Discrete Global Optimization with Binary Decision Diagrams

Tarik Hadzic
IT University of Copenhagen
J. N. Hooker
Carnegie Mellon University

December 2006

Optimization with BDDs

- Goal: Use **binary decision diagrams** (**BDD**s) to solve discrete global optimization problems.
- Focus on **postoptimality analysis** as well as solution.
- Why?
  - Postoptimality analysis can be valuable in practice.
  - BDDs provide it for free.

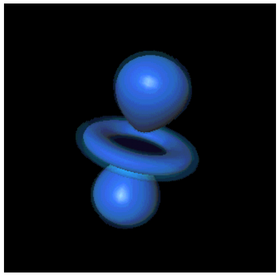
Why Postoptimality Analysis?

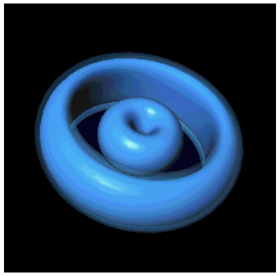
- A single optimal solution offers limited information.
- Postoptimality analysis can provide insight into model.
  - It takes a model seriously as a *model*.

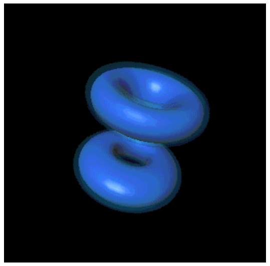
Why Postoptimality Analysis?

- 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

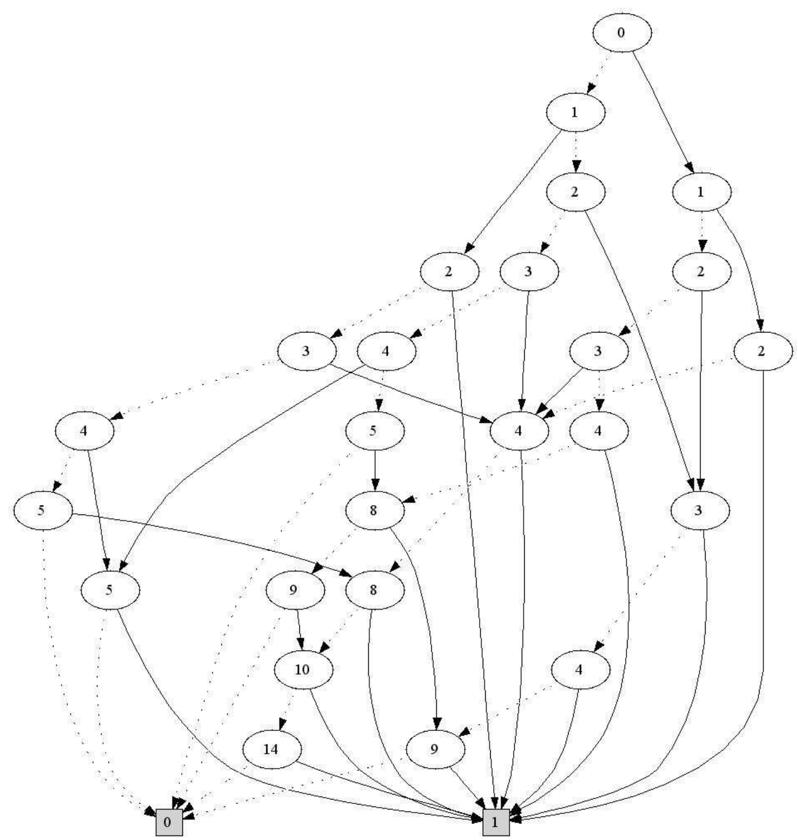
- Postoptimality analysis tends to be computationally intractable for discrete and global optimization.
  - Particularly real-time or interactive analysis.

Postoptimality Analysis

- Postoptimality analysis tends to be computationally intractable for discrete and global optimization.
  - Particularly real-time or interactive analysis.
- Ideal: a compact representation of the set of optimal, near-optimal, or feasible solutions.
  - Can be queried in real time.

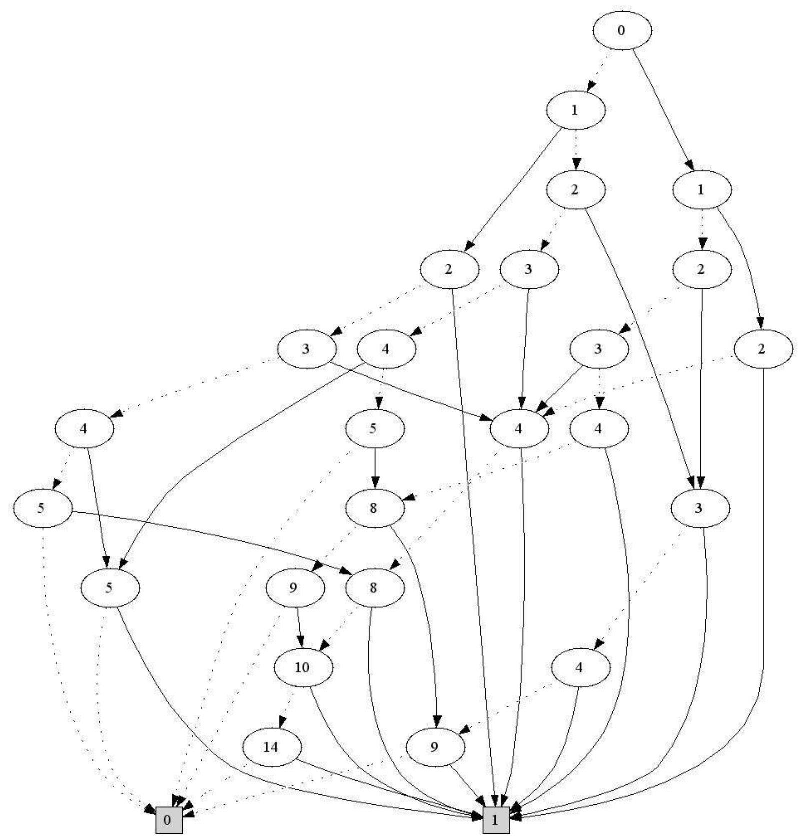
Analysis Using BBDs

- Proposal: Use **binary decision diagrams** (BDDs)
  - A BDD represents any boolean function.



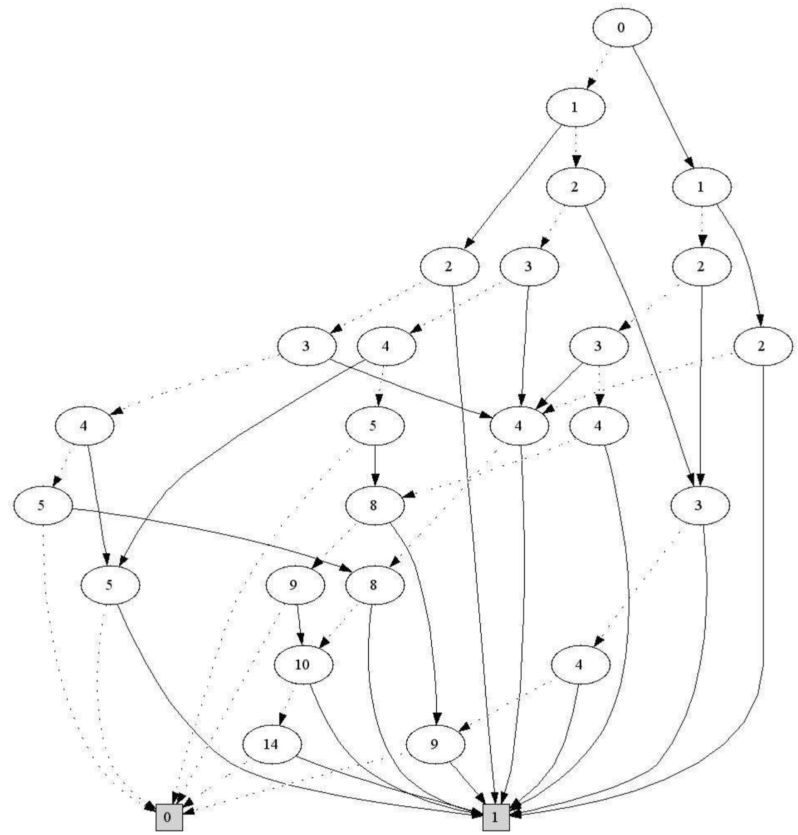
Analysis Using BBDs

- Proposal: Use **binary decision diagrams** (BDDs)
  - A BDD represents any boolean function.
- A boolean function has a **unique** *reduced* BDD.
  - …for a given variable ordering.



Analysis Using BBDs

- Proposal: Use **binary decision diagrams** (BDDs)
  - A BDD represents any boolean function.
- A boolean function has a **unique** *reduced* BDD.
  - …for a given variable ordering.
- Common applications:
  - VLSI verification & model checking
  - Configuration problems



Analysis Using BBDs

- A BDD can represent the feasible set of an optimization problem.
  - View constraint set as a boolean function of its variables
    - Value = 1 when constraints are satisfied

Analysis Using BBDs

- A BDD can represent the feasible set of an optimization problem.
  - View constraint set as a boolean function of its variables
    - Value = 1 when constraints are satisfied
  - Nonbinary finite domain variables can be accommodated
    - Write finite domain variable as vector of binary variables

Analysis Using BBDs

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

Analysis Using BBDs

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

Analysis Using BBDs

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

Previous Work

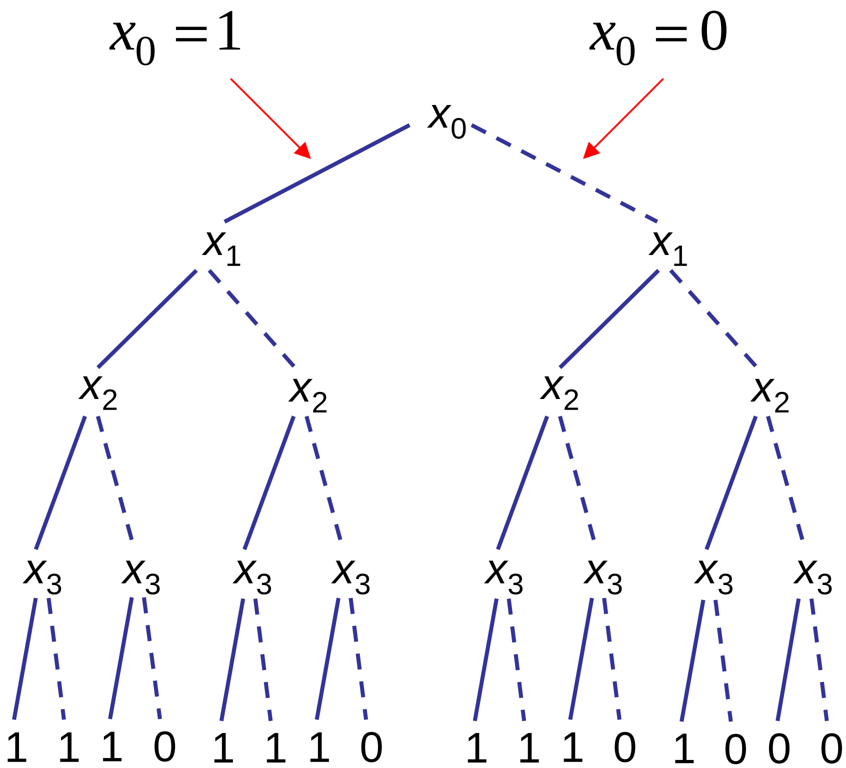
- Becker, Behle, Eisenbrand, Wimmer (2005)
  - Use BDDs to obtain valid cuts for 0-1 linear programming.
  - Cuts are derived from small subsets of constraints.
    - Generate BDD for subset.
    - Use subgradient optimization to solve Lagrangean dual on BDD to obtain separating cuts.

Current Status

- We are in the **initial stages** of research.
- For now, we are using off-the-shelf software to compute BDDs.
  - CLab 1.0, developed at IT Univ. of Copenhagen.
- We are investigating more efficient ways to build BDDs for optimization problems.

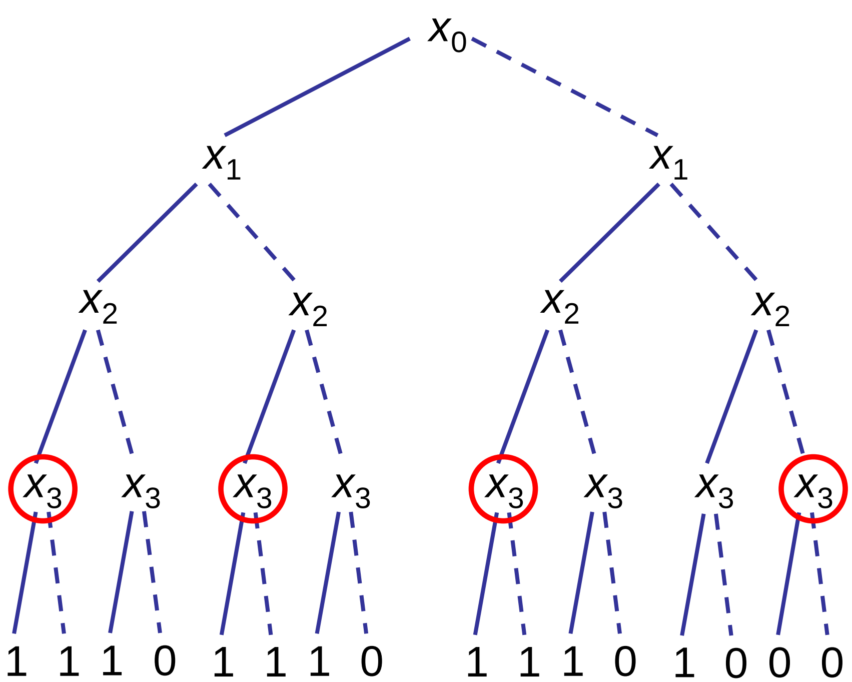
Binary Decision Diagrams

- A reduced BDD for a constraint set is a compact representation of the branching tree for a 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

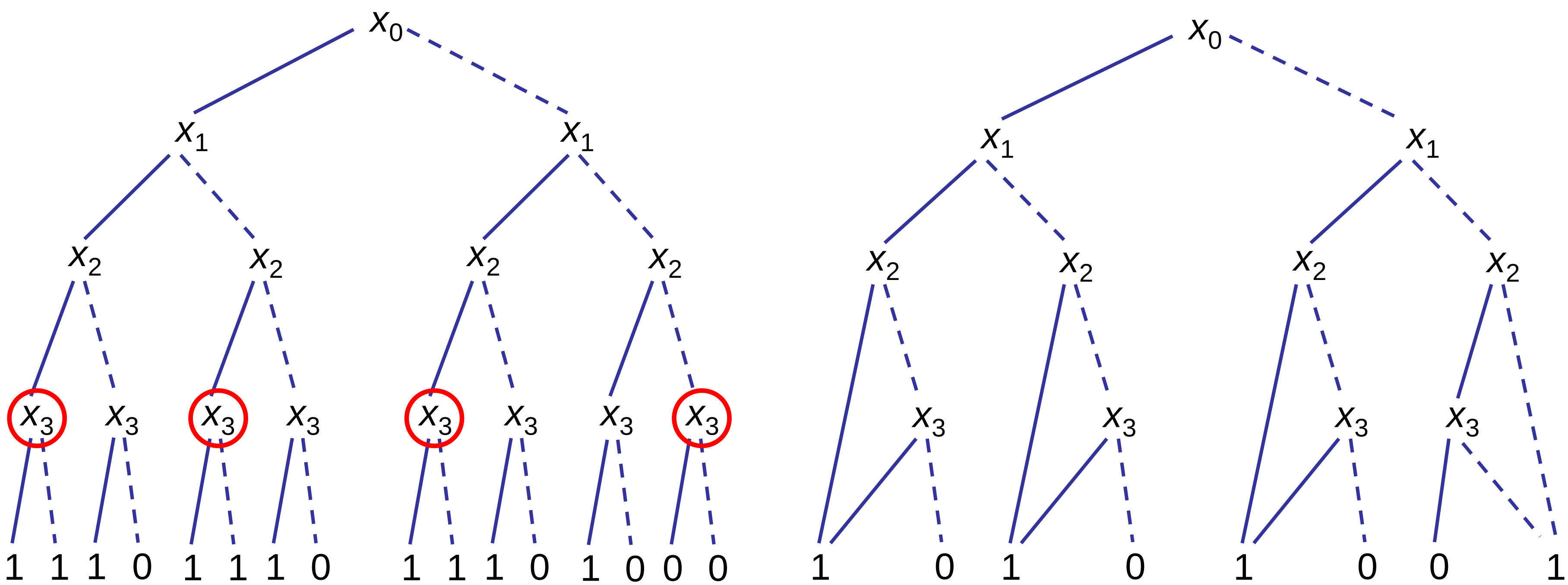
$$2x_0 + 3x_1 + 5x_2 + 5x_3 \geq 7$$



Branching tree for 0-1 inequality

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

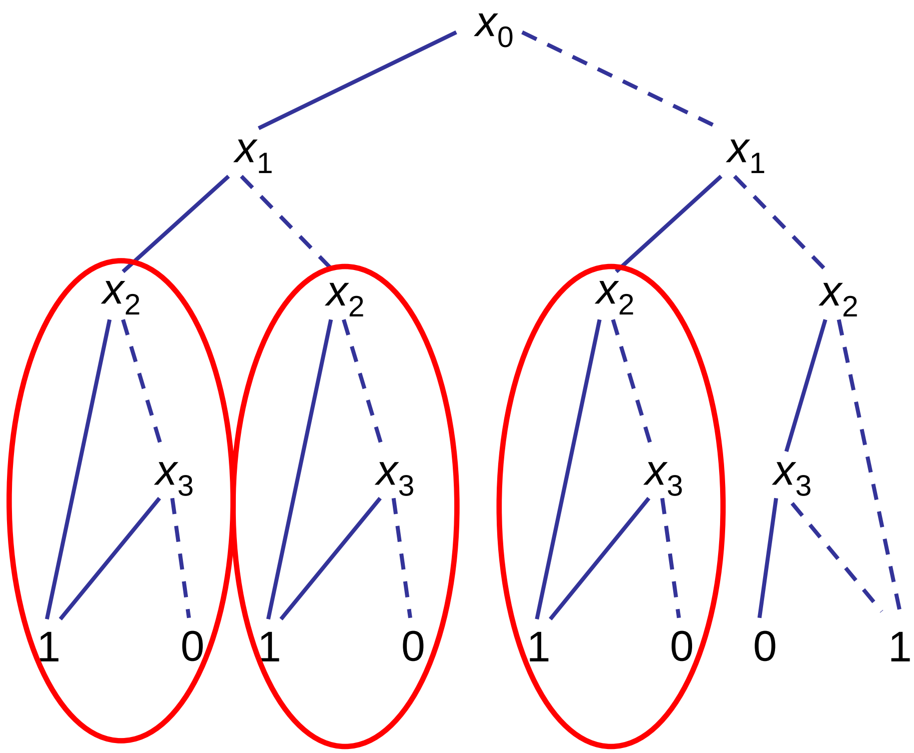
Remove redundant nodes…

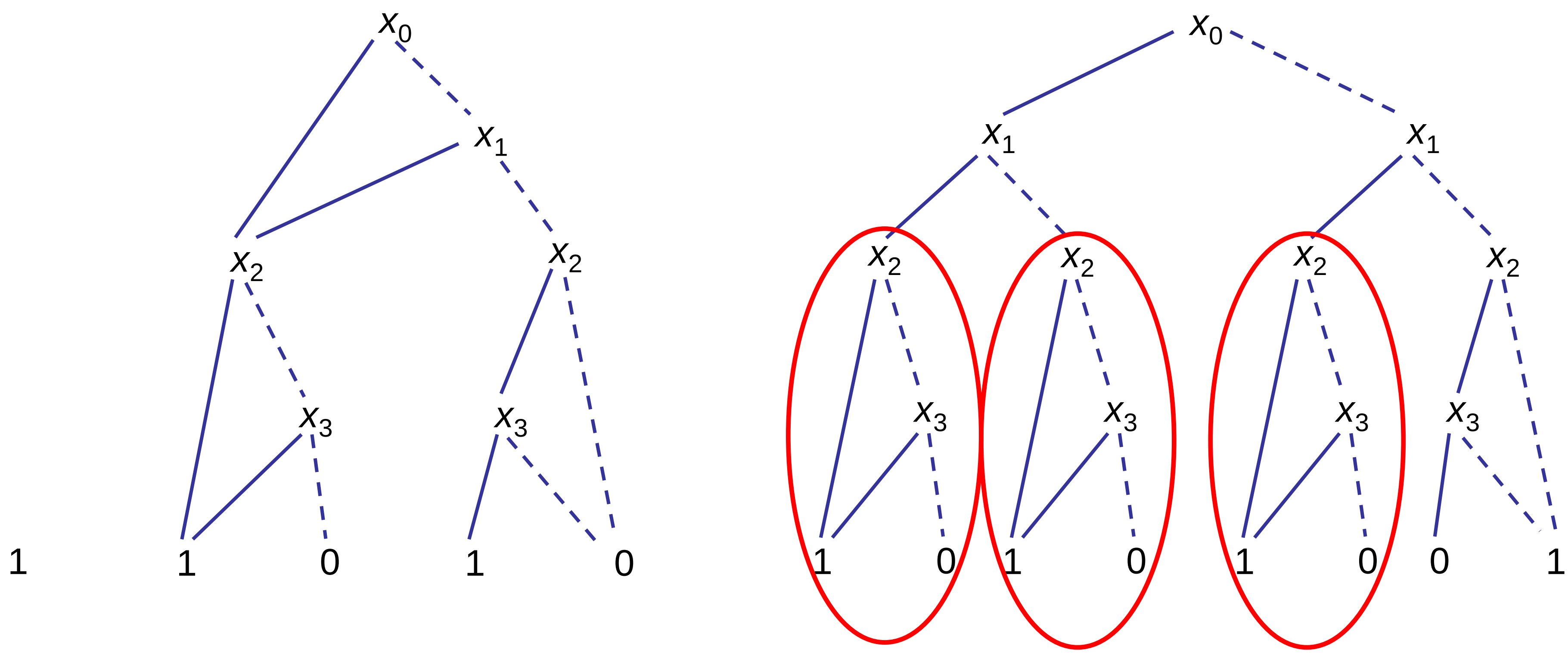


x0
x1
x1
x2
x2
x2
x2
x3
x3
x3
x3
x3
x3
x3
x3
1 1 1 0 1 1 1 0 1 1 1 0 1 0 0 0
x0
x1
x1
x2
x2
x2
x2
x3
x3
x3
x3
1 0 1 0 1 0 0 1

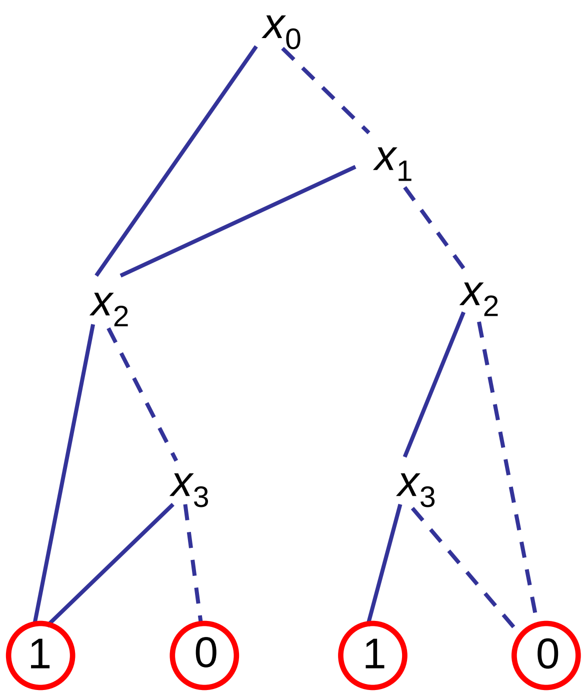
Superimpose identical subtrees…

1

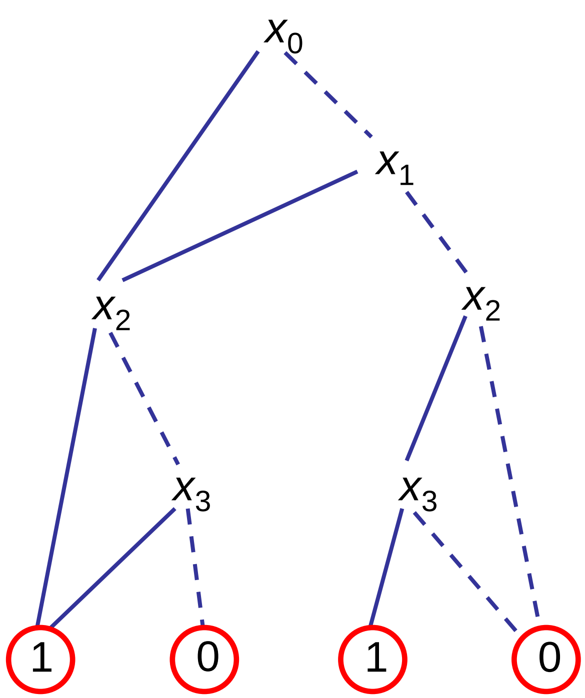




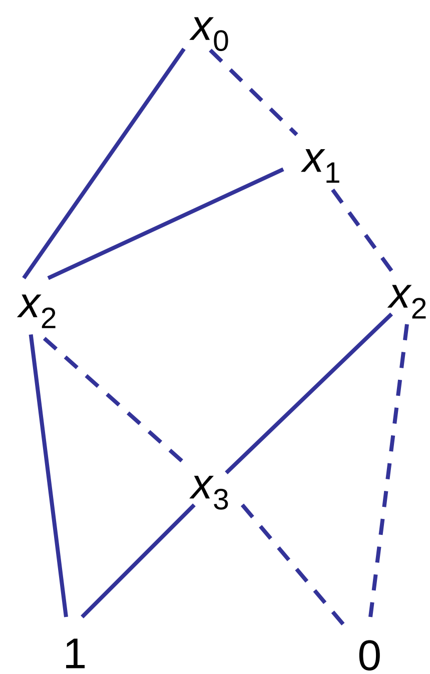
1
x0
x1
x2
x3
1 1 0 1 0
x0
x1 x1
x2 x2 x2 x2
x3 x3 x3 x3
1 0 1 0 1 0 0 1



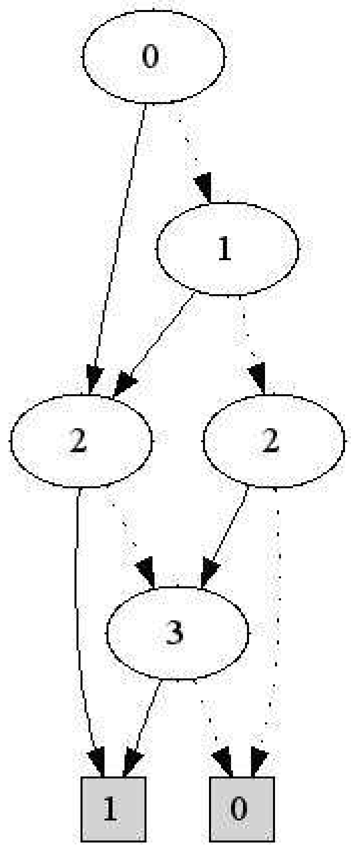
Superimpose identical leaf nodes…



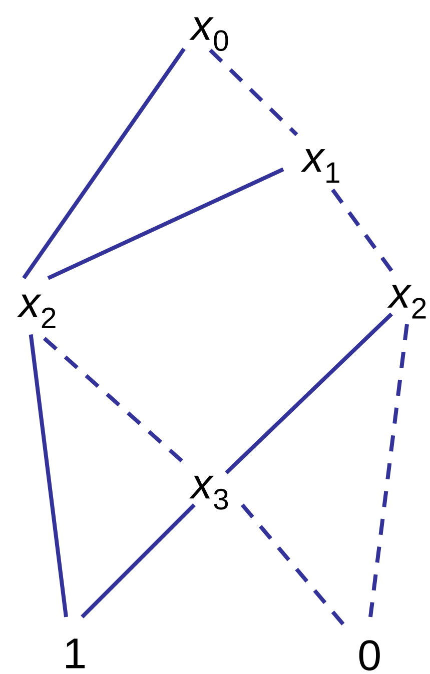
x0
x1
x2
x2
x3
x3
1
0
1
0



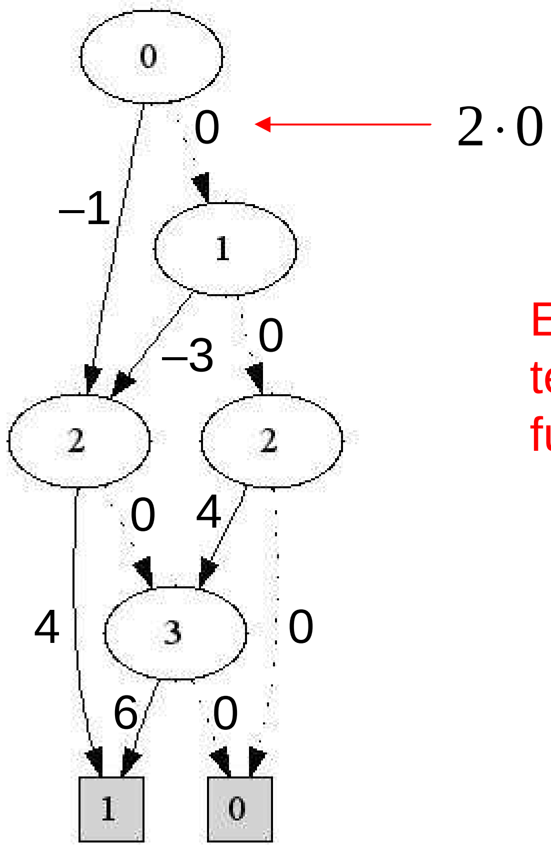
x0
x1
x2
x2
x3
1
0



as generated by software



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

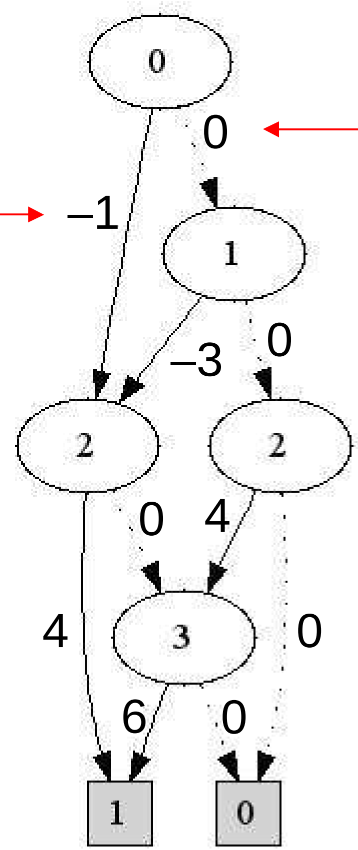


Edge lengths reflect terms of objective function

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

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

$$2 \cdot 0$$



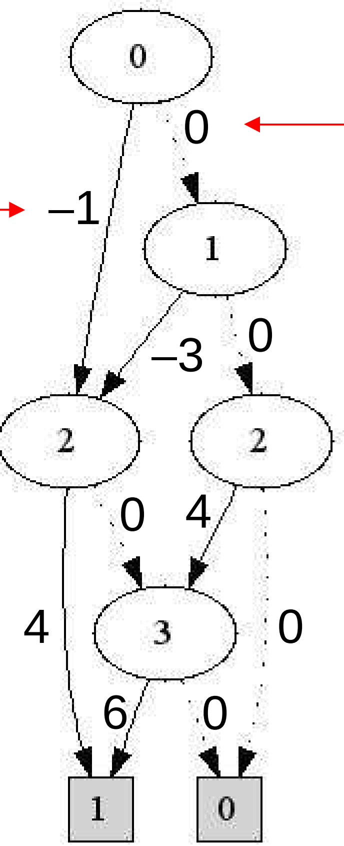
Edge lengths reflect terms of objective function

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

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

$$2 \cdot 0$$

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



Edge lengths reflect terms of objective function

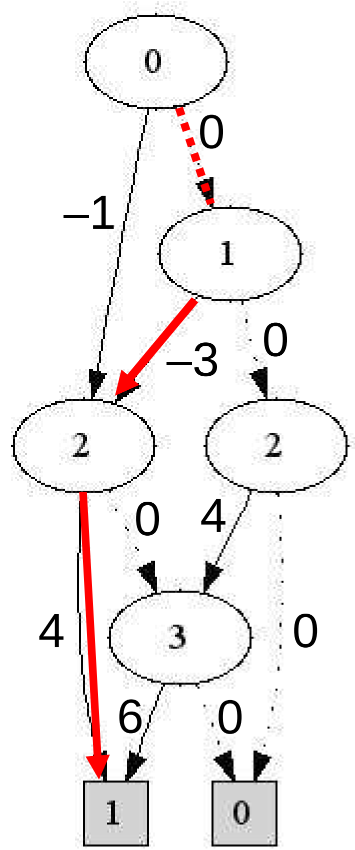
Min $2x_0 - 3x_1 + 4x_2 + 6x_3$ subject to $2x_0 + 3x_1 + 5x_2 + 5x_3 \geq 7$

Shortest path has length 1

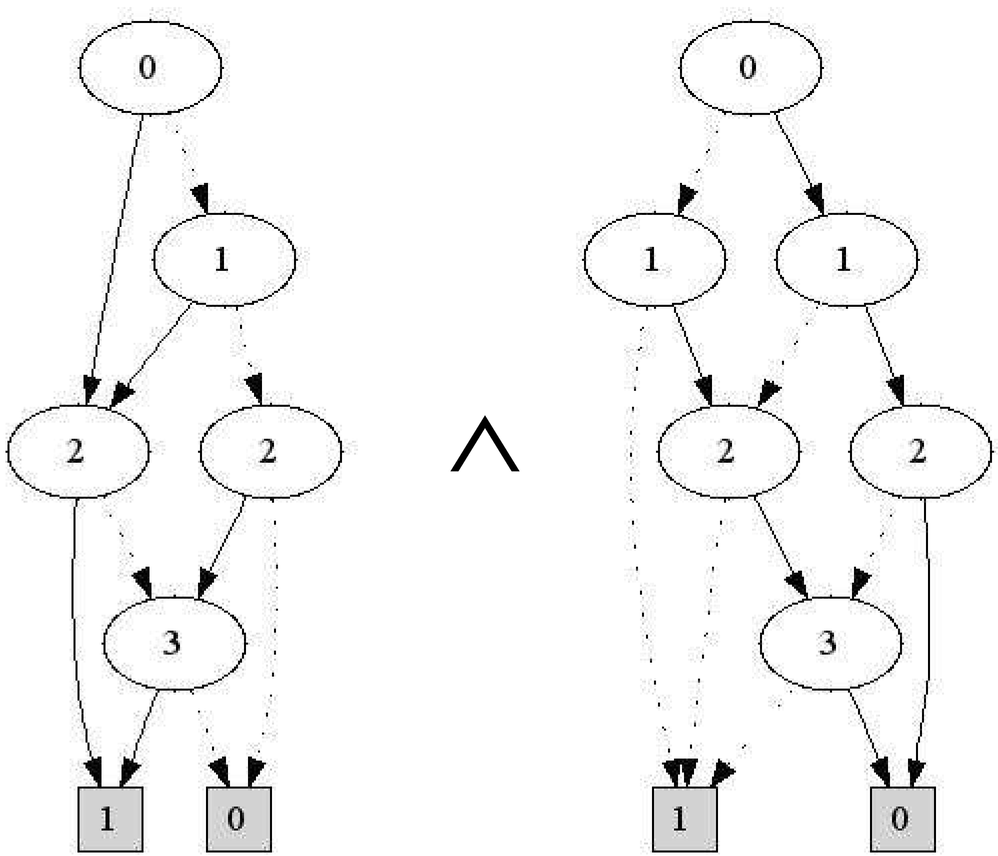
Optimal solution:

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

Set to minimizing value

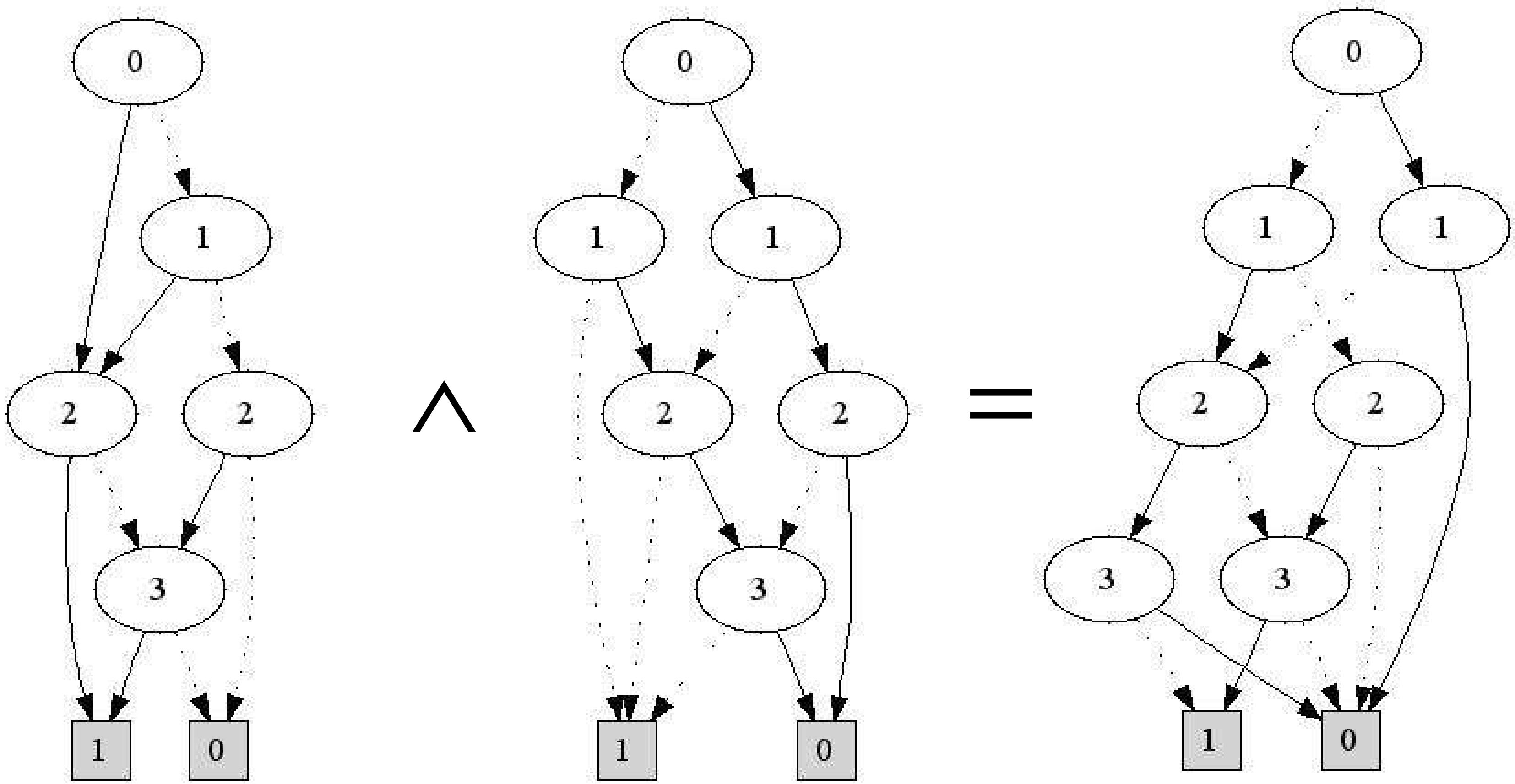


Conjunction of BDDs



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

Conjunction of BDDs



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

Binary Decision Diagrams

- BDD for a knapsack constraint can be surprisingly small…

Binary Decision Diagrams

- BDD for a knapsack constraint can be surprisingly small…

The 0-1 inequality

$$300x_0 + 300x_1 + 285x_2 + 285x_3 + 265x_4 + 265x_5 + 230x_6 + 23x_7 + 190x_8 + 200x_9 + 400x_{10} + 200x_{11} + 400x_{12} + 200x_{13} + 400x_{14} + 200x_{15} + 400x_{16} + 200x_{17} + 400x_{18} \geq 2701$$

has 117,520 minimal feasible solutions

Binary Decision Diagrams

- BDD for a knapsack constraint can be surprisingly small…

The 0-1 inequality

$$300x_0 + 300x_1 + 285x_2 + 285x_3 + 265x_4 + 265x_5 + 230x_6 + 23x_7 + 190x_8 + 200x_9 + 400x_{10} + 200x_{11} + 400x_{12} + 200x_{13} + 400x_{14} + 200x_{15} + 400x_{16} + 200x_{17} + 400x_{18} \geq 2701$$

has 117,520 minimal feasible solutions

Or equivalently,

$$300x_0 + 300x_1 + 285x_2 + 285x_3 + 265x_4 + 265x_5 + 230x_6 + 23x_7 + 190x_8 + 200x_9 + 400x_{10} + 200x_{11} + 400x_{12} + 200x_{13} + 400x_{14} + 200x_{15} + 400x_{16} + 200x_{17} + 400x_{18} \leq 2700$$

has 117,520 minimal covers

Binary Decision Diagrams

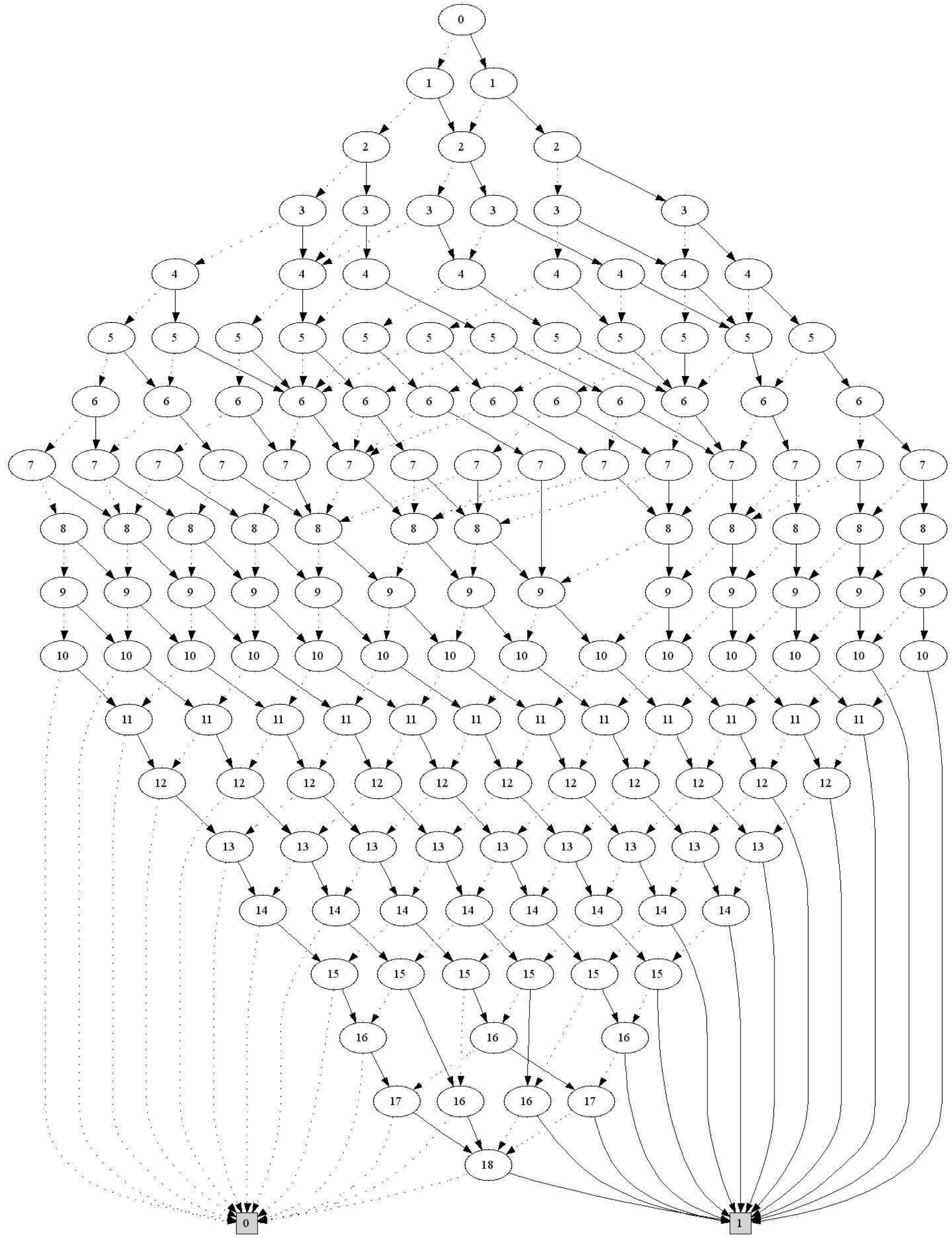
- BDD for a knapsack constraint can be surprisingly small…

The 0-1 inequality

$$300x_0 + 300x_1 + 285x_2 + 285x_3 + 265x_4 + 265x_5 + 230x_6 + 23x_7 + 190x_8 + 200x_9 + \\ 400x_{10} + 200x_{11} + 400x_{12} + 200x_{13} + 400x_{14} + 200x_{15} + 400x_{16} + 200x_{17} + 400x_{18} \geq 2701$$

has 117,520 minimal feasible solutions

But its reduced BDD has only 152 nodes…



Finite Domain Variables

- Expand finite domain variable into several binary variables.
  - BDD is layered, each layer containing binary variables that correspond to one finite domain variable.
  - For shortest path computations, add edges between layers to create *augmented graph*.

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:

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

where

$$c_j = \text{ return from facility } j$$
$$a_j = \text{ cost of facility } j, \quad b = \text{ budget}$$
$$x_j = \text{ number of facilities of type } j$$

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.
- 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.

$x_1(0)$

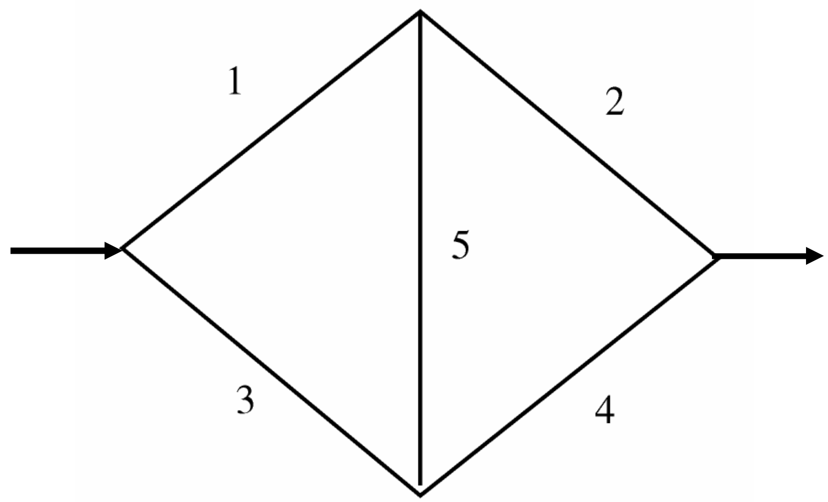
$c_{opt} - \Delta$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
3678:	0	3	2	0	0	0	1	1	2	0
3677:			1		1			3	0	
3676:							0		1	
3673:	1,2	0,1,2	3		3		2,3	2	3	1
3672:					2			0		
3669:										2
3668:			0							
3666:						1				
3664:										3
3658:				1						
3657:	3					2				
3646:						3				
3633: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

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

$$R = R_1 R_2 + (1 - R_2) R_3 R_4 + (1 - R_1) R_2 R_3 R_4 + R_1 (1 - R_2)(1 - R_3) R_4 R_5 + (1 - R_1) R_2 R_3 (1 - R_4) R_5$$

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

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

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

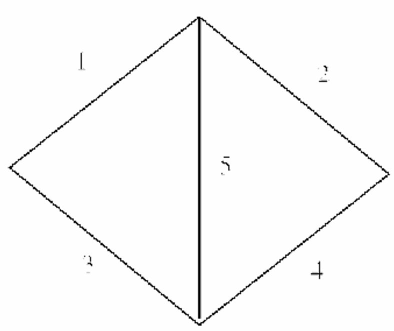
$r = (0.9, 0.85, 0.8, 0.9, 0.95)$

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

Cost-based
domain analysis

308 nodes in BDD

1.1 seconds
to compile BDD

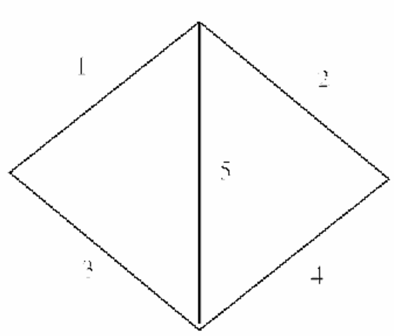


$c_{opt} + \Delta$	x_1	x_2	x_3	x_4	x_5	R
50:	0	0	1	1	0	72
60:	1	1	0	0,2		79
85:	2					84
90:			2	3		86
95:		2			1	88
100:						95
120:						97
125:	3					
155:		3			2	
160:						98
170:						99
180:			3			
230:					3	

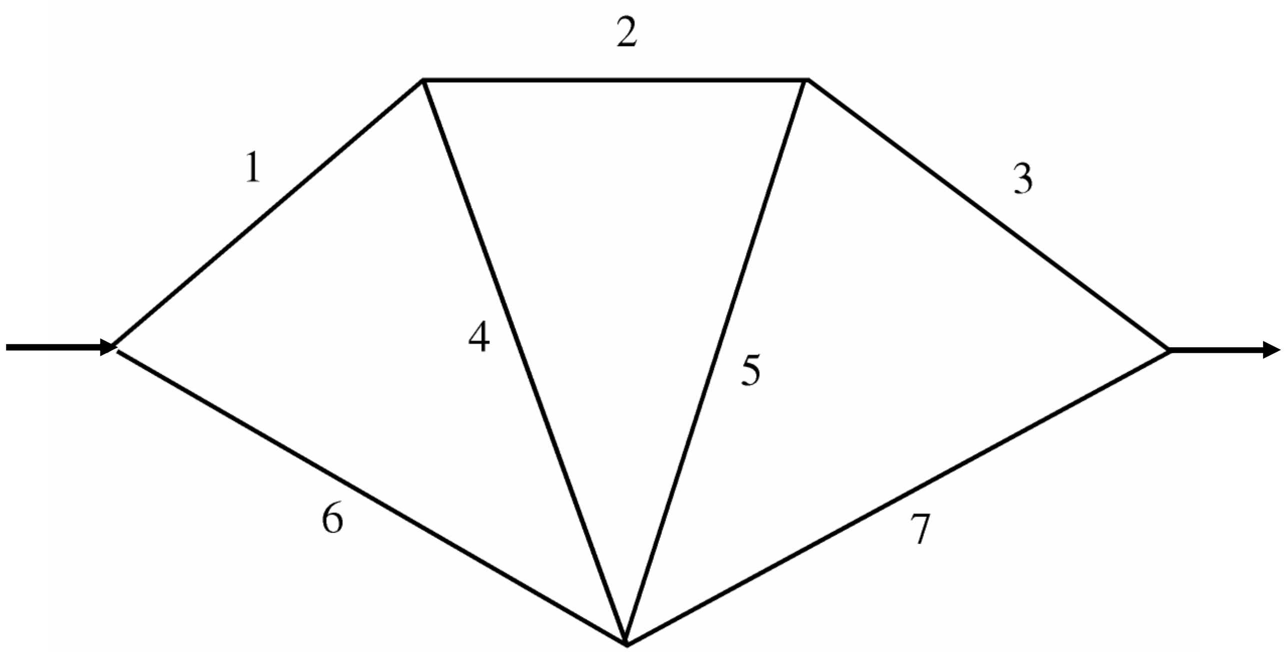
Domain analysis with respect to *R*

Same BDD as before

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				



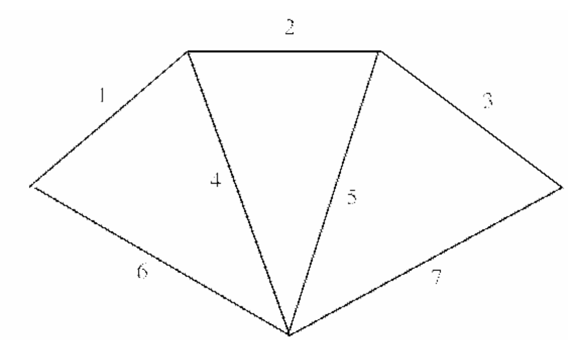
7 bridges



Cost-based domain analysis

1779 nodes in BDD

14.8 seconds to compile BDD

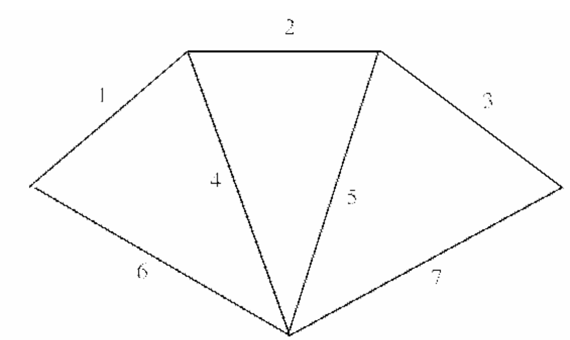


$c_{opt} + \Delta$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	R
9:	0	0	0	0	0	1	1	72.2
11:			1		1		0	
12:	1			1		0		
13:		1				2		82.9
14:					2		2	
15:			2	2				
16:	2							
17:					3	3		84.6
18:		2		3				95.2
19:			3				3	
20:	3							
22:								97.2
23:		3						
27:								99.2
34:								99.4
40:								99.6
43: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

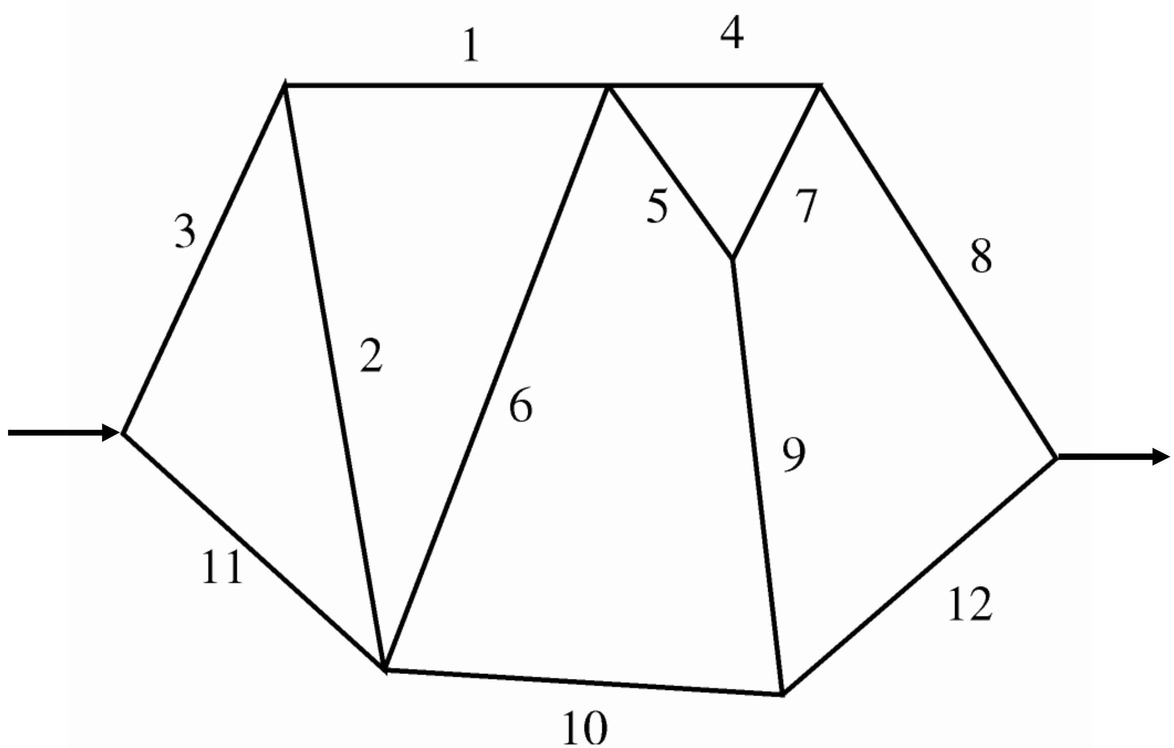
Domain analysis with respect to R

Same BDD as before

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	



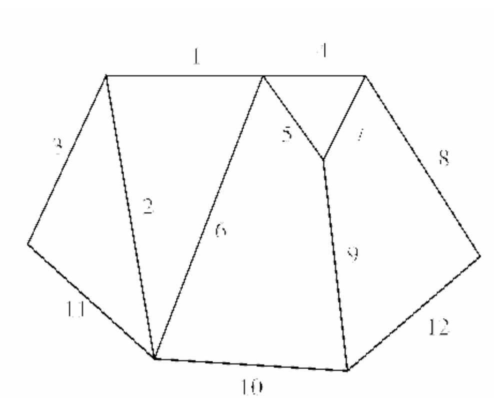
12 bridges



Cost-based domain analysis

69,457 nodes in BDD

2933 seconds
to compile BDD

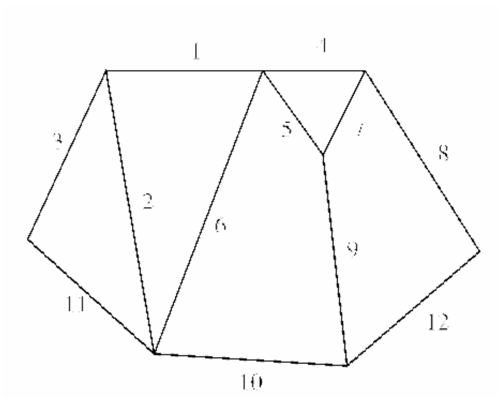


$c_{opt} + \Delta$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	R
180	1	0	2	3	0	0	0	2	0	0	0	0	80
185			3	2									82
190								3					
195													83
200										1			
205													86
210	2						1						
215													88
220										2			
225					1				1				
230	0		0			1,2					1		
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 BDD as before



Portfolio Design

- Maximize expected return subject to an upper bound on variance.

$$\max \sum_{i=1}^{n} \mu_i x_i$$

$$\sum_{i=i}^{n} \sum_{j=1}^{n} c_i c_j \sigma_{ij} x_i x_j \leq V_{max}$$

$$\sum_{i=1}^{n} c_i x_i \leq W$$

$$\sum_{i=1}^{n} \delta(x_i) \leq K$$

$$x_i \in D_i, \; i = 1, \ldots, n$$

Expected yield rate of security i

Number of blocks of security *i* purchased

$$\max \sum_{i=1}^{n} \mu_i x_i$$

Variance/covariance

Maximum variance

$$\sum_{i=i}^{n} \sum_{j=1}^{n} c_i c_j \sigma_{ij}^2 x_i x_j \leq V_{\max}$$

Cost of one block of security *i*

$$\sum_{i=1}^{n} c_i \cdot x_i \leq W$$

Maximum investment

1 if $x_i > 0$, 0 otherwise *(no need to model this with integer variables)*

$$\sum_{i=1}^{n} \delta(x_i) \leq K$$

Maximum number of securities in portfolio

$$x_i \in D_i, \; i = 1, \ldots, n$$

n = 10 securities

Weekly yield rates and variances/covariances from from Hang Seng Index

$$\max \sum_{i=1}^{n} \mu_i x_i$$

Use 15 most significant entries of variance/covariance matrix

$$\sum_{i=i}^{n} \sum_{j=1}^{n} c_i c_j \sigma_{ij}^2 x_i x_j \leq V_{\max}$$

4.5 million HKD

$$\sum_{i=1}^{n} c_i \cdot x_i \leq W$$

Max 7 securities

$$\sum_{i=1}^{n} \delta(x_i) \leq K$$

D_i = {0,…,7}
That is, 0 to 7 blocks of each security

$$x_i \in D_i, \; i = 1, \ldots, n$$

Cost-based domain analysis

59,802 nodes in BDD

63 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}
21797	6	0	7	0	3	7	6	0	3	7
21754						6	7			
21705				1					0	6
21683									2	
21678	7		6							
21673					5					
21670					4					
21663				2	0					
21647								2		
21642							5		1	
21630					2					
21624								1,3		
21604					1					
21599						5				
21572			5							
21567									4	
21562										5
21532	5									
21529							4			
21484									5	4
21467						4				
21456								4		
21412					6					
21404							3			
21370		1								
21351								5		
21330									6	
21312			4							
21232						3				
21215										3
21134					7					
21133		2								

Yield/risk tradeoff

Y_{risk} = 10% downside risk

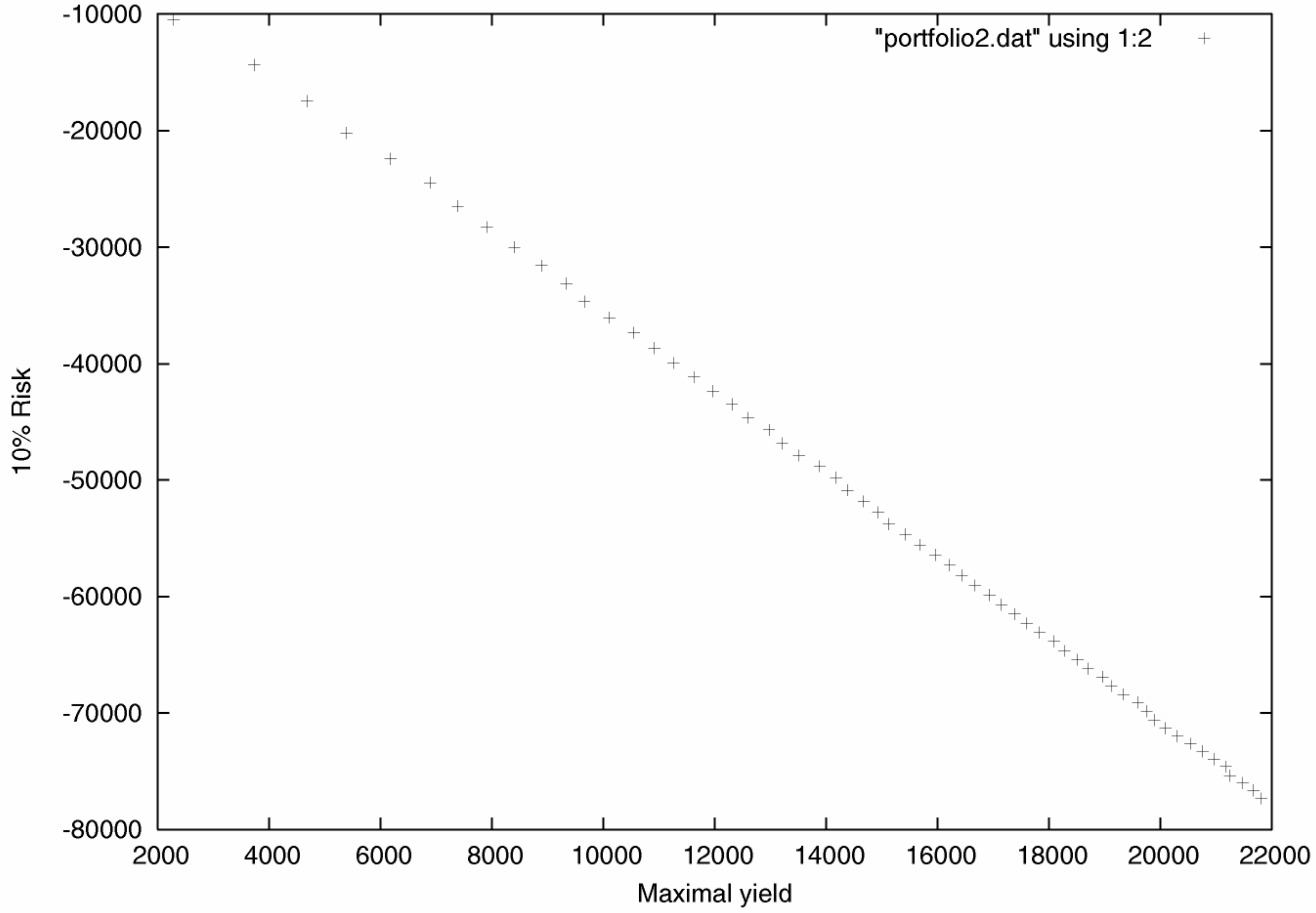
= $Y_{max} - 1.28\sigma$

636,568 nodes in BDD since V_{max} is a variable

186 seconds to compile BDD

Y_{max}	Y_{risk}	V_{max}	Y_{max}	Y_{risk}	V_{max}
21812	-77336	60	15426	-54682	30
21671	-76647	59	15128	-53802	29
21479	-76002	58	14936	-52795	28
21254	-75383	57	14677	-51833	27
21177	-74609	56	14383	-50884	26
20960	-73967	55	14183	-49817	25
20760	-73300	54	13889	-48817	24
20543	-72642	53	13515	-47871	23
20309	-71993	52	13213	-46824	22
20092	-71318	51	12978	-45678	21
19892	-70617	50	12604	-44639	20
19756	-69844	49	12310	-43483	19
19598	-69083	48	11969	-42336	18
19339	-68413	47	11633	-41142	17
19121	-67692	46	11259	-39941	16
18963	-66902	45	10922	-38652	15
18704	-66201	44	10548	-37345	14
18512	-65423	43	10113	-36038	13
18277	-64676	42	9662	-34678	12
18095	-63864	41	9343	-33109	11
17836	-63118	40	8892	-31585	10
17601	-62334	39	8406	-29994	9
17384	-61520	38	7912	-28291	8
17150	-60709	37	7389	-26476	7
16933	-59867	36	6895	-24458	6
16674	-59051	35	6185	-22436	5
16439	-58197	34	5392	-20208	4
16221	-57309	33	4681	-17489	3
15962	-56445	32	3736	-14365	2
15686	-55581	31	2291	-10509	1

Yield/risk tradeoff



Future Research

- More efficient BDD construction for postoptimality analysis.
  - Use known optimal cost to create BDD of near-optimal solutions.
  - Reconstruct BDD of near-optimal solutions from branch-and-bound tree.
  - Integrate BDD construction and branch-and-bound search.
  - Variable ordering heuristics.

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.
  - BDD as basis for explanation (generalized duality).
- Extension to discrete/continuous problems.

Other Uses for BDDs

- Polyhedral relaxations of BDDs
  - Apply to subsets of constraints.
- BDD relaxations.
  - Replace domain store with BDD store.