Decision Diagrams for Constraint Programming

Willem-Jan van Hoeve

Tepper School of Business

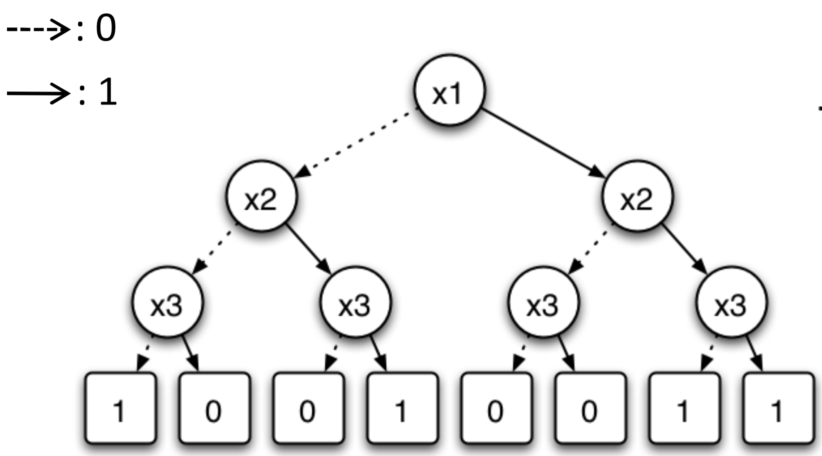
Carnegie Mellon University

Overview

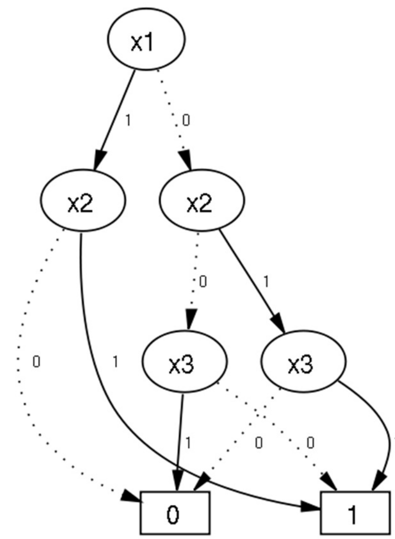
- BDDs and MDDs show great promise for CP

- Goals of this short presentation
  - what are MDDs?
  - how are they used in CP?
  - how can they impact the modeling and solving aspects of CP?

Decision Diagrams



x1	x2	x3	*f*
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

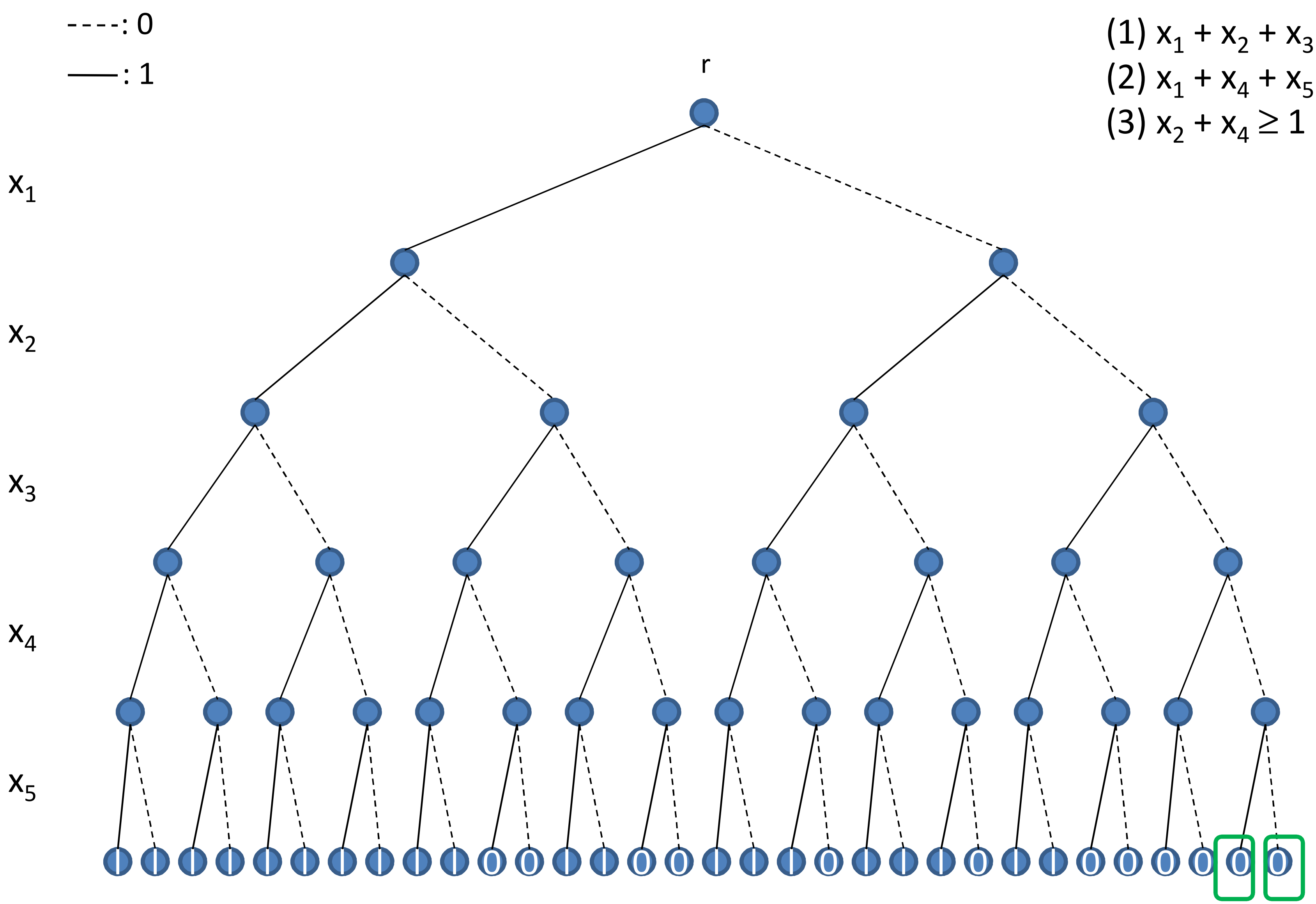


$$f(x1, x2, x3) = (\neg x1 \wedge \neg x2 \wedge \neg x3) \vee (x1 \wedge x2) \vee (x2 \wedge x3)$$

- Binary Decision Diagrams were introduced to compactly represent Boolean functions [Lee, 1959], [Akers, 1978], [Bryant, 1986]
- Main operation: merge isomorphic subtrees of a given binary decision tree
- MDDs are Multi-valued Decision Diagrams (i.e., for discrete variables)

Example: Exact MDD for a CSP

(1) $x_1 + x_2 + x_3 \geq 1$
(2) $x_1 + x_4 + x_5 \geq 1$
(3) $x_2 + x_4 \geq 1$



Example: Exact MDD for a CSP

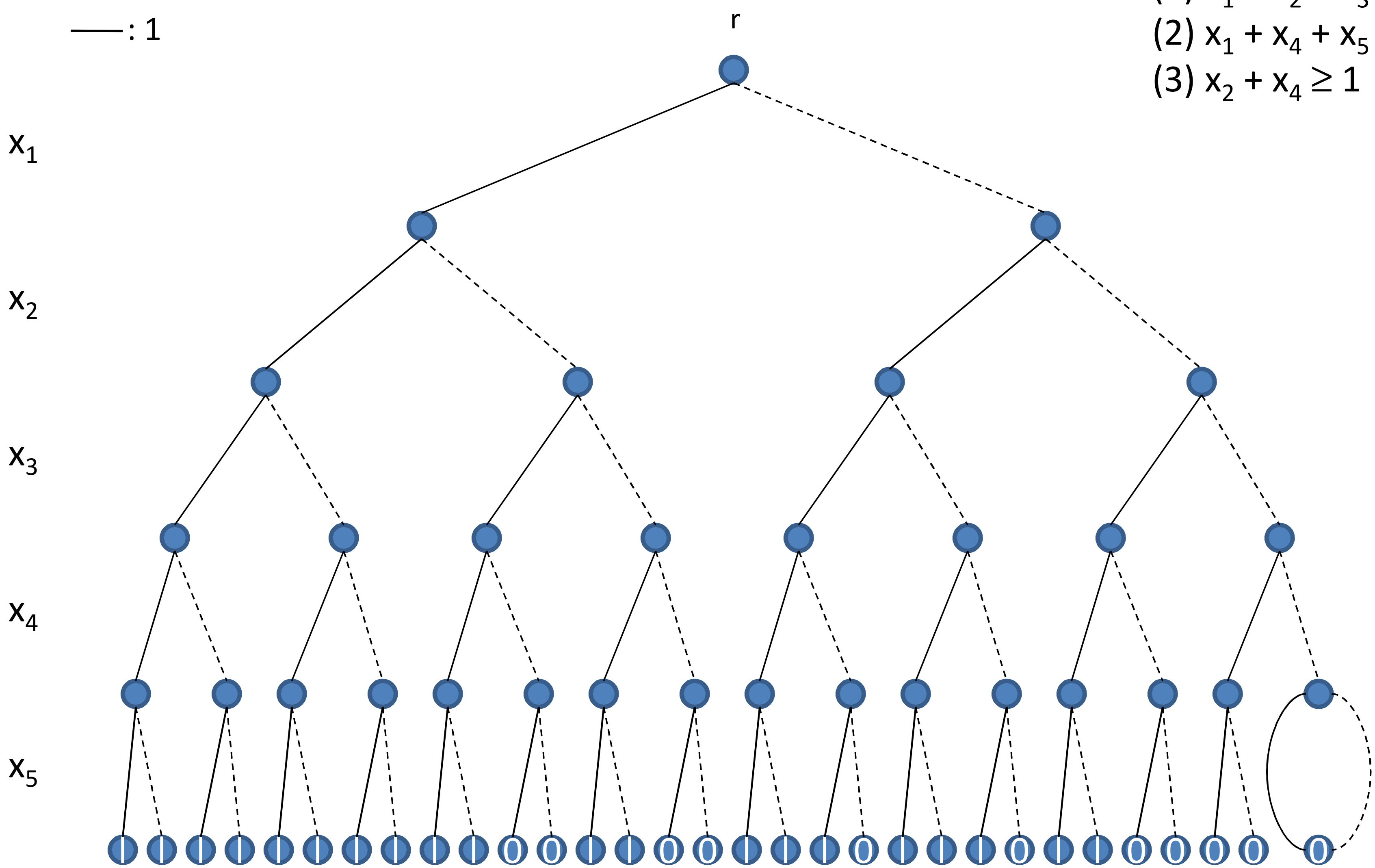
---: 0

——: 1

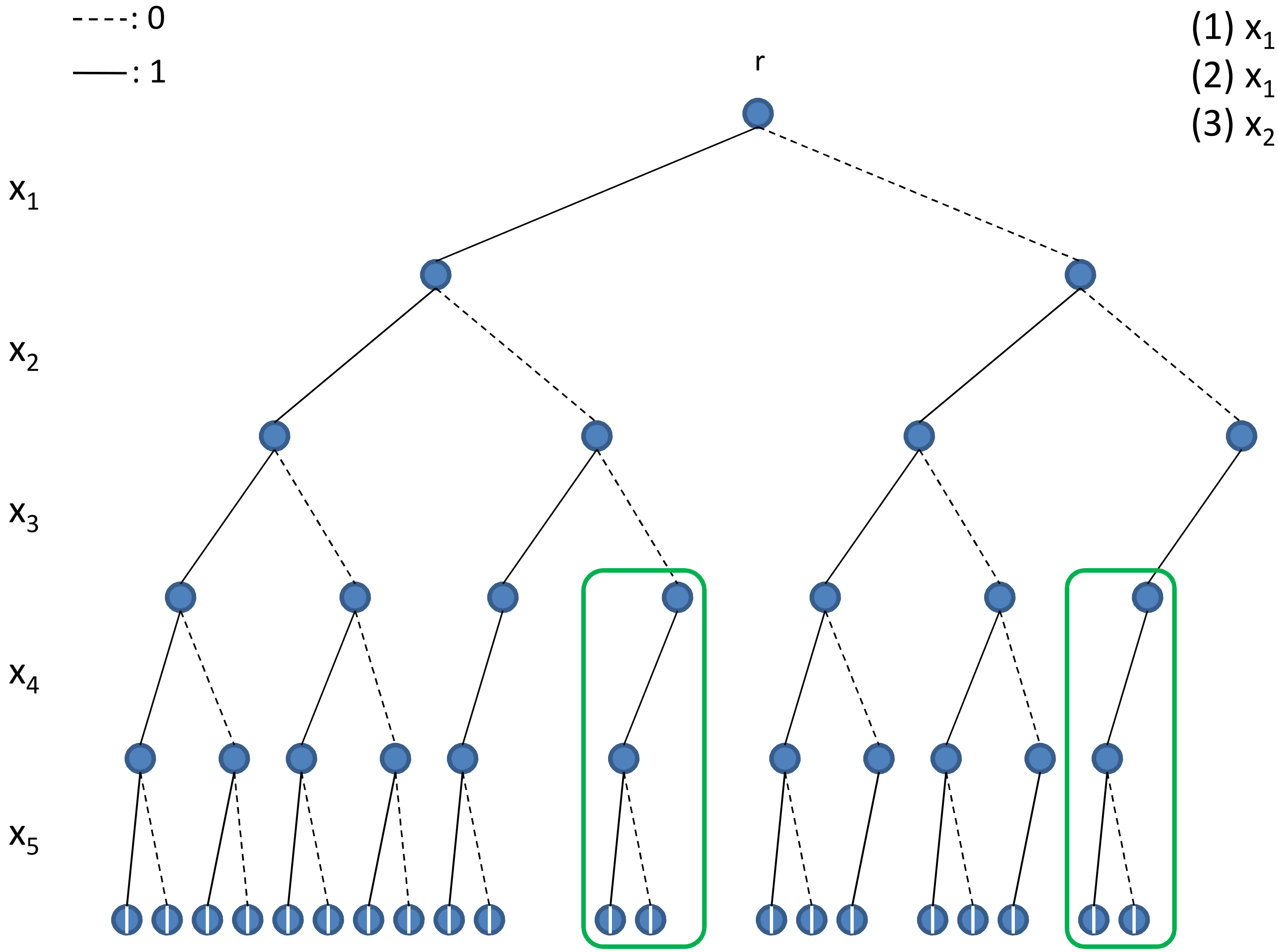
(1) $x_1 + x_2 + x_3 \geq 1$

(2) $x_1 + x_4 + x_5 \geq 1$

(3) $x_2 + x_4 \geq 1$



Example: Exact MDD for a CSP



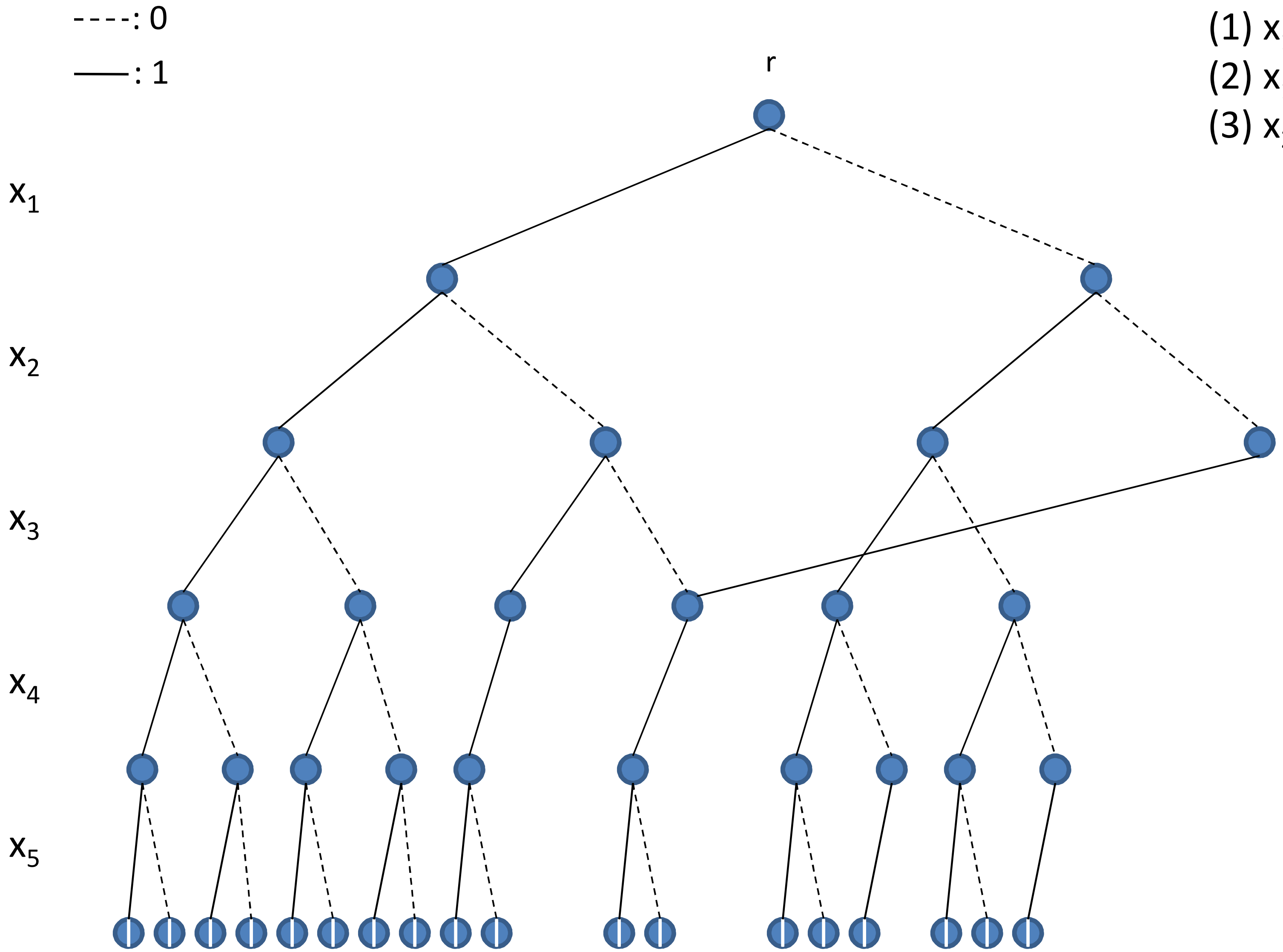
(1) $x_1 + x_2 + x_3 \geq 1$

(2) $x_1 + x_4 + x_5 \geq 1$

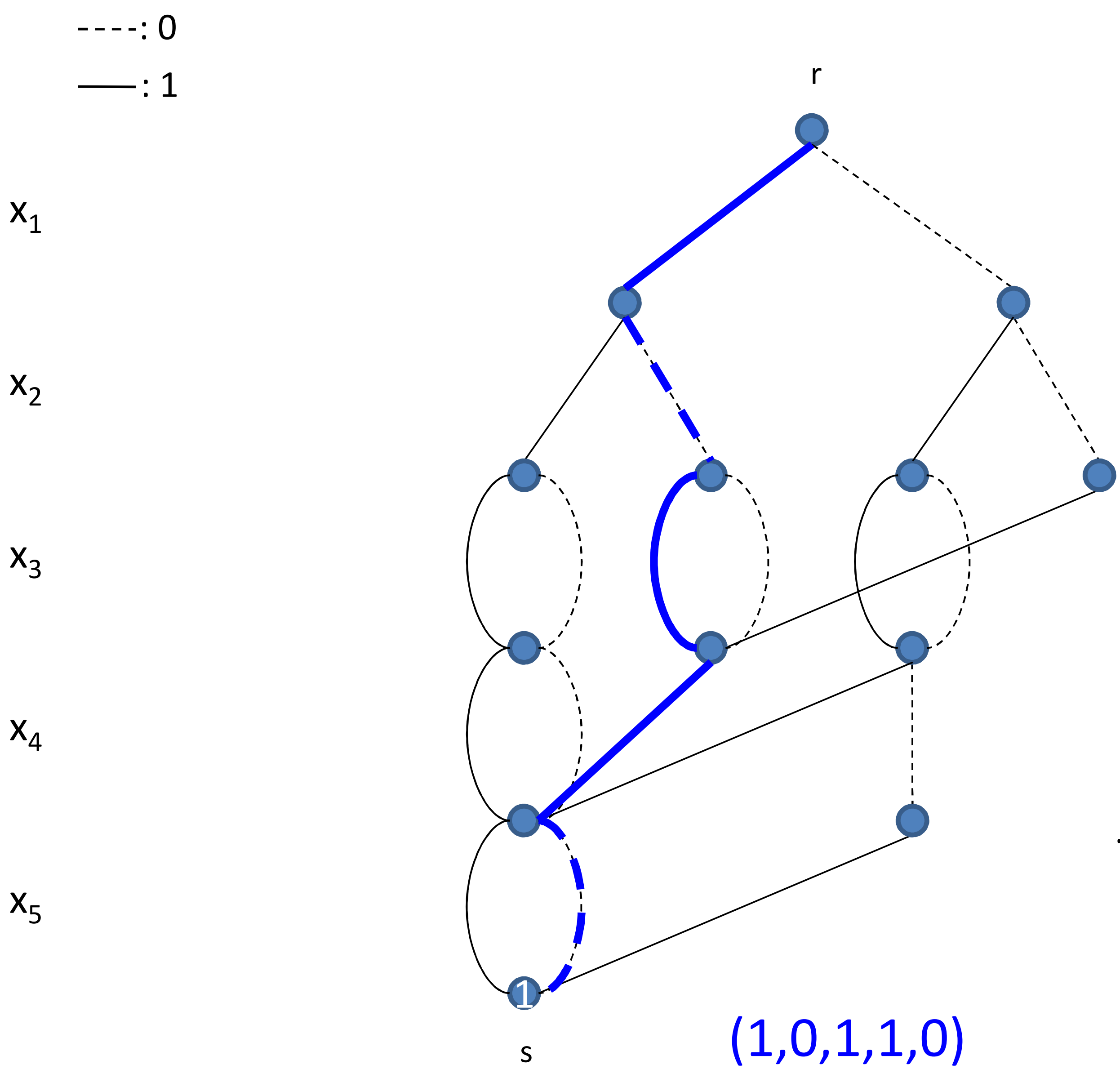
(3) $x_2 + x_4 \geq 1$

Example: Exact MDD for a CSP

(1) $x_1 + x_2 + x_3 \geq 1$
(2) $x_1 + x_4 + x_5 \geq 1$
(3) $x_2 + x_4 \geq 1$



Example: Exact MDD for a CSP



(1) $x_1 + x_2 + x_3 \geq 1$
(2) $x_1 + x_4 + x_5 \geq 1$
(3) $x_2 + x_4 \geq 1$

Each path corresponds to a solution

Exact MDDs

- Compact representation of solution space
  - in some case exponential number of solution can be represented in polynomial size
- Allows to quickly query and process solution space [Hadzic and Hooker, 2006, 2008], [Gange et al., 2011]

Use in CP:

- Table constraints [Cheng and Yap, 2008], regular constraints [Lagerkvist, 2008]
- Set variables [Hawkins et al., 2005]
- Overlapping knapsack constraints [Hadzic et al., 2009]

Approximate MDDs

- Exact MDDs can be of exponential size in general
- We can limit the size of the MDD and still have a meaningful representation [Andersen et al., 2007]

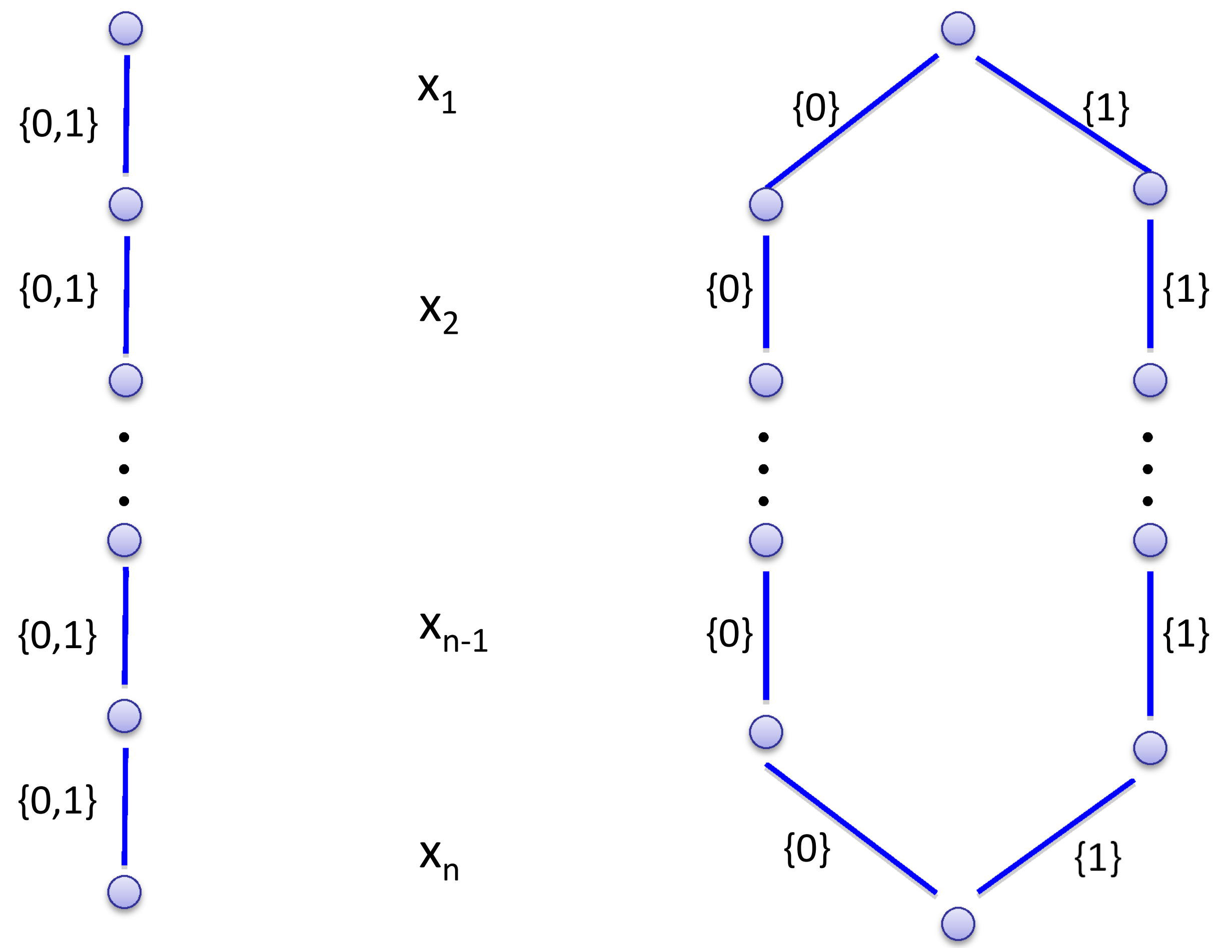
Use in CP:

- Replace domain propagation by *MDD propagation*
  - each constraint gets to filter and refine the MDD
  - Alldiff, linear constraints, element, among, sequence, unary resource scheduling,... [Andersen et al., 2007], [Hadzic et al., 2008], [Hoda et al., 2010], [v.H., 2011], [Cire and v.H., 2011]
- MDD relaxations for optimization
  - lower and upper bounds [Bergman et al., 2011]

Illustrative Example

Carnegie Mellon Tepper School of Business

AllEqual$(x_1, x_2, \ldots, x_n)$, all x_i binary

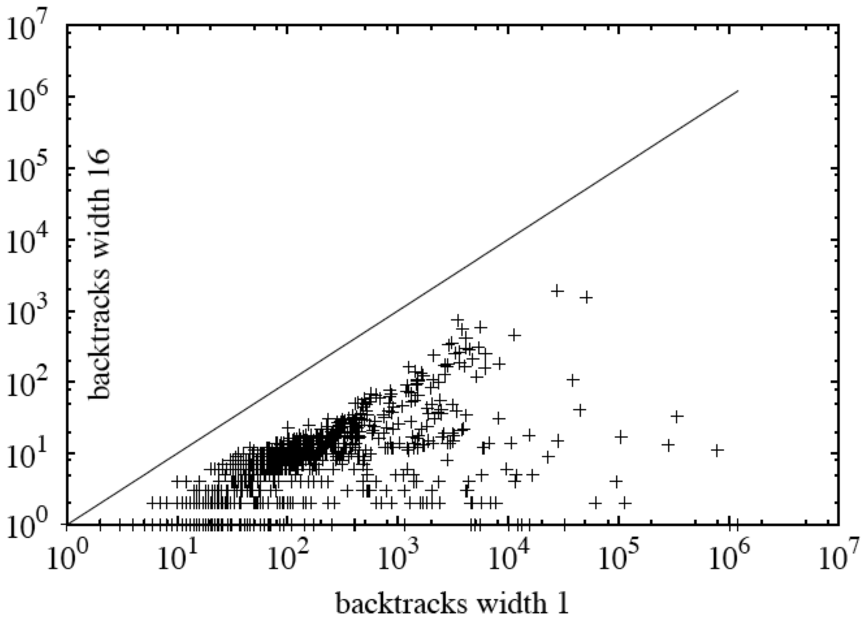


domain representation, size 2^n

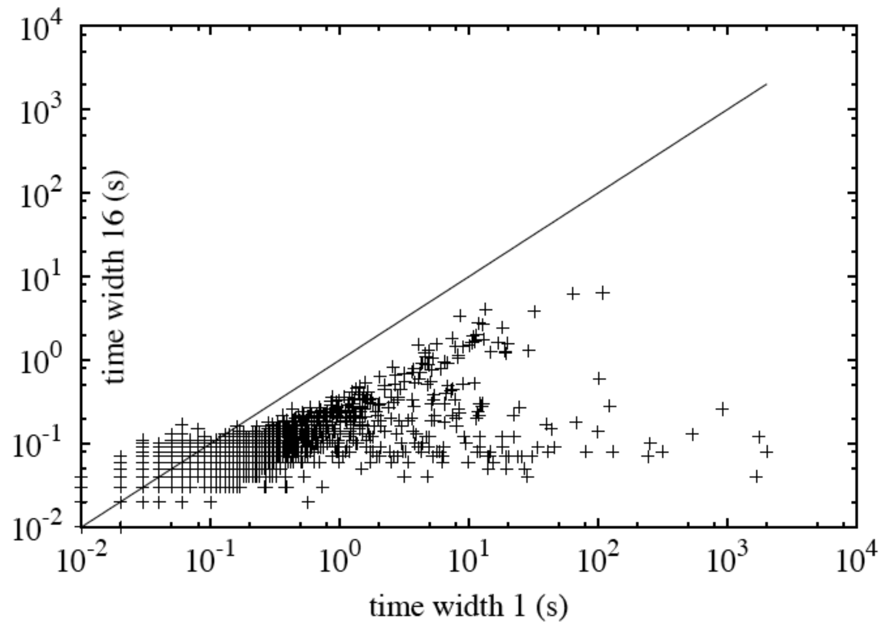
MDD representation, size 2

Benefits

- Limited-width MDDs can yield orders of magnitude reductions in search tree size and computation time



multiple amongs
#backtracks, width 1 vs 16



multiple amongs
time (s), width 1 vs 16

[Hoda et al., CP 2010]

Integration into CP systems: Modeling

- MDDs for individual (global) constraints
  - just add MDD propagator, invisible to user
- MDDs as propagation tool
  - propagate MDD between constraints
  - probably most effective on subsets of constraints
  - user could provide information which constraints should be grouped together, and how effort is spent
  - ideally, however, the solver should automatically group constraints together

- Thus, impact on user can be minimal w.r.t. model

Integration into CP systems: Solving

- Most likely, MDD propagation will be used in parallel to domain propagation
  - we need close interaction between MDD representation and domain representation
  - projection of MDD onto variable domains is typically weak

- MDDs for individual constraints
  - CP solving mechanism is almost unchanged
- MDDs as propagation tool
  - maintain and manipulate MDD efficiently during search