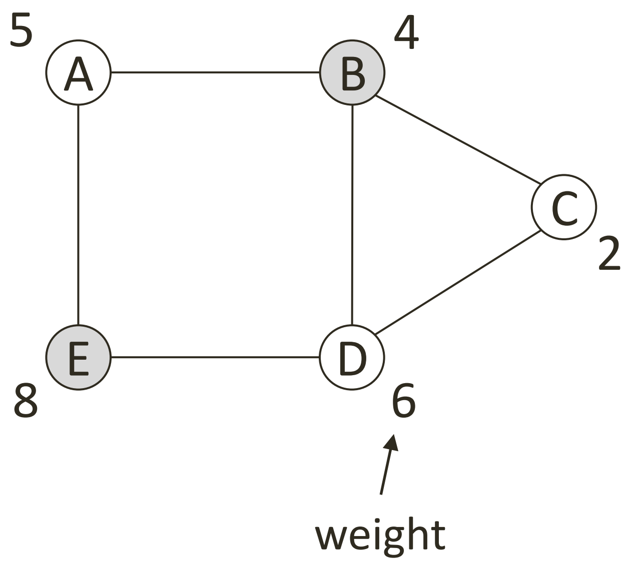
Carnegie Mellon University
Tepper School of Business

Decision Diagrams for Discrete Optimization

Willem-Jan (Willem) van Hoeve
Carnegie Mellon University

15816 Advanced Topics in Logic: Automated Reasoning and Satisfiability – Fall 2022

Example: Maximum Independent Set Problem



Independent set in a graph:

- Subset of non-adjacent vertices

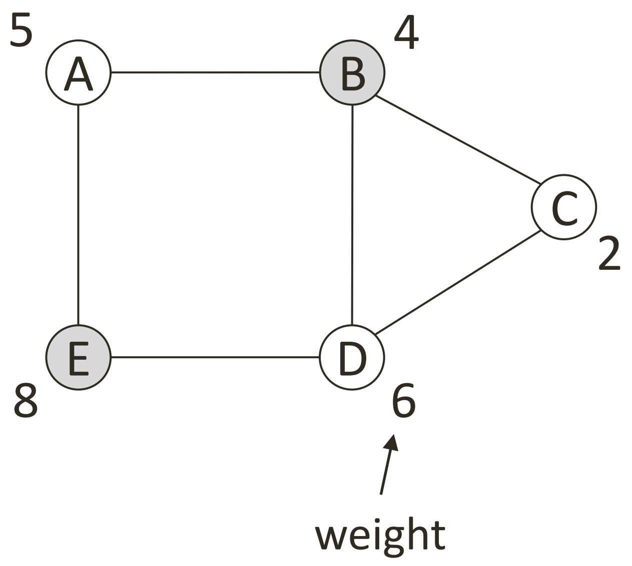
Maximum Independent Set Problem:

- Find independent set with maximum weight

Integer Programming Formulation:

$$
\begin{aligned}
\max\ & 5x_A + 4x_B + 2x_C + 6x_D + 8x_E \\
\text{subject to}\ & x_A + x_B \leq 1 \\
& x_A + x_E \leq 1 \\
& x_B + x_C \leq 1 \\
& x_B + x_D \leq 1 \\
& x_C + x_D \leq 1 \\
& x_D + x_E \leq 1 \\
& x_A, x_B, x_C, x_D, x_E \in \{0,1\}
\end{aligned}
$$

BDDs can Represent Optimization Problems

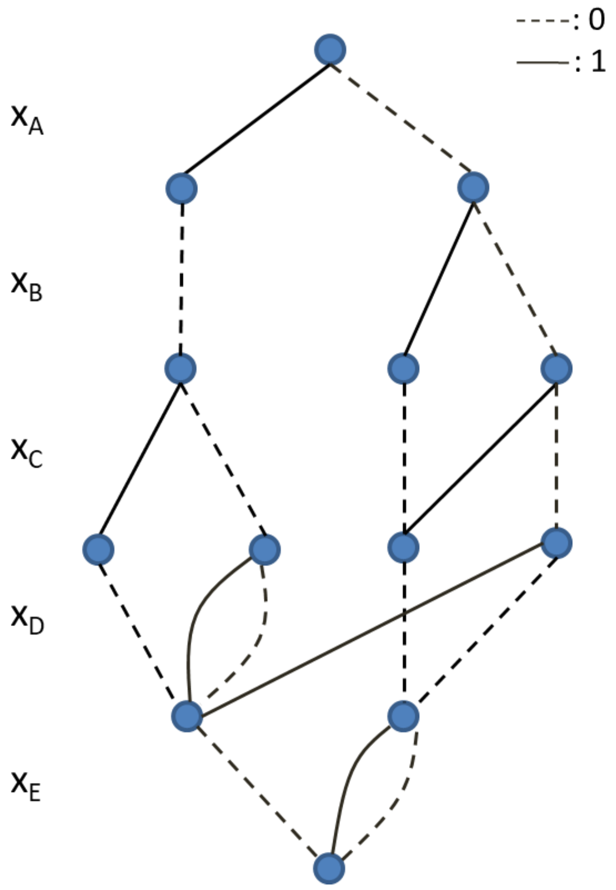


Independent set in a graph:

- Subset of non-adjacent vertices

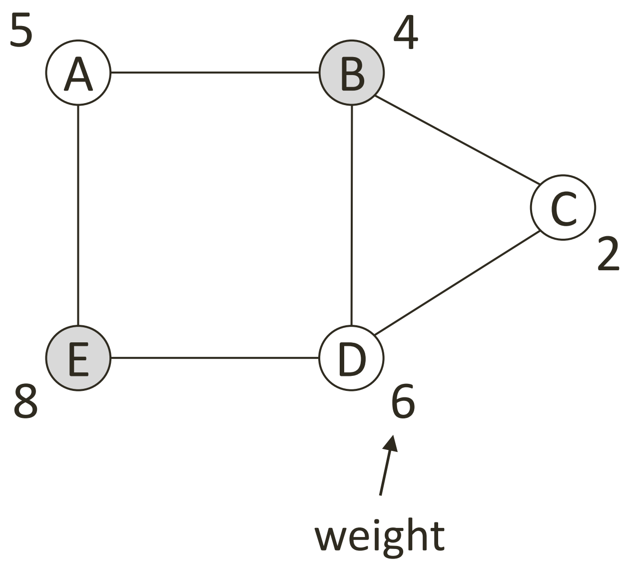
Maximum Independent Set Problem:

- Find independent set with maximum weight



BDD representing all independent sets of the graph

BDDs can Represent Optimization Problems

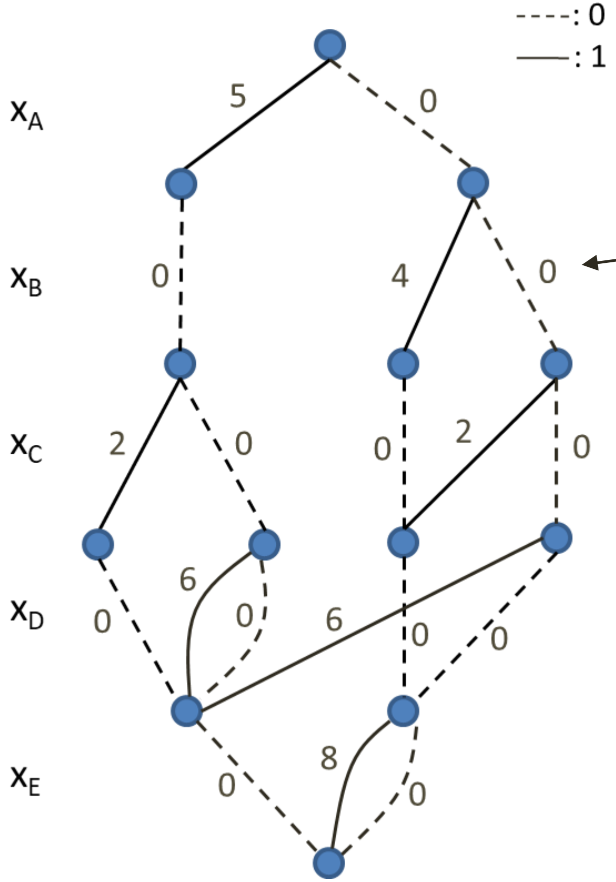


Independent set in a graph:

- Subset of non-adjacent vertices

Maximum Independent Set Problem:

- Find independent set with maximum weight



length of 0-arc: 0

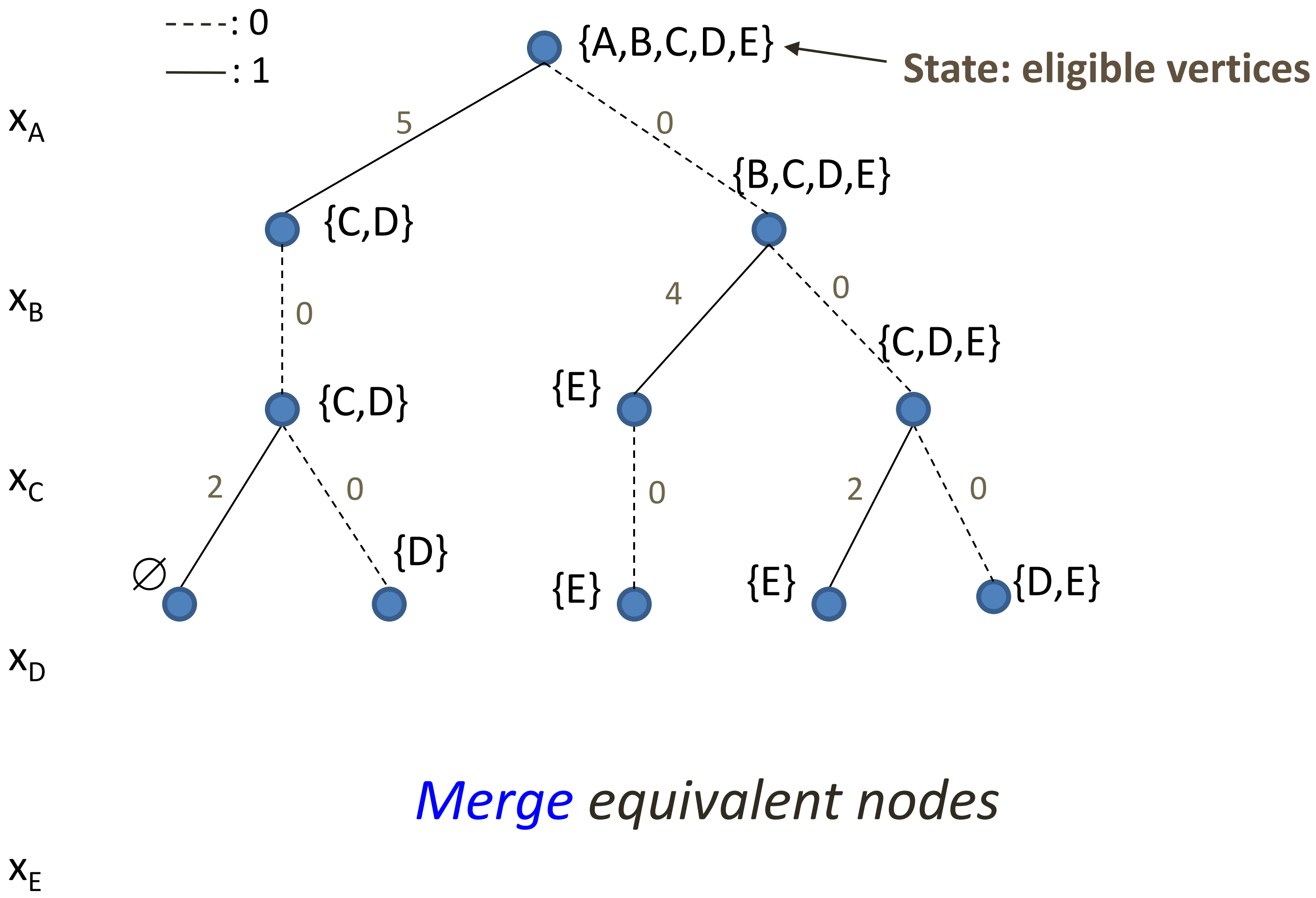
length of 1-arc: vertex weight

Maximum independent set: Longest root-to-terminal path in the BDD

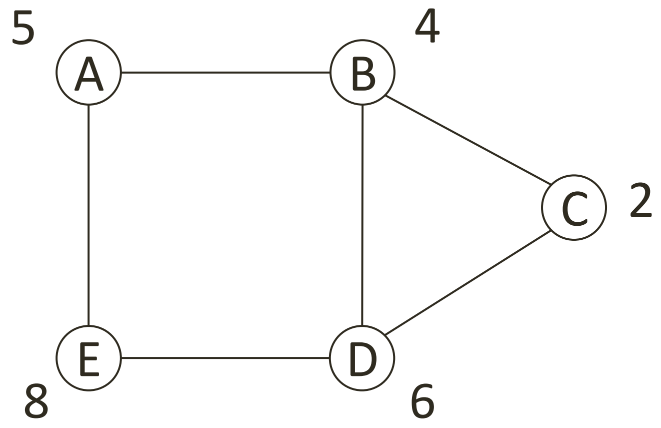
An Optimization Solver Based on Decision Diagrams

- Input: problem specification (e.g., weighted graph)
- Model: state-based formulation (e.g., dynamic program)
  - Question: How to compile the BDD? We need states and transitions.
- Solver: DD-based branch-and-bound
  - systematic search procedure (to prove optimality)
  - decision diagram provides upper and lower bounds
  - decision diagram defines the search method

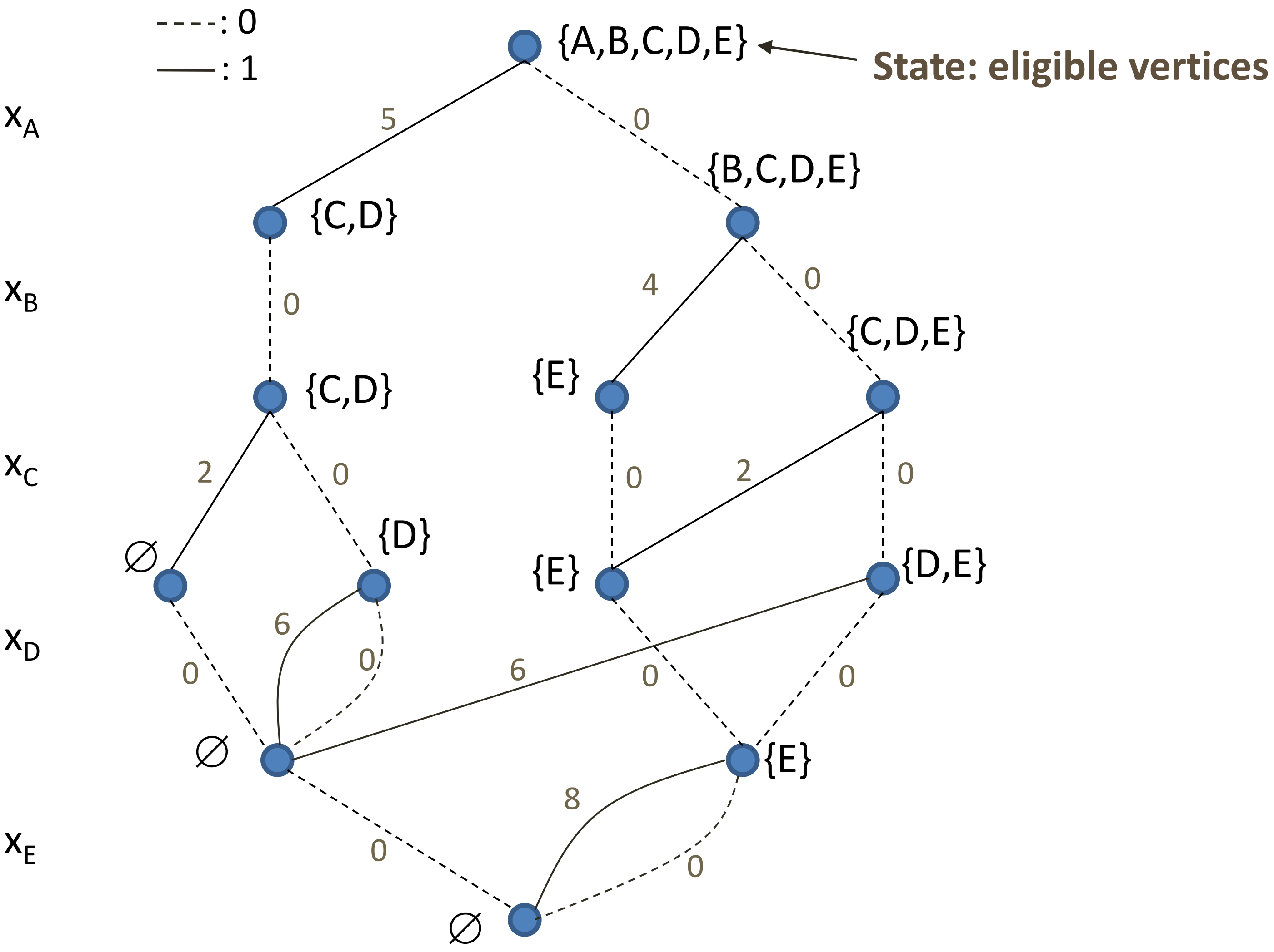
BDD Compilation for Maximum Independent Set

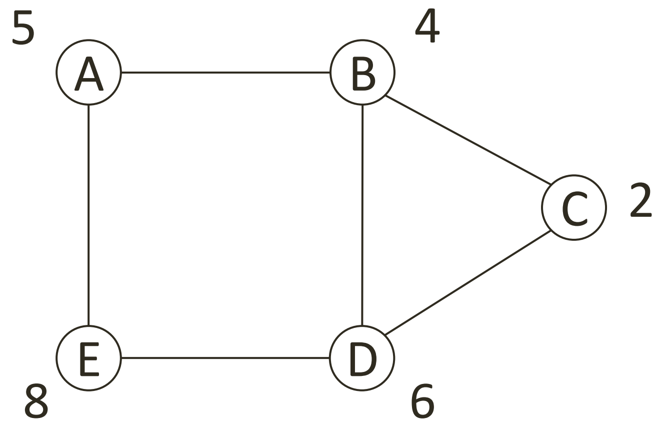


Merge equivalent nodes



BDD Compilation for Maximum Independent Set



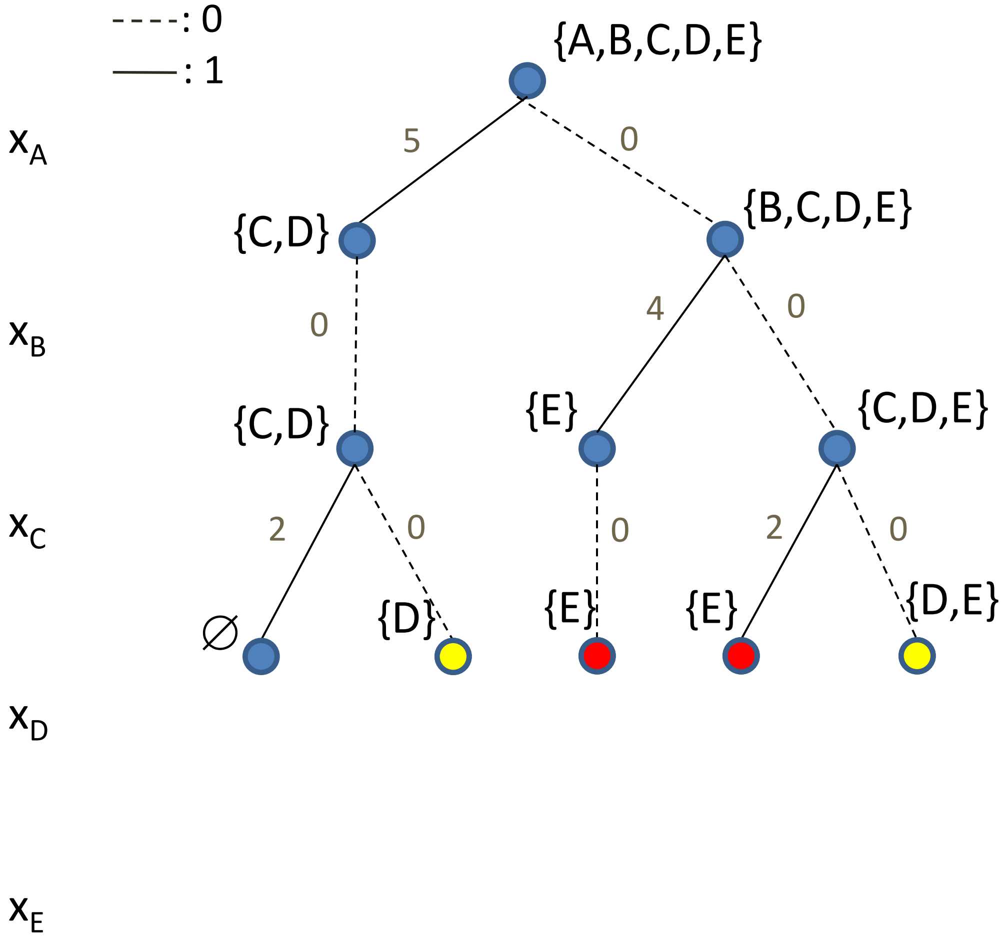


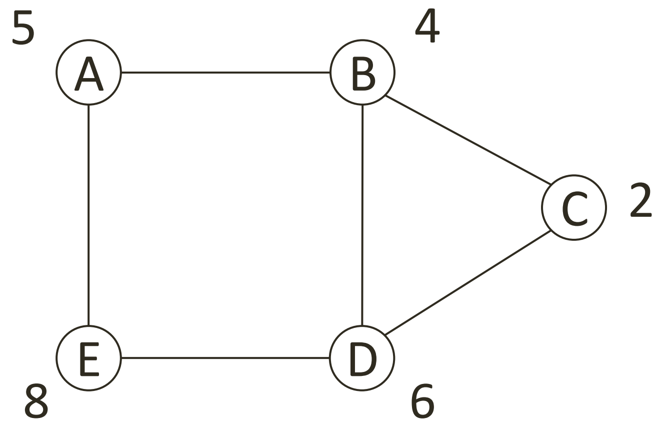
Theorem: This top-down compilation procedure generates a reduced exact BDD

[Bergman, Cire, vH, Hooker, IJOC 2014]

Optimal solution: Longest path

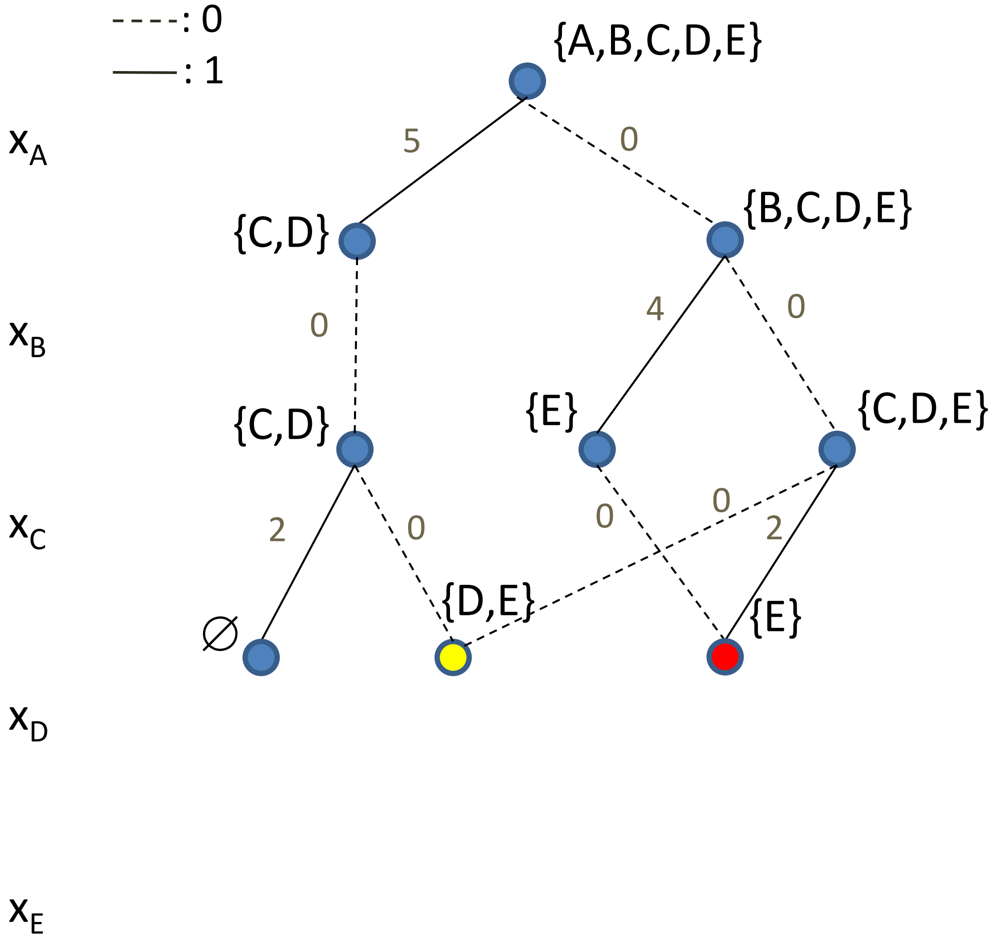
Independent Set Problem: Relaxed DD

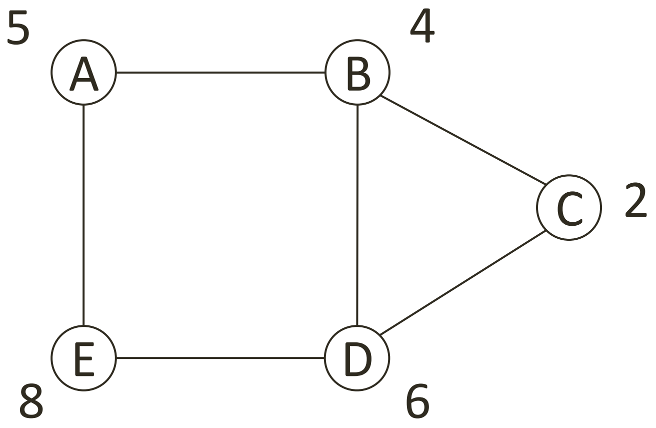




Maximum width = 3

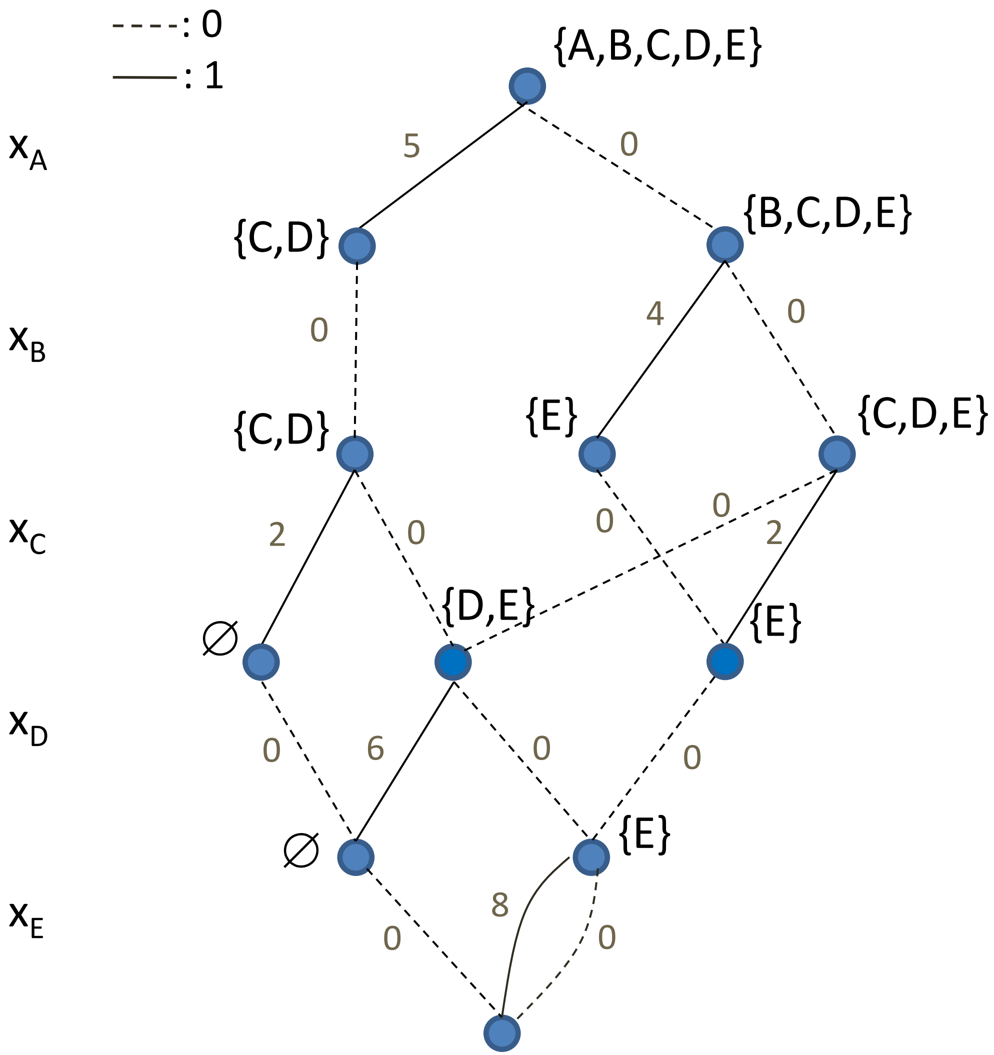
Independent Set Problem: Relaxed DD

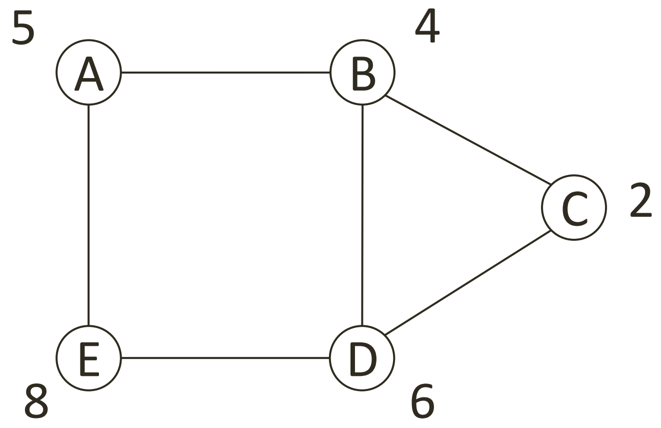




Maximum width = 3

Independent Set Problem: Relaxed DD

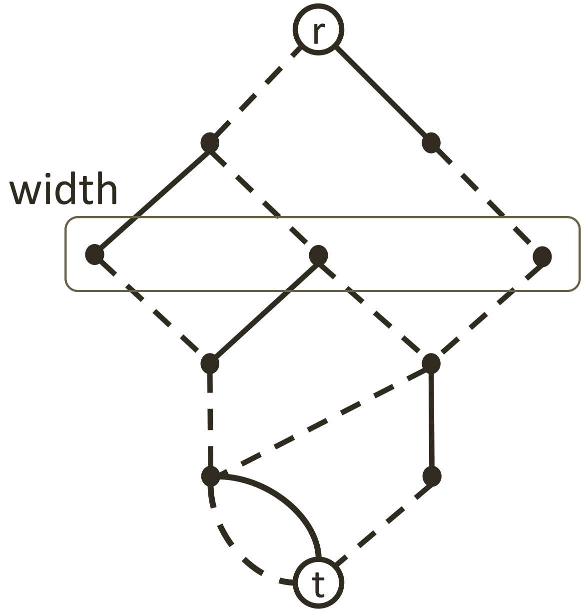




Maximum width = 3

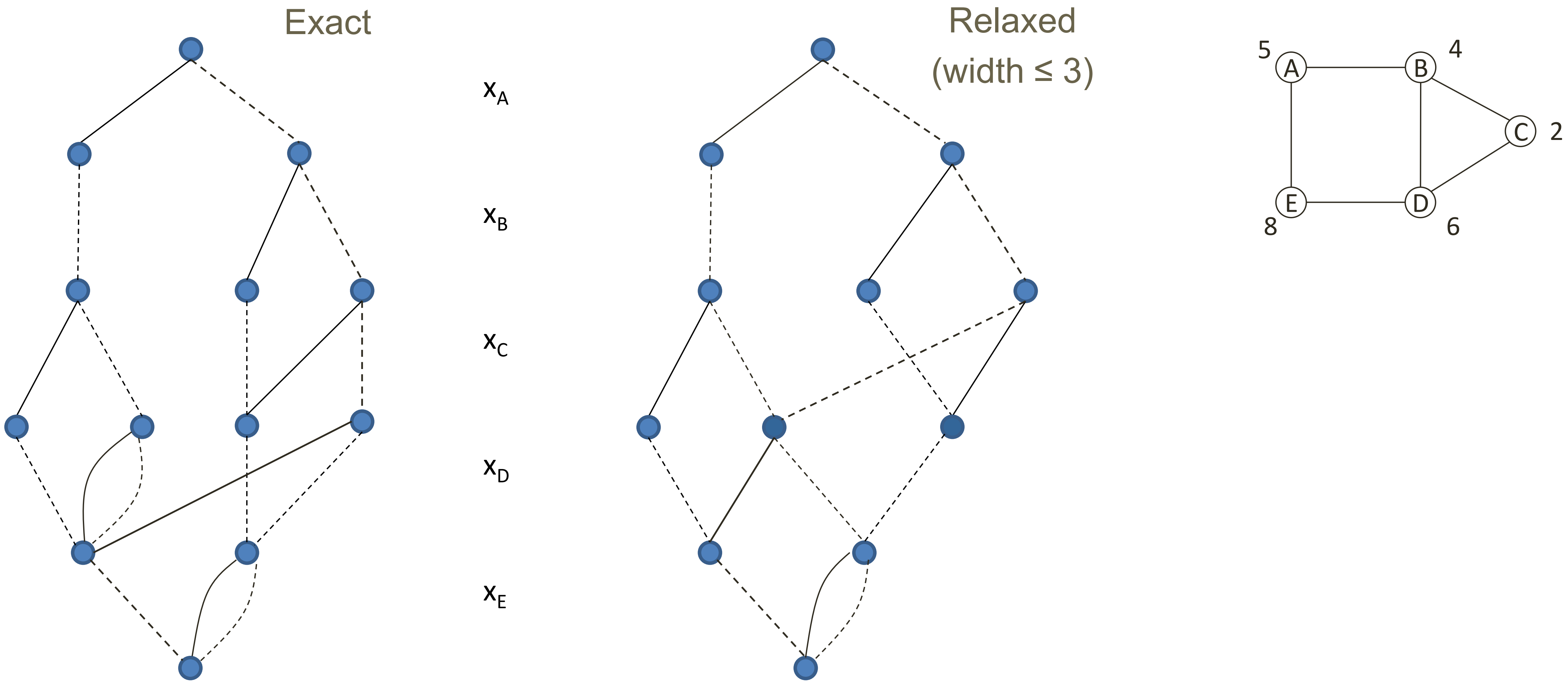
Relaxed Decision Diagrams: Polynomial Size

- Exponential size is handled by explicitly limiting the size (e.g., the width) of the diagram
- Merge non-equivalent nodes
  - define *node merging rule* to safely aggerate states
  - for independent set: take the *union* of the states
- Requirement: no solution is lost
  - over-approximation of the solution space
  - provides *discrete relaxation*
  - strength is controlled by the maximum width

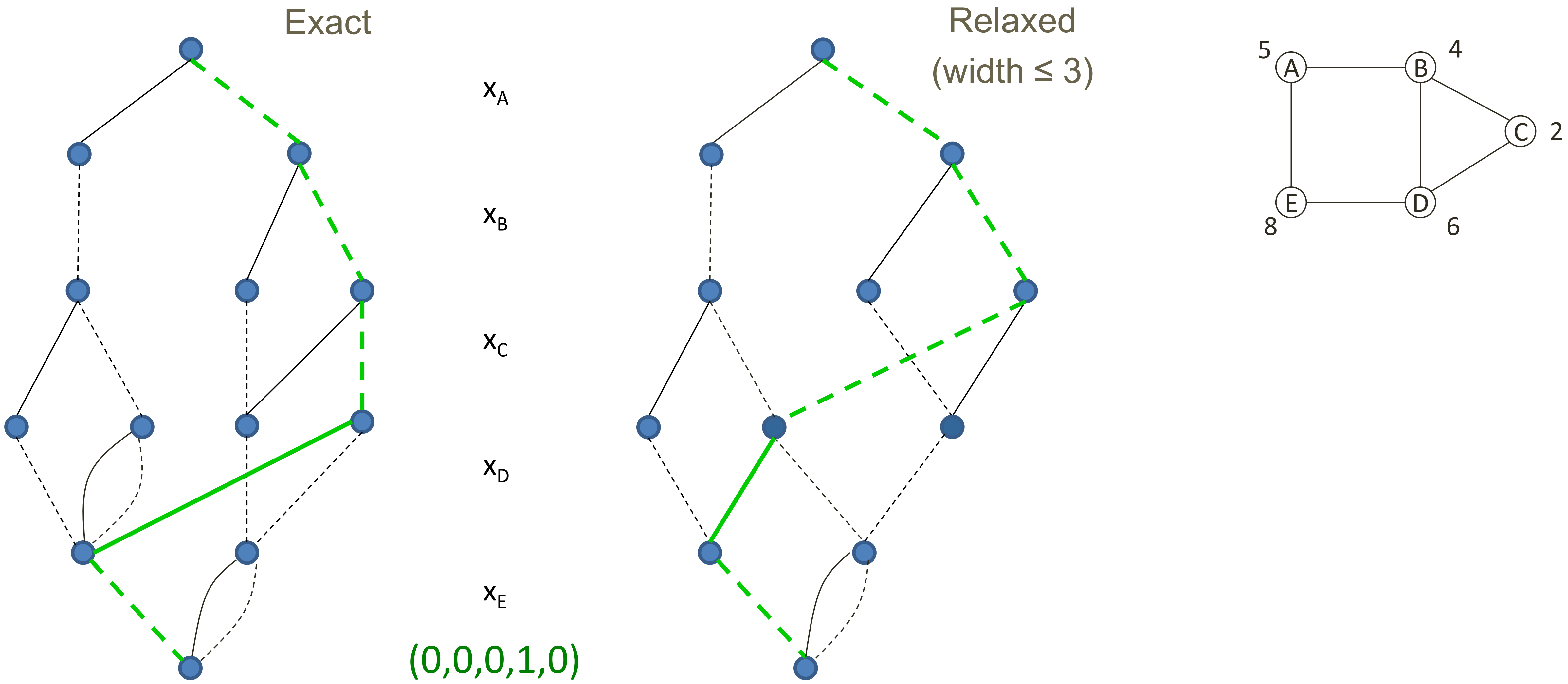


[Andersen, Hadzic, Hooker, Tiedemann, CP 2007]
[Bergman, Cire, vH, Hooker, CPAIOR 2011, IJOC 2016]

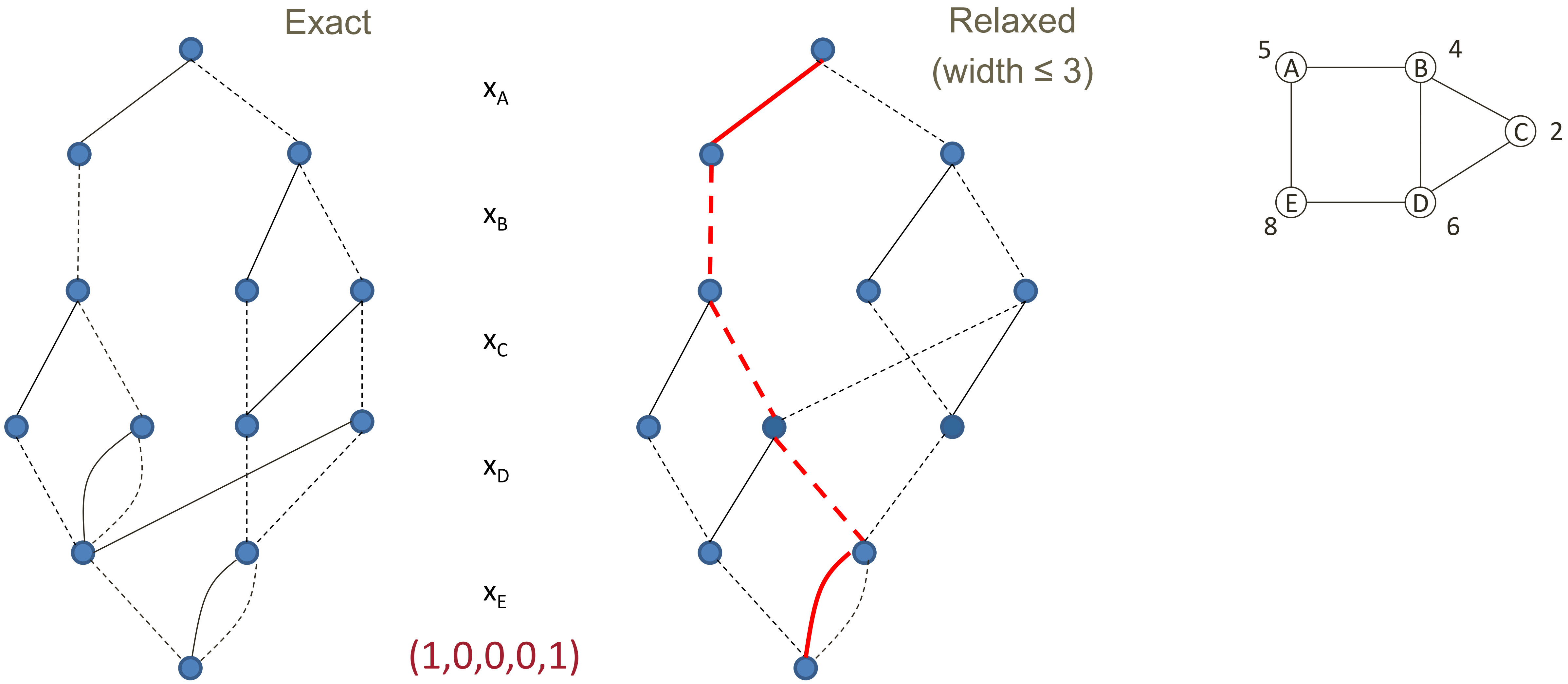
Exact vs. Relaxed Decision Diagrams



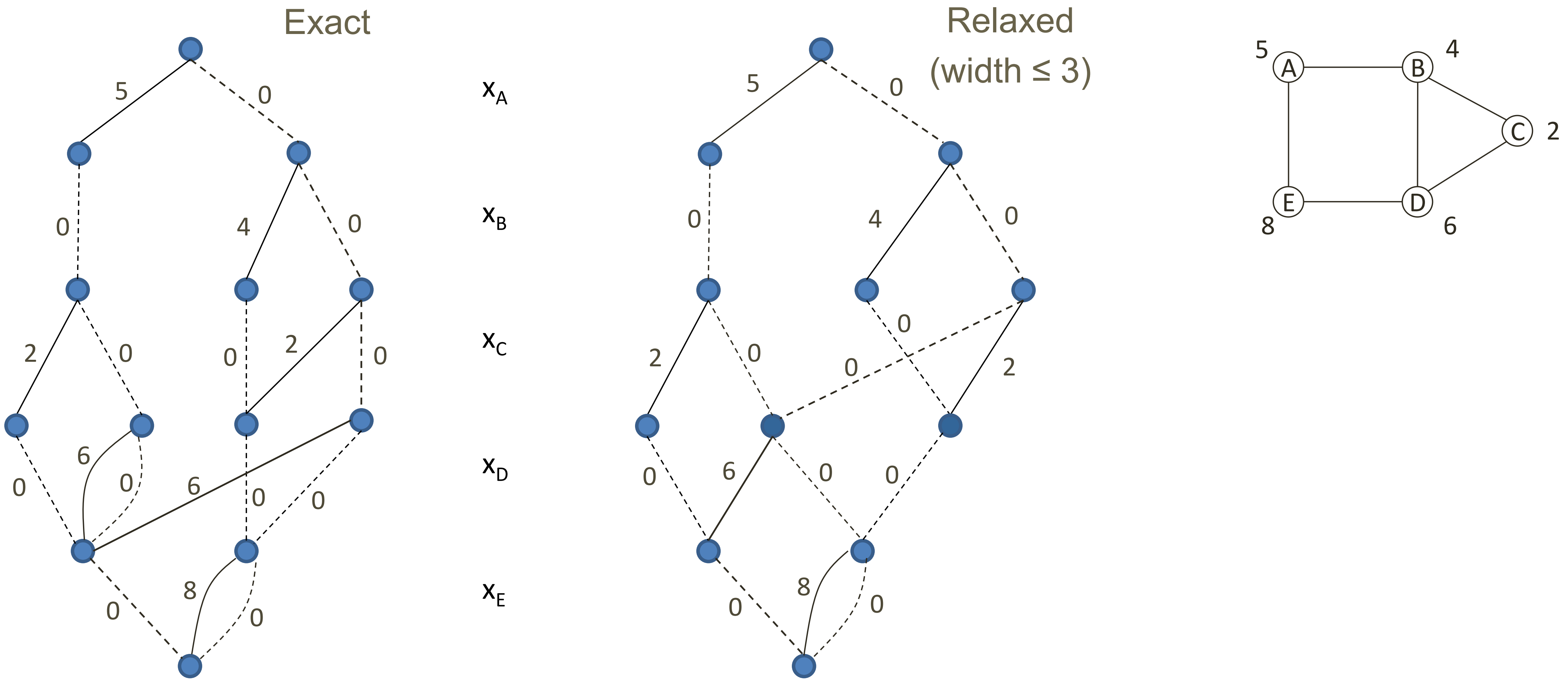
Exact vs. Relaxed Decision Diagrams



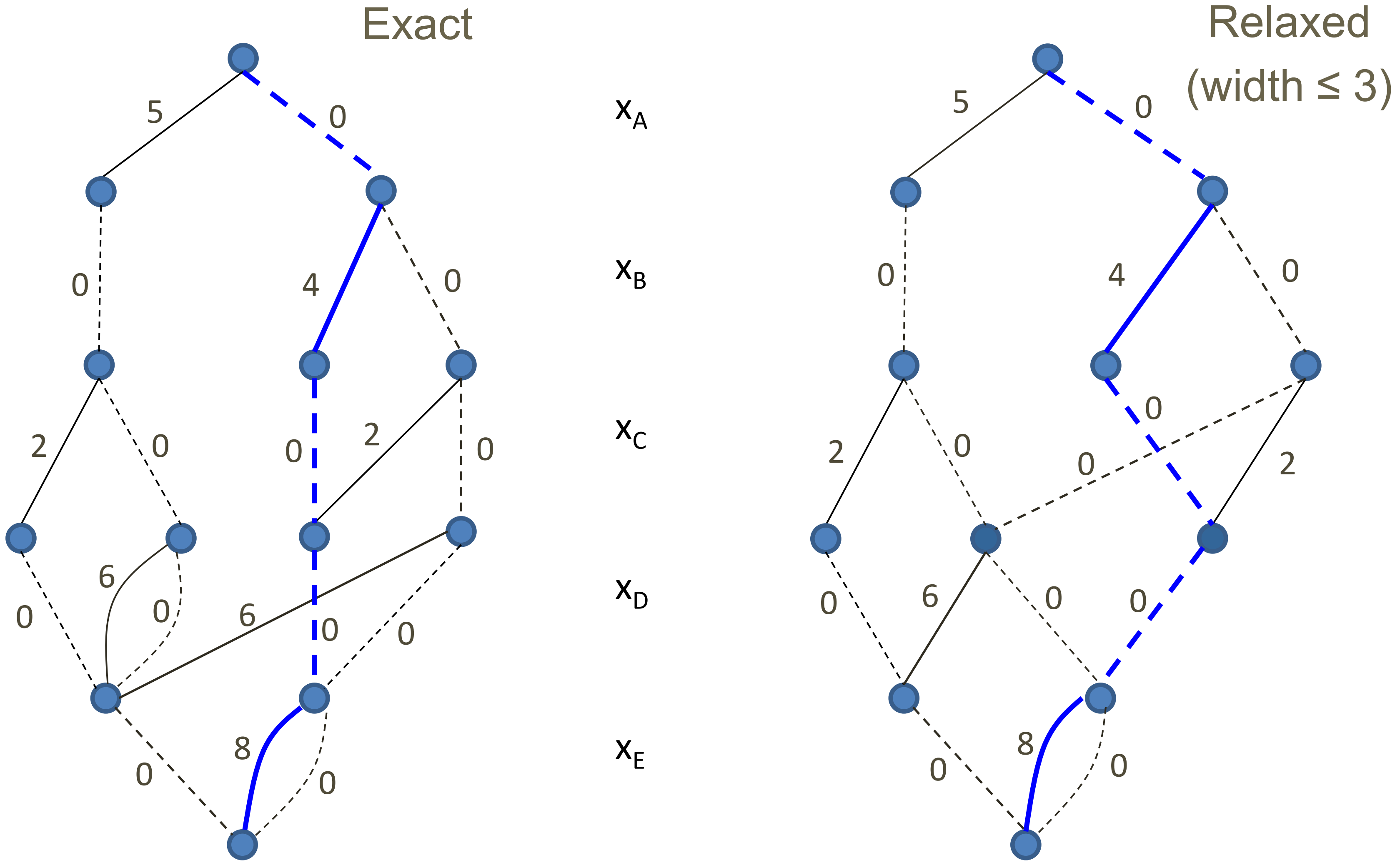
Exact vs. Relaxed Decision Diagrams

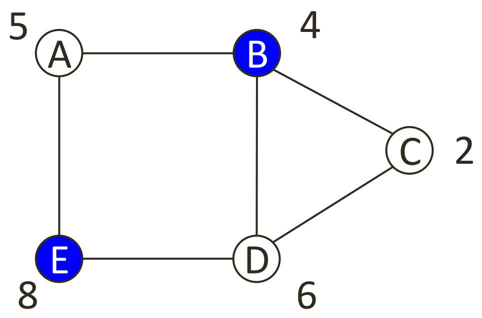


Exact vs. Relaxed Decision Diagrams



Exact vs. Relaxed Decision Diagrams

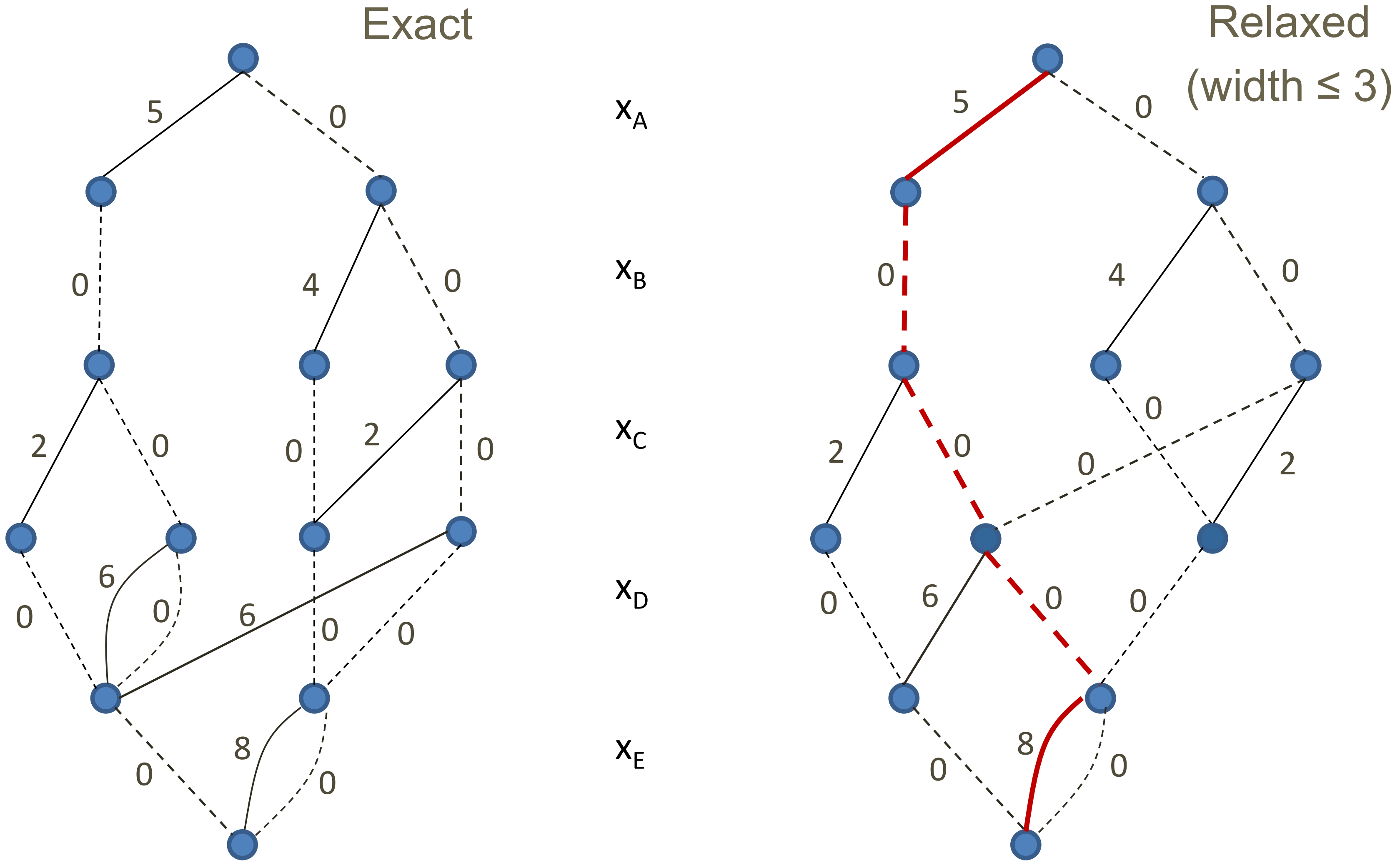


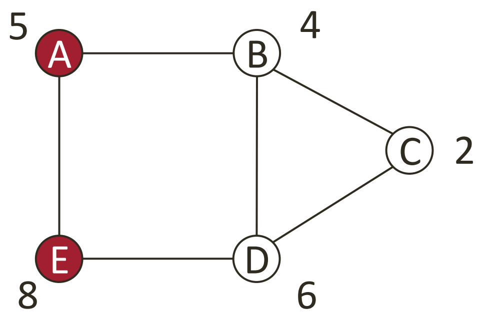


x = (0, 1, 0, 0, 1)

Solution value = 12

Exact vs. Relaxed Decision Diagrams

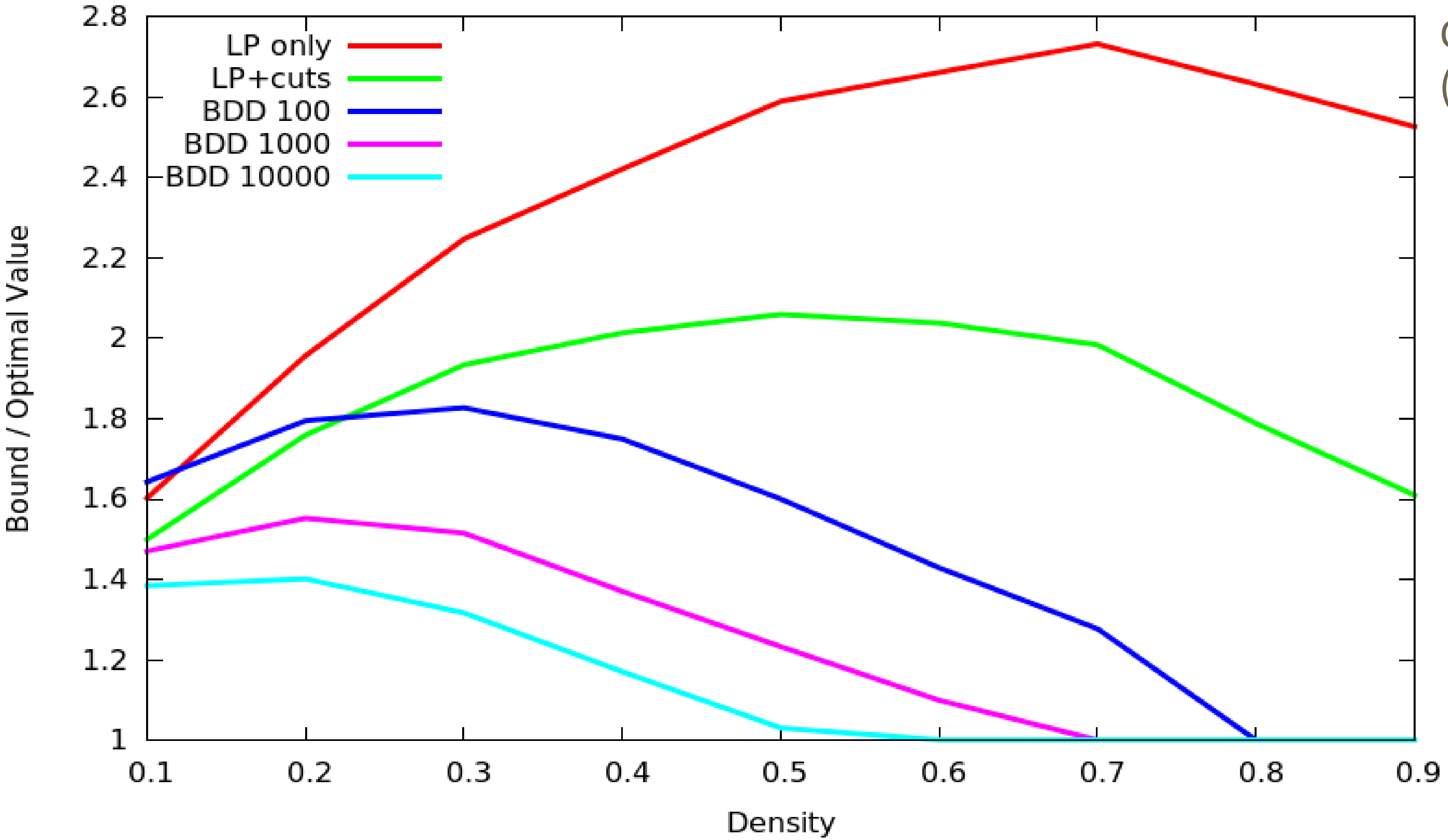




$x = (1, 0, 0, 0, 1)$

Upper bound = 13

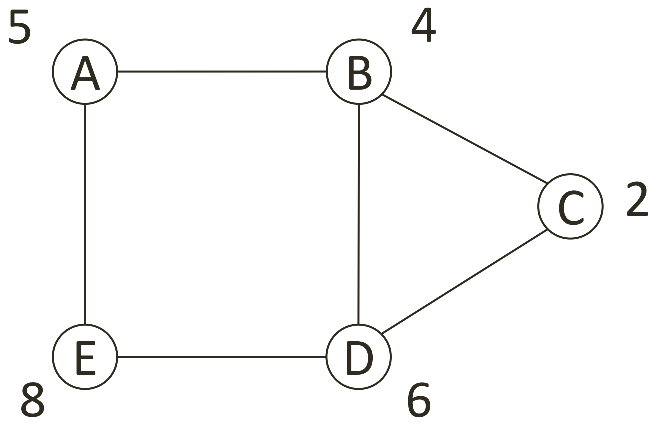
Relaxation Bound: Independent Set



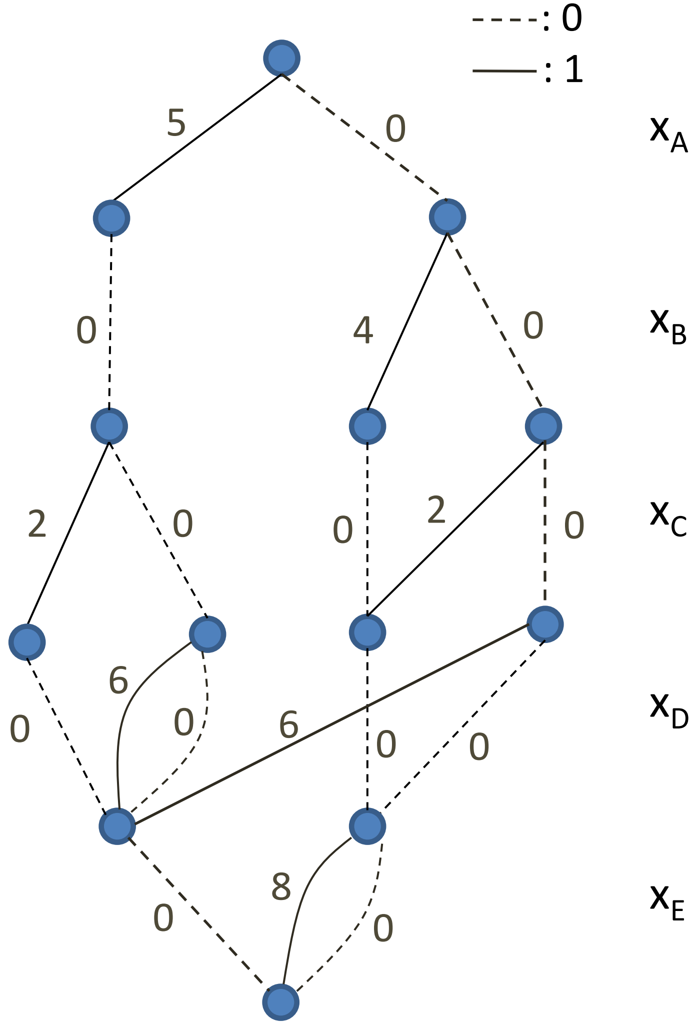
Compare with LP bound (CPLEX)

Restricted Decision Diagrams

- Under-approximation of the feasible set



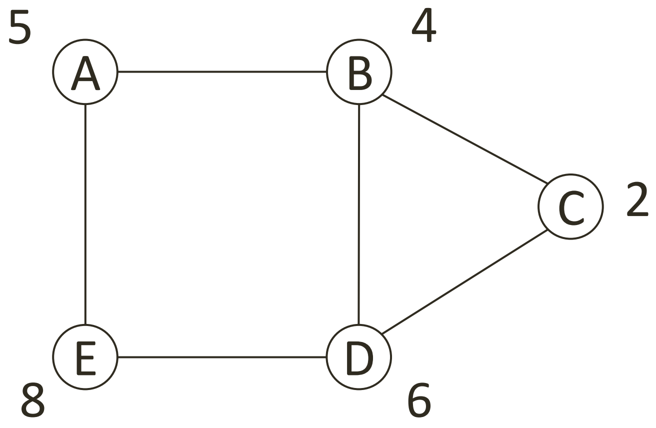
Maximum width = 3



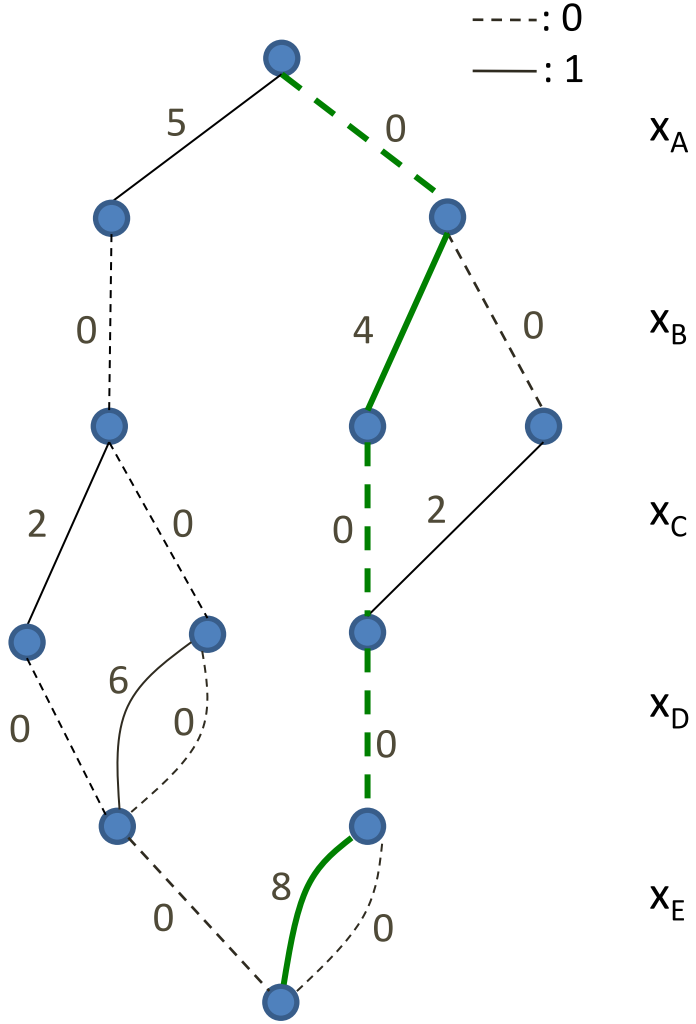
[Bergman, Cire, vH, Yunes, J Heur. 2014]

Restricted Decision Diagrams

- Under-approximation of the feasible set



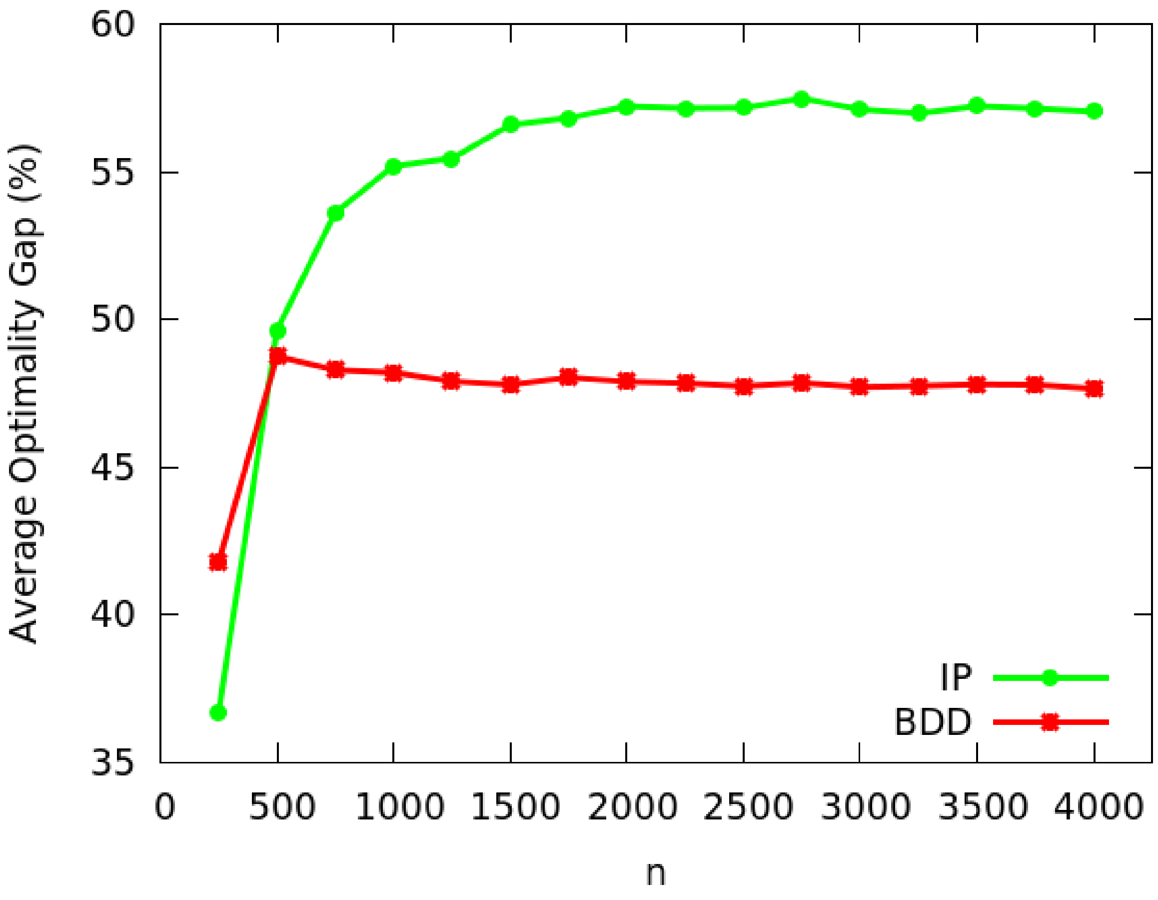
Maximum width = 3

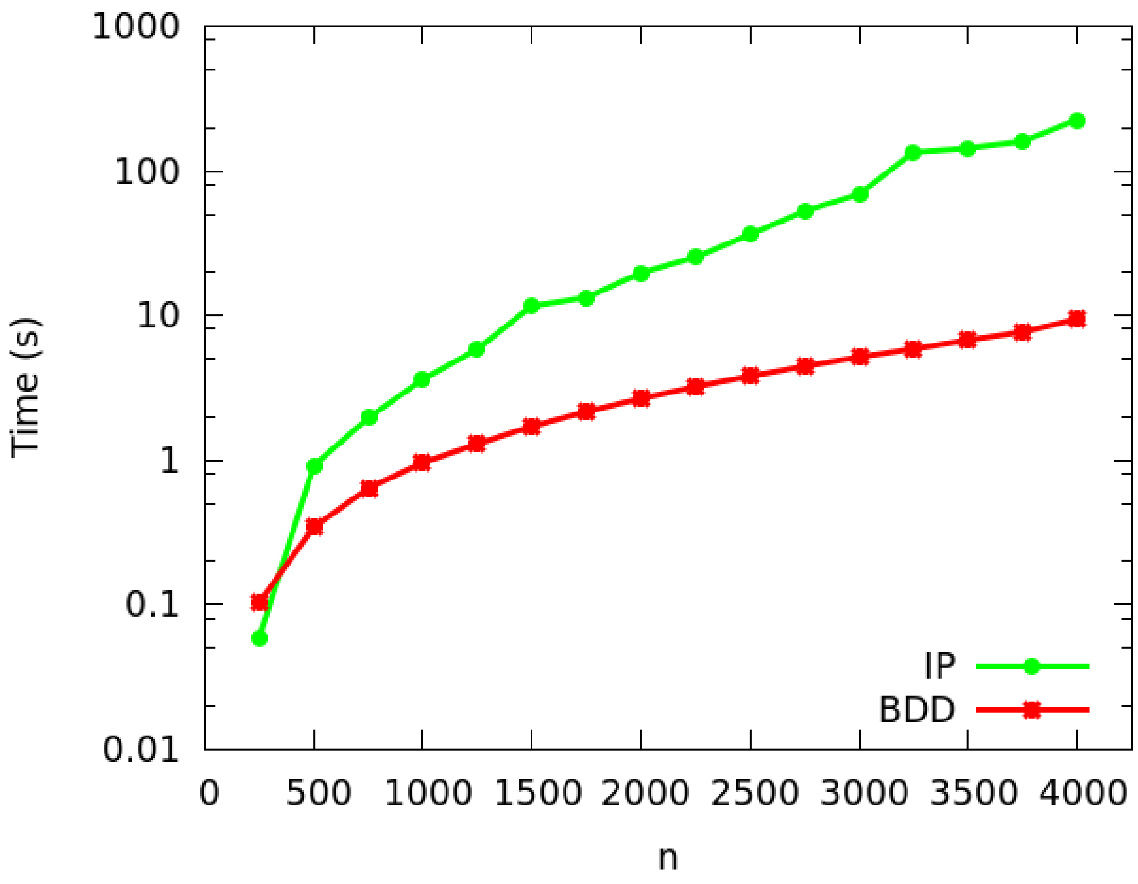


x = (0, 1, 0, 0, 1)
Lower bound = 12

[Bergman, Cire, vH, Yunes, J Heur. 2014]

Heuristic Bound: Set Covering Problem





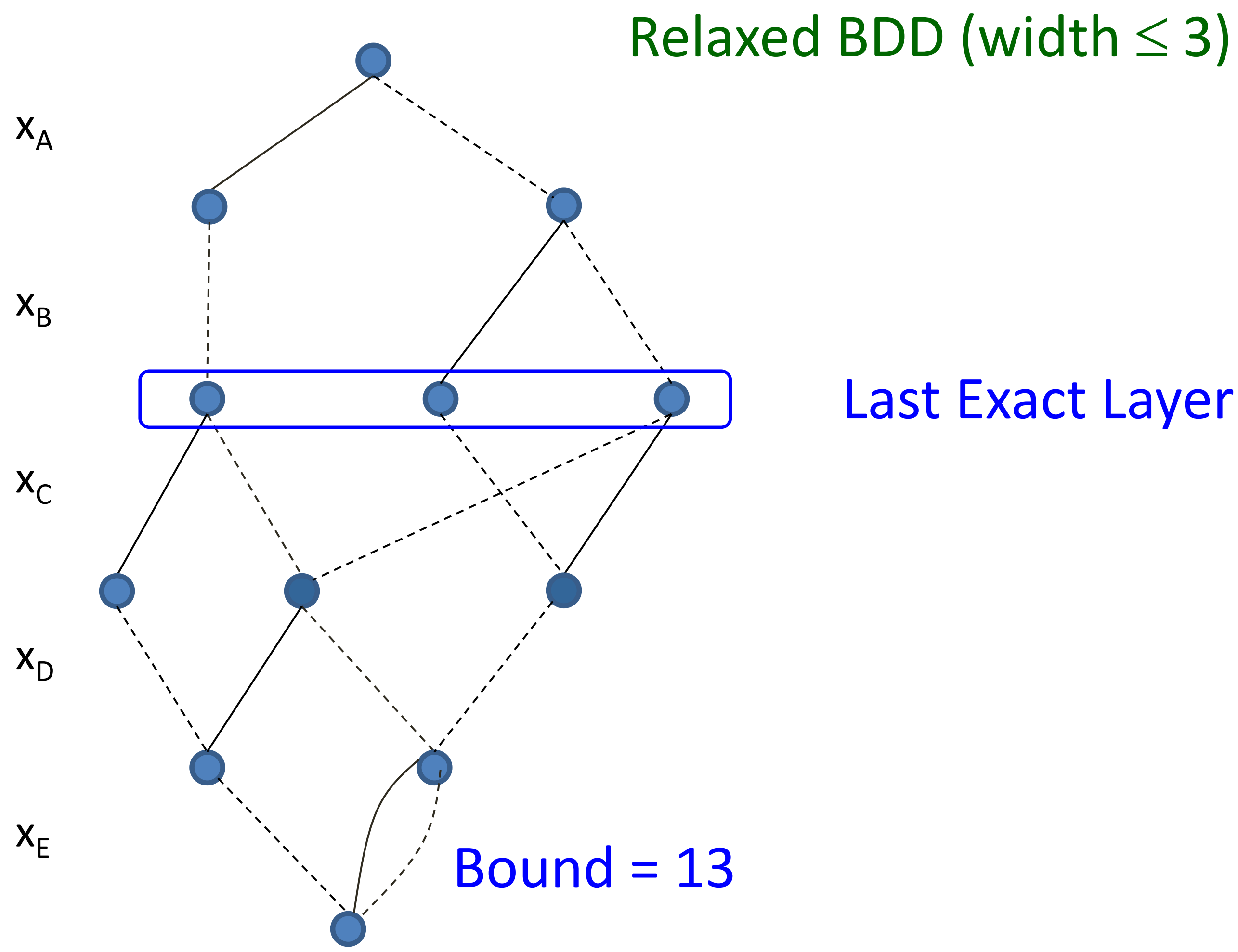
Exact Search Method

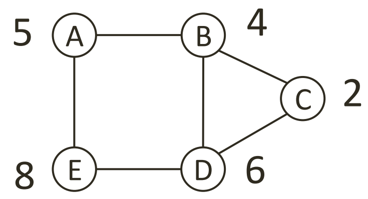
- Novel decision diagram branch-and-bound scheme
  - Relaxed diagrams play the role of the LP relaxation
  - Restricted diagrams are used as primal heuristics

- Branching is done on the *nodes* of the diagram
  - Branching on pools of partial solutions
  - Eliminates some search symmetry
  - No need to backtrack!

[Bergman, Cire, vH, Hooker, IJOC 2016]

Branch and Bound

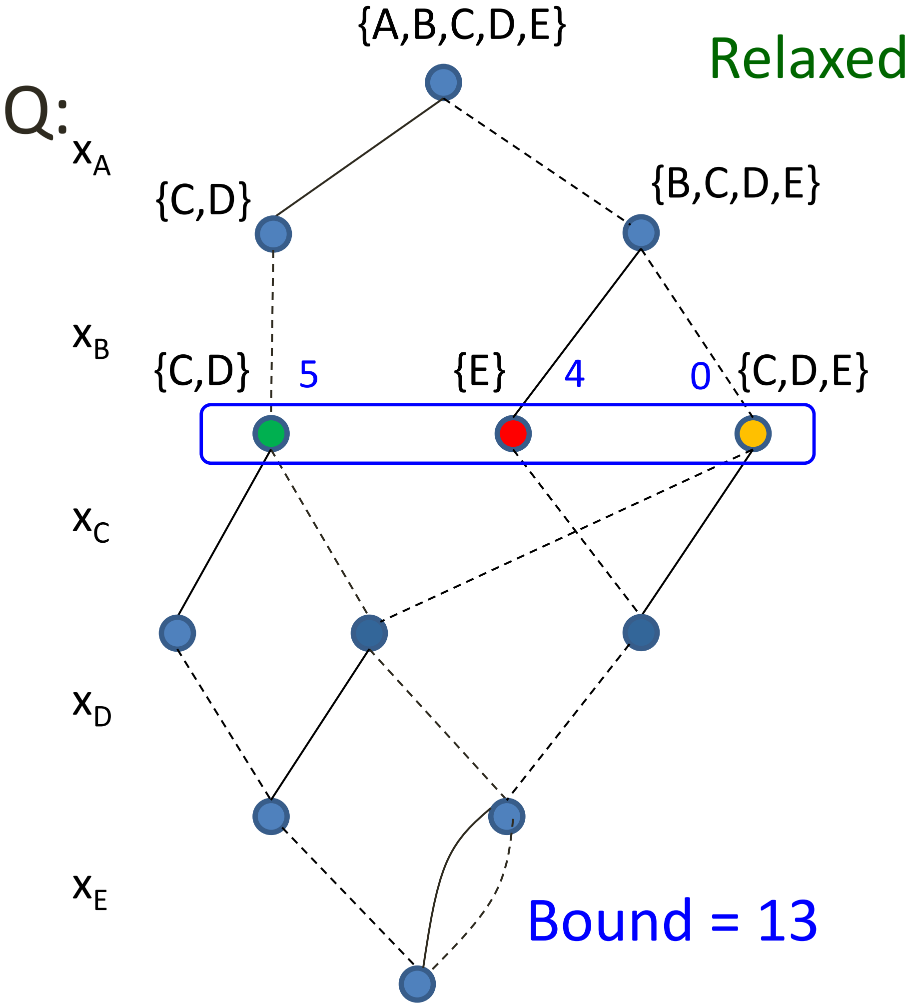




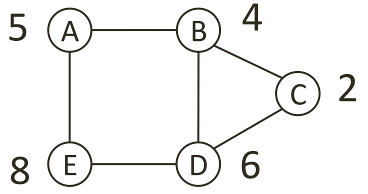
Node Queue

Relaxed BDD (width ≤ 3)

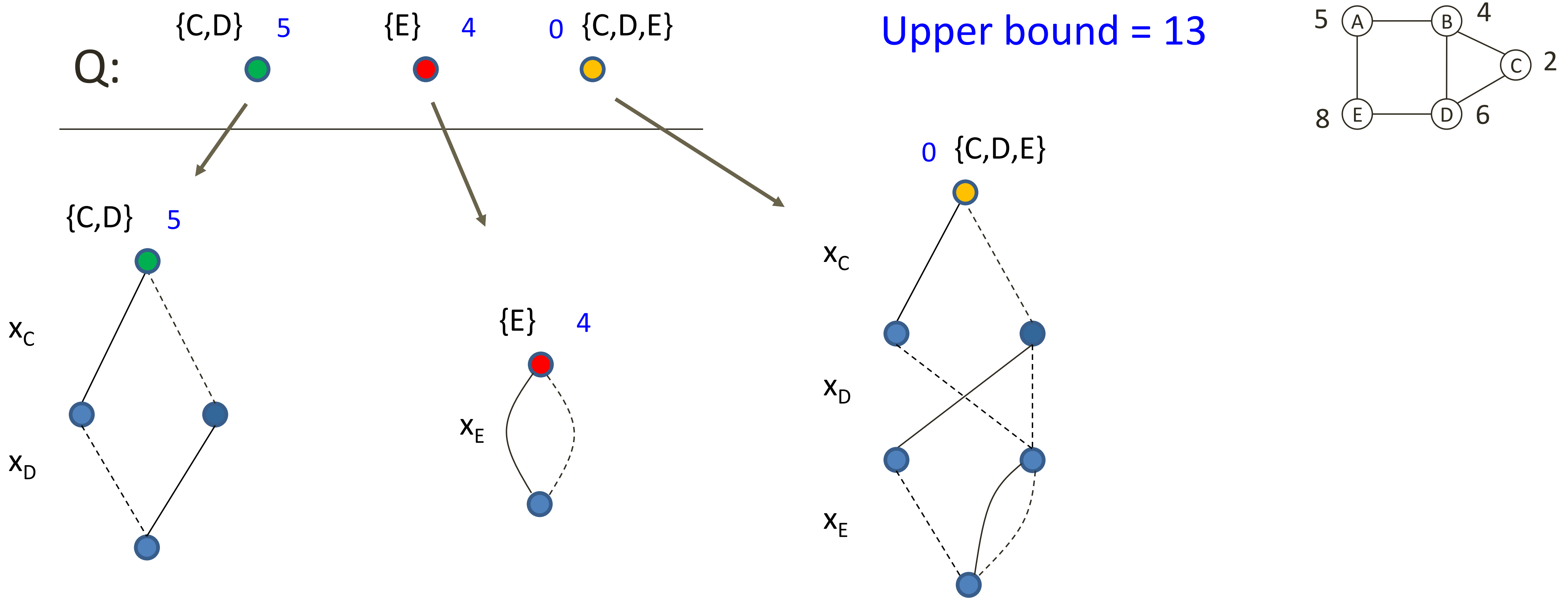
Upper bound = 13



Last Exact Layer



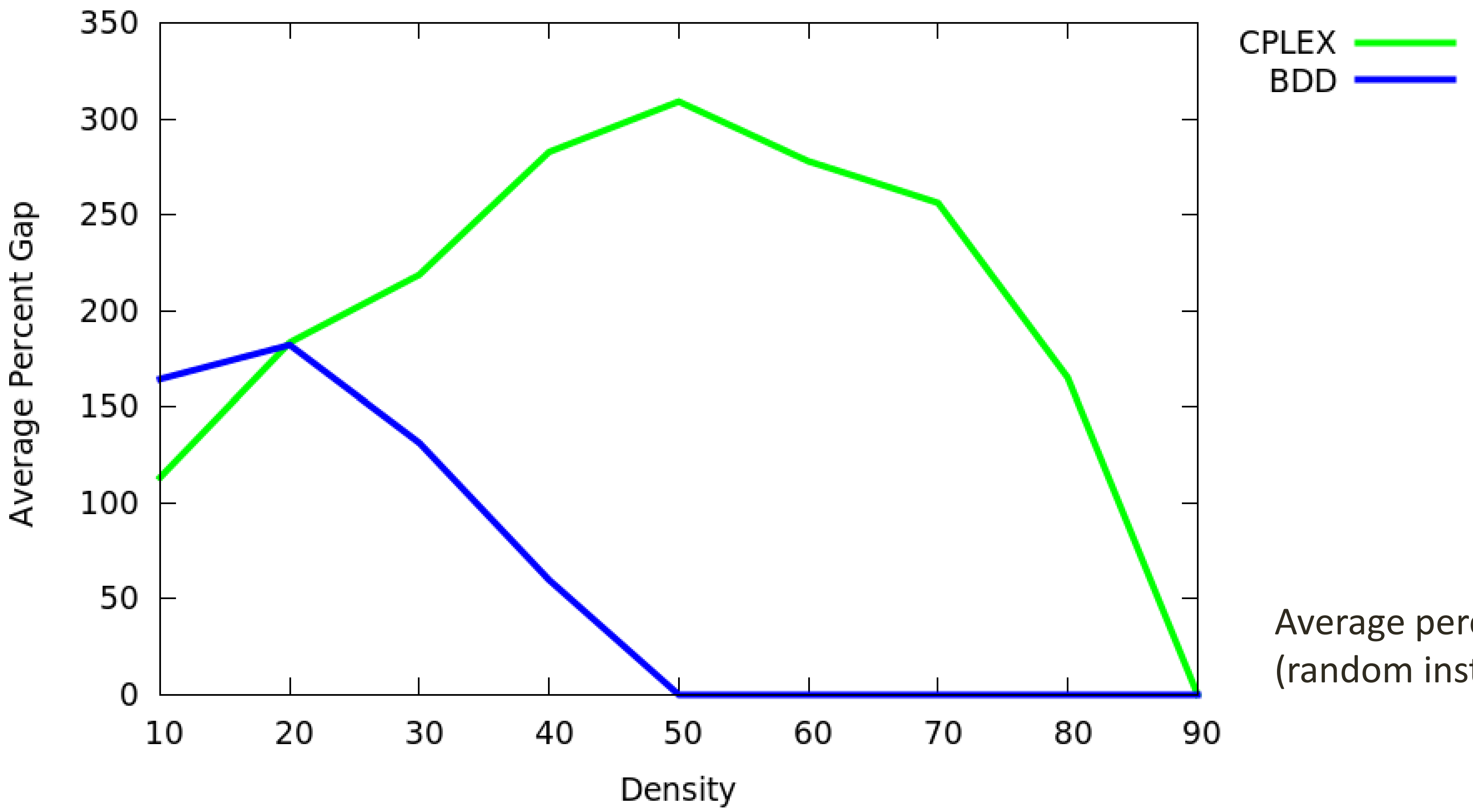
Node Queue



Exact solution: 11 Exact solution: 12 Exact solution: 10

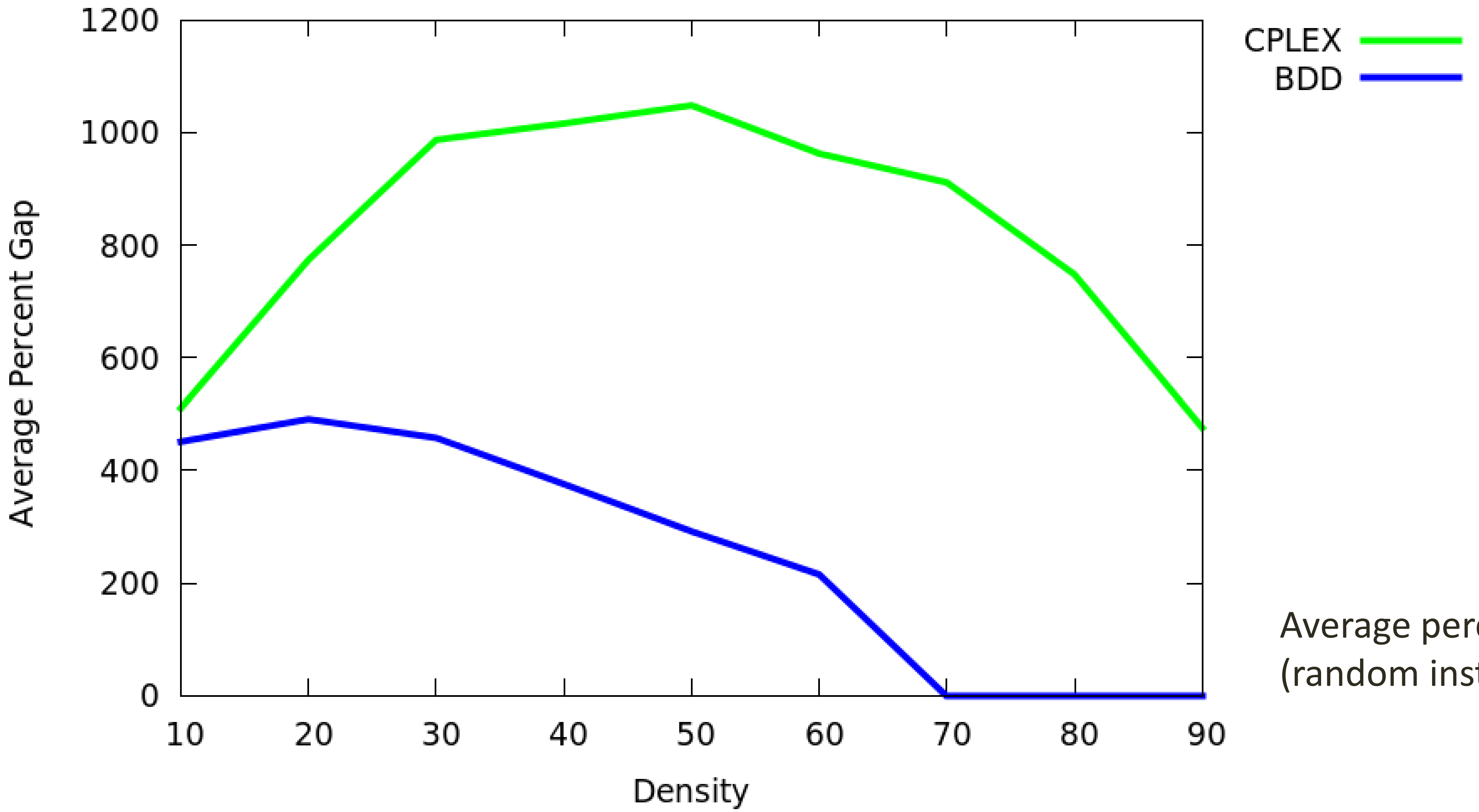
Optimal solution: 12

Maximum Independent Set: 500 variables



Average percent optimality gap (random instances) with 1h time limit

Maximum Independent Set: 1500 variables

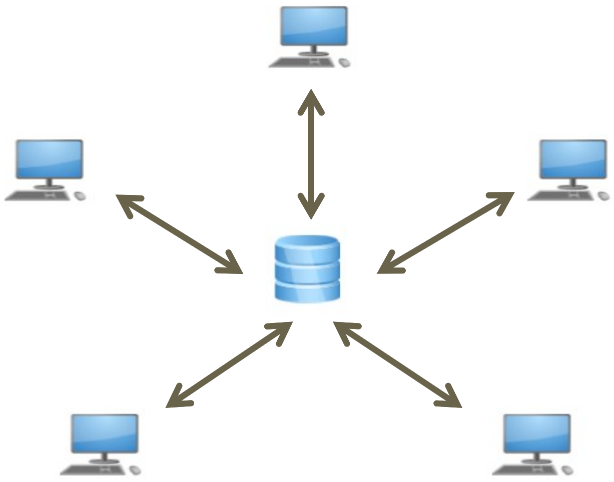


Average percent optimality gap (random instances) with 1h time limit

Maximum Cut Problem: BiqMac vs BDD

	BiqMac		BDD		Best known (2015)	
instance	LB	UB	LB	UB	LB	UB
g50	5880	5988.18	5880	5899*	5880	5988.18
g32	1390	1567.65	1410*	1645	1398	1560
g33	1352	1544.32	1380*	1536*	1376	1537
g34	1366	1546.70	1376*	1688	1372	1541
g11	558	629.17	564	567*	564	627
g12	548	623.88	556	616*	556	621
g13	578	647.14	580	652	580	645

Parallelization: Centralized Architecture



Master maintains a **pool** of BDD nodes to process

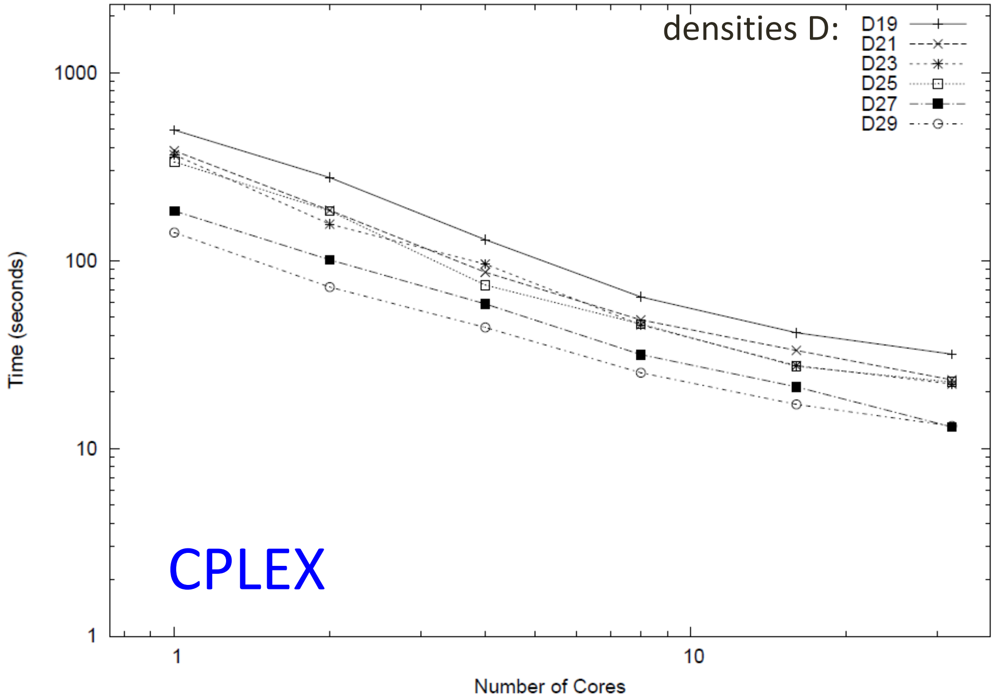
- nodes with larger upper bound have higher priority

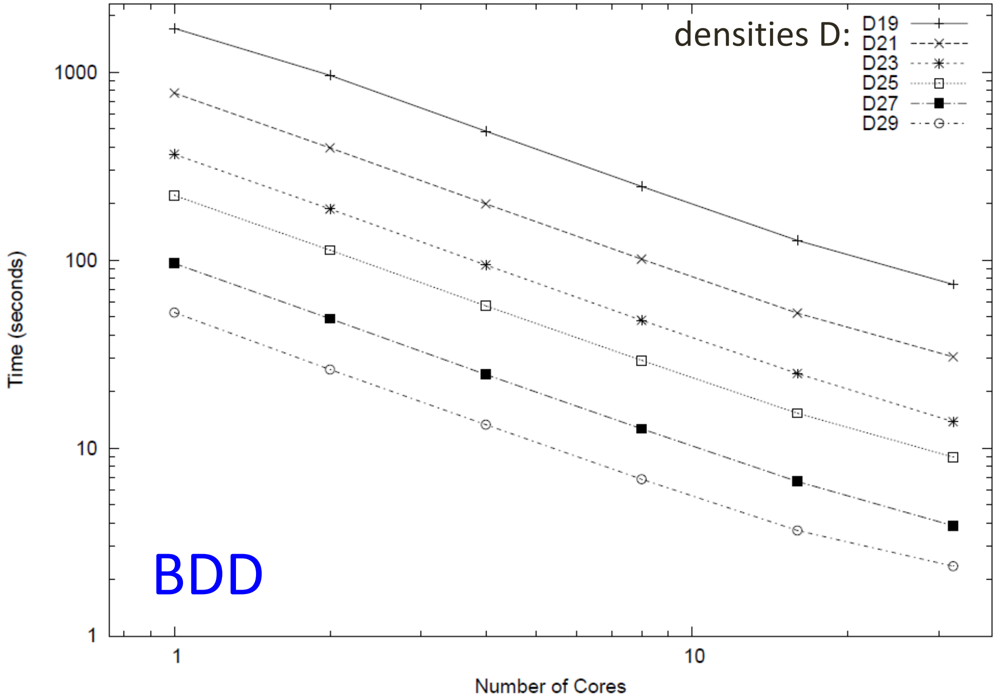
Workers receive BDD nodes, generate *restricted* & *relaxed* BDDs, and send new BDD nodes and bounds to master

- they also maintain a **local pool** of nodes

[Bergman et al. CPAIOR 2014]

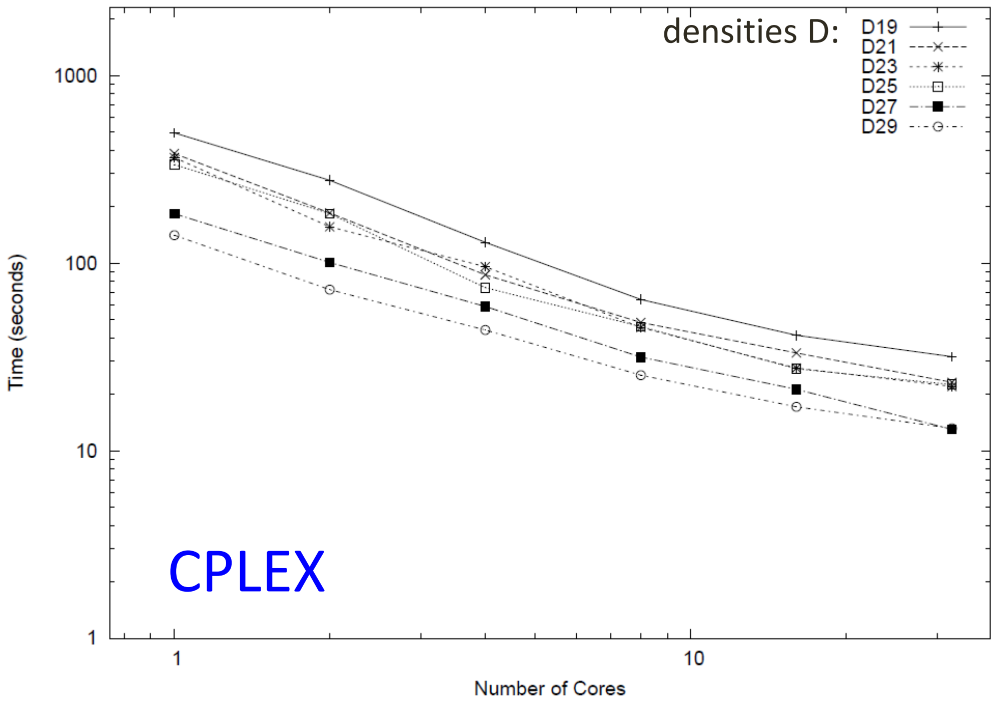
Parallelization: BDD vs CPLEX

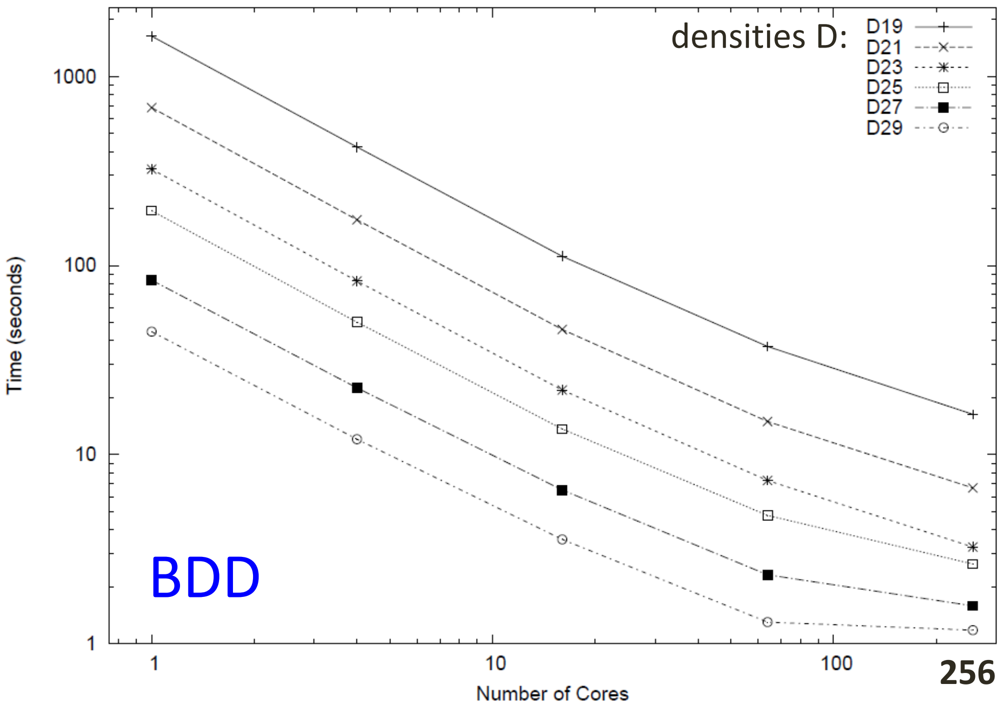




- n = 170, each data point avg over 30 maximum independent set instances
- 1 worker: BDD 1.25 times faster than CPLEX (for density 0.29)
- 32 workers: BDD 5.5 times faster than CPLEX (for density 0.29)

Parallelization: BDD vs CPLEX





256 cores

- n = 170, each data point avg over 30 maximum independent set instances
- 1 worker: BDD 1.25 times faster than CPLEX (for density 0.29)
- 32 workers: BDD 5.5 times faster than CPLEX (for density 0.29)
- BDDs scale well to (at least) 256 workers

Other Applications of Relaxed Decision Diagrams

Excellent survey paper:
Castro, Cire & Beck [IJOC 2022]

- Constraint Programming
  - DD-based constraint propagation
- Integer Linear and Nonlinear Programming
  - Cutting plane generation, addition of DD-bounds in MIP search tree, column ‘elimination’
- Sequencing, routing, and scheduling
  - State-of-the-art results on machine scheduling, TSPTW with side constraints, sequential ordering problem, …
- AI planning
  - Provide admissible heuristic for A* search

References

- Henrik Reif Andersen, Tarik Hadzic, John N. Hooker, Peter Tiedemann: A Constraint Store Based on Multivalued Decision Diagrams. CP 2007: 118-132
- David Bergman, Willem Jan van Hoeve, John N. Hooker: Manipulating MDD Relaxations for Combinatorial Optimization. CPAIOR 2011: 20-35
- David Bergman, André A. Ciré, Willem Jan van Hoeve, John N. Hooker: Optimization Bounds from Binary Decision Diagrams. INFORMS J. Comput. 26(2): 253-268 (2014)
- David Bergman, André A. Ciré, Willem Jan van Hoeve, Tallys H. Yunes: BDD-based heuristics for binary optimization. J. Heuristics 20(2): 211-234 (2014)
- David Bergman, André A. Ciré, Ashish Sabharwal, Horst Samulowitz, Vijay A. Saraswat, Willem Jan van Hoeve: Parallel Combinatorial Optimization with Decision Diagrams. CPAIOR 2014: 351-367
- David Bergman, André Augusto Ciré, Willem-Jan van Hoeve, John N. Hooker: Discrete Optimization with Decision Diagrams. INFORMS J. Comput. 28(1): 47-66 (2016)
- Margarita P. Castro, Andre A. Cire, J. Christopher Beck: Decision Diagrams for Discrete Optimization: A Survey of Recent Advances. INFORMS J. Comput. 34(4):2271-2295 (2022) https://doi.org/10.1287/ijoc.2022.1170

- David Bergman, André A. Ciré, Willem-Jan van Hoeve, John N. Hooker: Decision Diagrams for Optimization. Springer 2016, ISBN 978-3-319-42847-5 https://link.springer.com/book/10.1007/978-3-319-42849-9