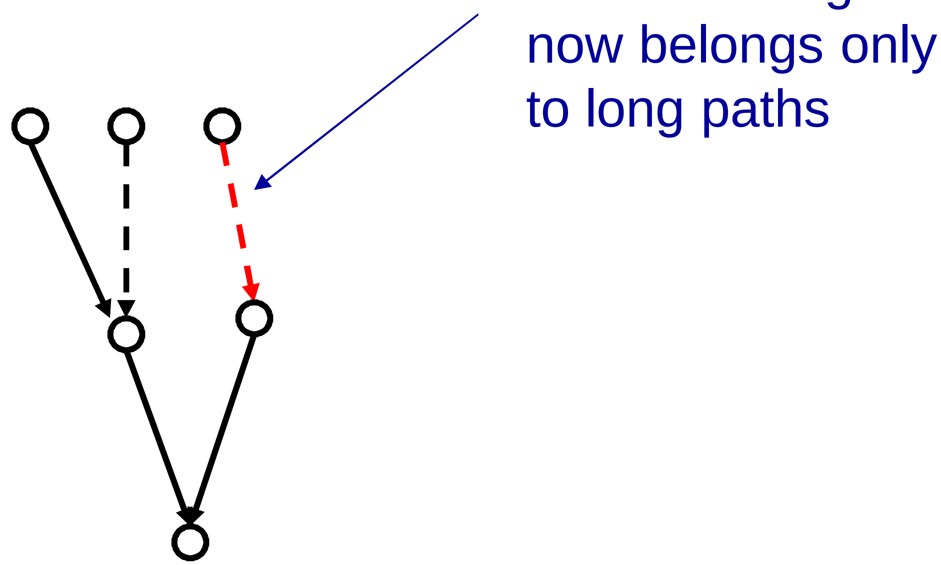
Pruning

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



Pruning

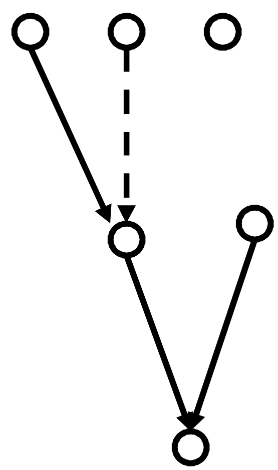
- Delete all edges that belong only to paths longer than $c_{opt} + \Delta_{max}$.



Pruning

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

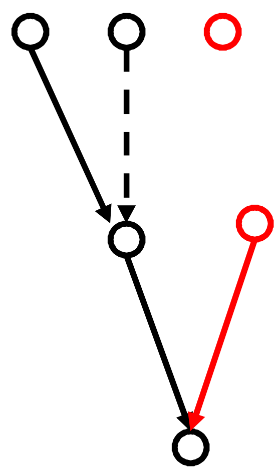
Delete it, too.



Pruning

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

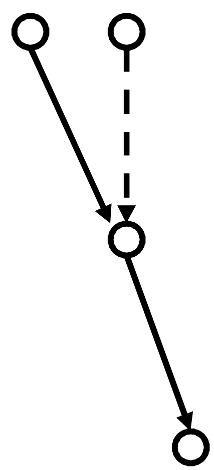
And simplify the BDD.



Pruning

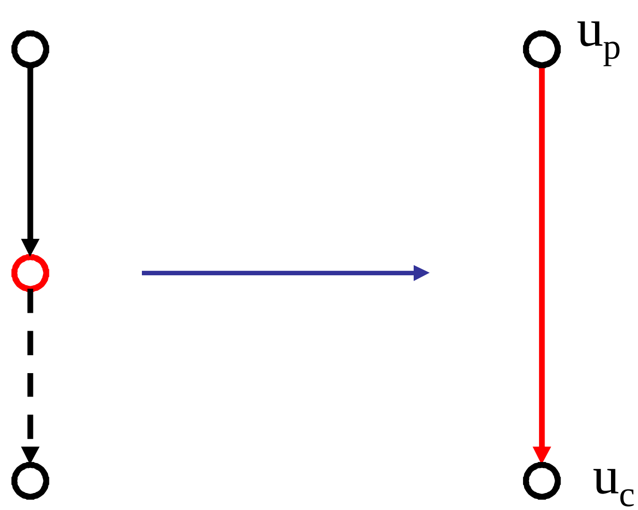
- Delete all edges that belong only to paths longer than $c_{opt} + \Delta_{max}$.

And simplify the BDD.



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 $[0,r]$

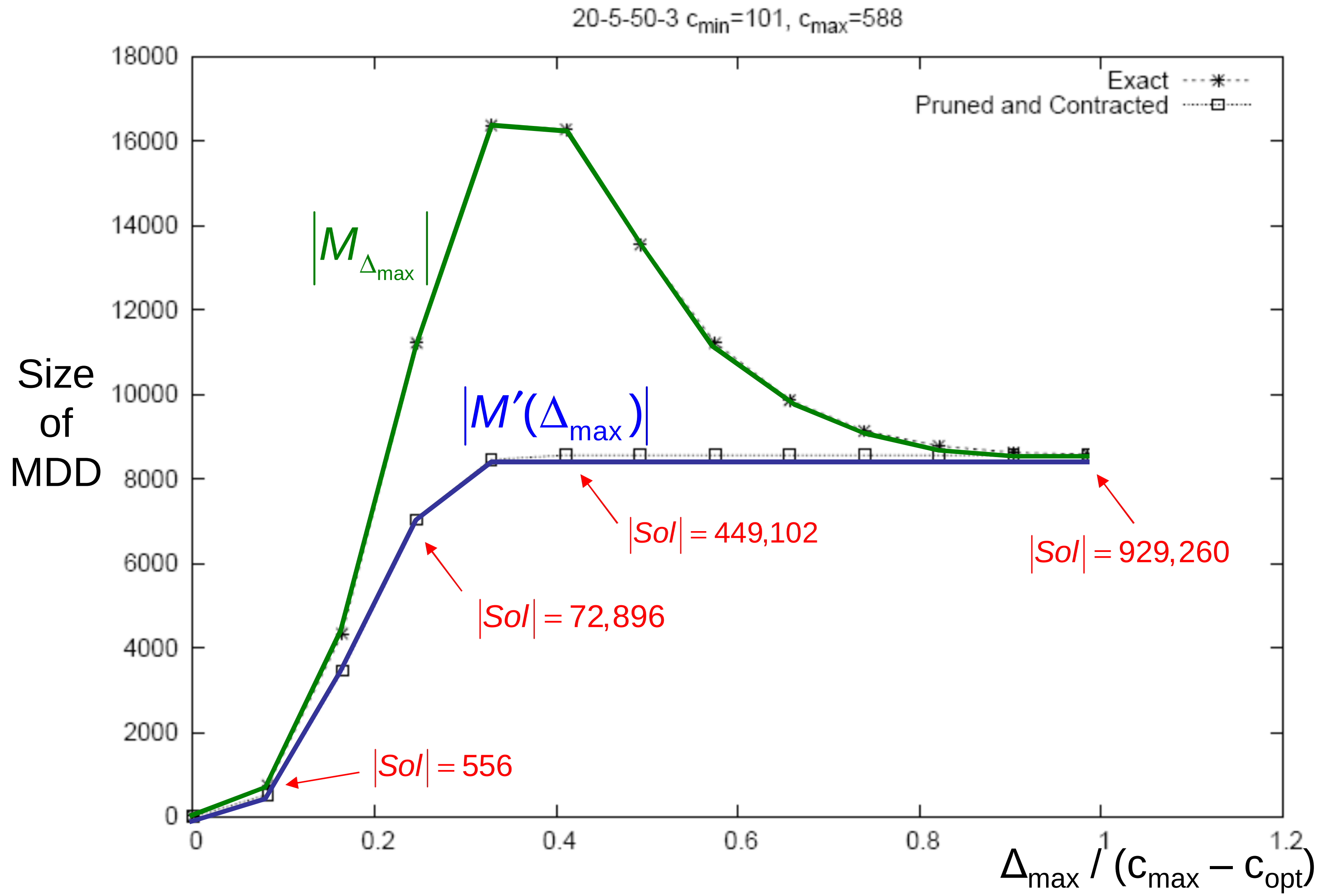
Experimental Results

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

Original MDD → $|M|$; Near-optimal MDD → $|M_{\Delta max}|$; Sound BDD → $|M'(\Delta_{max})|$

Δ_{max}	$\|M\|$	$\|M_{\Delta max}\|$	$\|M'(\Delta_{max})\|$
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

Memory = 0.2 MB
Query time = 0.04 sec



20-5-50-3 c_{min}=101, c_{max}=588
Size of MDD
Exact
Pruned and Contracted
$|M_{\Delta_{max}}|$
$|M'(\Delta_{max})|$
$|Sol| = 556$
$|Sol| = 72,896$
$|Sol| = 449,102$
$|Sol| = 929,260$
$\Delta_{max} / (c_{max} - c_{opt})$

Experimental Results

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

Δ_{max}	$\|M\|$	$\|M_{\Delta max}\|$	$\|M'(\Delta_{max})\|$
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

Memory = 20 MB
Query time = 4 sec

Experimental Results

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

Δ_{max}	$\lvert M \rvert$	$\lvert M_{\Delta max} \rvert$	$\lvert M'(\Delta_{max}) \rvert$
0	?	40	12
15		1,143	402
35		3,003	1,160
70		11,040	7,327
100		404,713	223,008
140		?	52,123

Memory = 4 MB
Query time = 1 sec

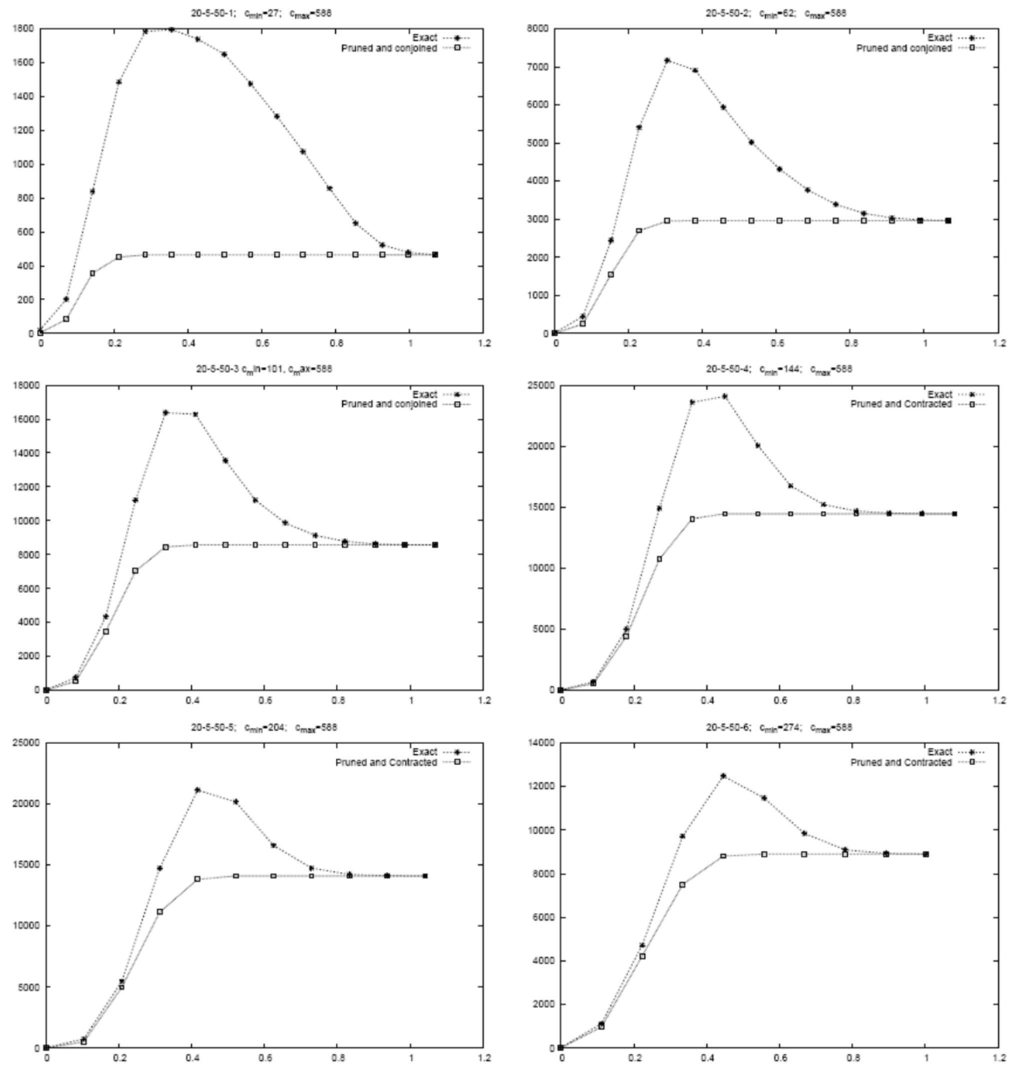
Experimental Results

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

Δ_{max}	$\|M\|$	$\|M_{\Delta max}\|$	$\|M'(\Delta_{max})\|$
0	?	60	7
50		5,519	1,814
100		111,401	78,023

Memory = 2 MB
Query time = 0.3 sec

Experimental Results



Results when tightness α is gradually reduced

Experimental Results

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

Instance	$\lvert M \rvert$	$\lvert M_{\Delta max} \rvert$	$\lvert M'(\Delta_{max}) \rvert$
lseu	?	99	19
p0033	375	41	21
p0201	310,420	737	84
stein27	25,202	6,260	4,882
stein45	5,102,257	1,765	1,176

Conclusion

- Cost-bounded BDDs provide reasonable scalabililty for BDD-based postoptimality analysis.
  - At least for 0-1 linear programming.
- Comprehensive postoptimality analysis available in real time.
  - Once MDD has been constructed.

Conclusions and Future Work

Future work:

- Extension of cost bounding to MDDs.
  - A matter of implementation.
  - Use cost bounded MDDs for nonlinear, nonconvex problems
- Test cost bounding on nonlinear problems.
  - Nonlinearity, nonconvexity should not be a major factor.
- Extension to mixed discrete/continuous variables.
  - ??

Future Research

- Deeper analysis of near-optimal set.
  - Characterize optimal and near-optimal solutions by decomposition properties, etc.
  - Existential quantification/projection methods to focus attention on important variables.
  - MDD as basis for explanation (generalized duality).

Other Uses for BDDs

- Polyhedral relaxations of MDDs
  - Apply to subsets of constraints.
- MDDs as solution technique – Shortest path approach
  - Use known bound to reduce MDD.

Other Uses for BDDs

- MDDs as solution technique – propagation thru MDD relaxation
  - Replace domain store with MDD store.
  - Use general splitting technique to generate relaxed MDD.
    - Tightness of relaxation depends on MDD width.
  - Order-of-magnitude speedups for multiple alldiffs.
  - Also applied to separable equality constraints.
  - Poor performance on 0-1 inequalities.