Discrete Optimization with Decision Diagrams

J. N. Hooker

Joint work with
David Bergman, André Ciré, Willem van Hoeve
Carnegie Mellon University

Univ. of Toronto, May 2013

Goal

- A **general discrete optimization method** that combines:
  - Relaxation and bounding of **mixed integer programming (MIP)**
  - Primal heuristics of **MIP**
  - Propagation of **constraint programming (CP)**
  - Recursive modeling style of **dynamic programming (DP)**
- How?
  - Use binary/multivalued **decision diagrams** as a framework.

Approach

- Decision diagram as a basic tool:
  - **Relaxed** decision diagrams provide **bounding** mechanism (MIP)
  - **Restricted** decision diagrams provide **primal heuristic** (MIP)
  - **Relaxed** decision diagrams enhanced **propagation** (CP)
  - **State variables** in decision diagram provide **DP modeling style**
- Plus…
  - D**ecision diagram** framework provides **novel branching scheme**.

Decision Diagrams

- **Binary decision diagrams (BDD**s) historically used for circuit design and verification.
  - Lee 1959, Akers 1978, Bryant 1986.

Decision Diagrams

- **Binary decision diagrams (BDD**s**)** historically used for circuit design and verification.
  - Lee 1959, Akers 1978, Bryant 1986.
- **Compact** graphical representation of **boolean** function.
  - Can also represent **feasible set** of problem with binary variables.
  - Slight generalization (MDDs) represents finite domain variables.

Decision Diagrams

- BDD is result of superimposing isomorphic subtrees in a search tree.
  - Unique reduced BDD for given variable ordering.
  - “Caching” is a popular theme in knowledge representation.
- Constraints need not have an inequality representation.

Binary Decision Diagrams

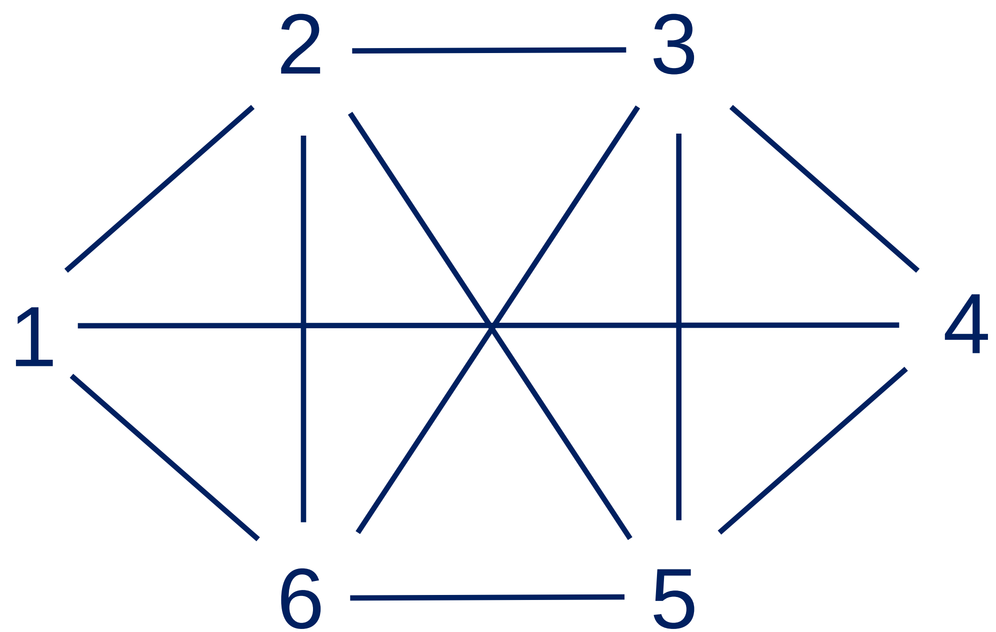
- BDD can grow exponentially with problem size.
  - So we use a smaller, **relaxed** BDD that represents **superset** of feasible set.
    - Andersen, Hadzic, Hooker, Tiedemann 2007.
    - For alldiff systems, reduced search tree from >1 million nodes to 1 node.
    - Subsequent papers with Hadzic, Hoda, van Hoeve, O'Sullivan.

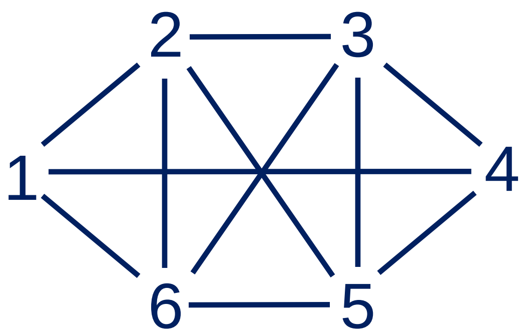
- We focus on **independent set problem** on a graph…

Independent Set Problem

Let each vertex have weight w_i

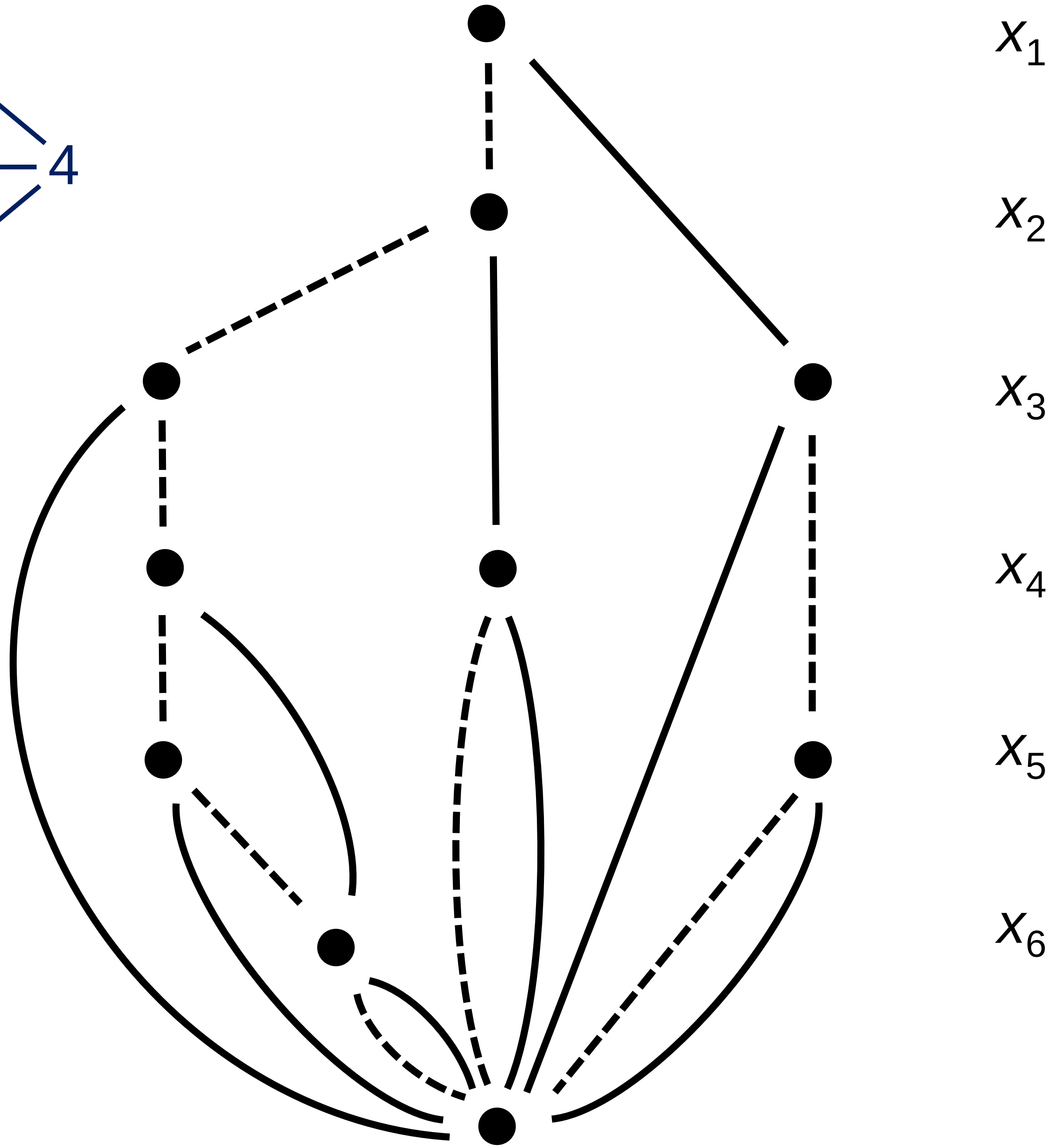
Select nonadjacent vertices to maximize $\sum_i w_i x_i$

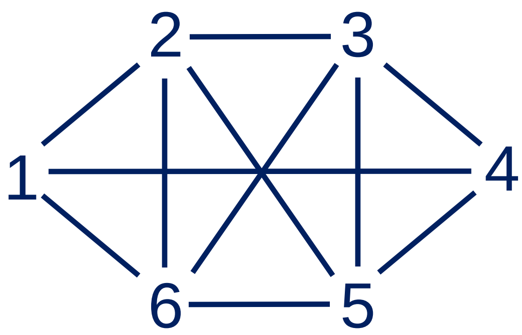




Exact BDD for independent set problem

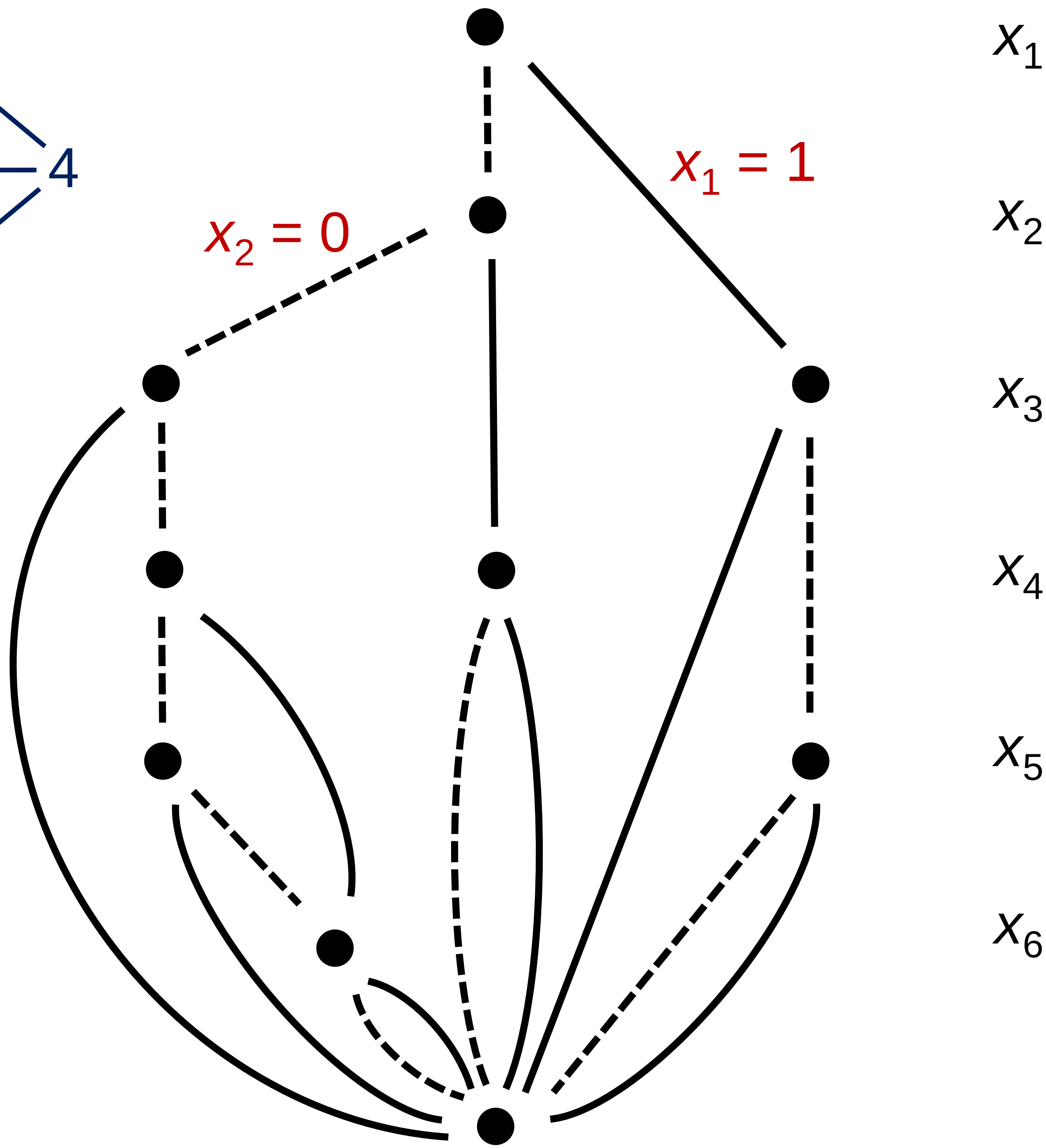
“zero-suppressed” BDD

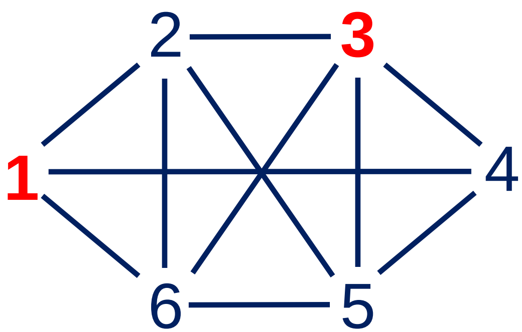




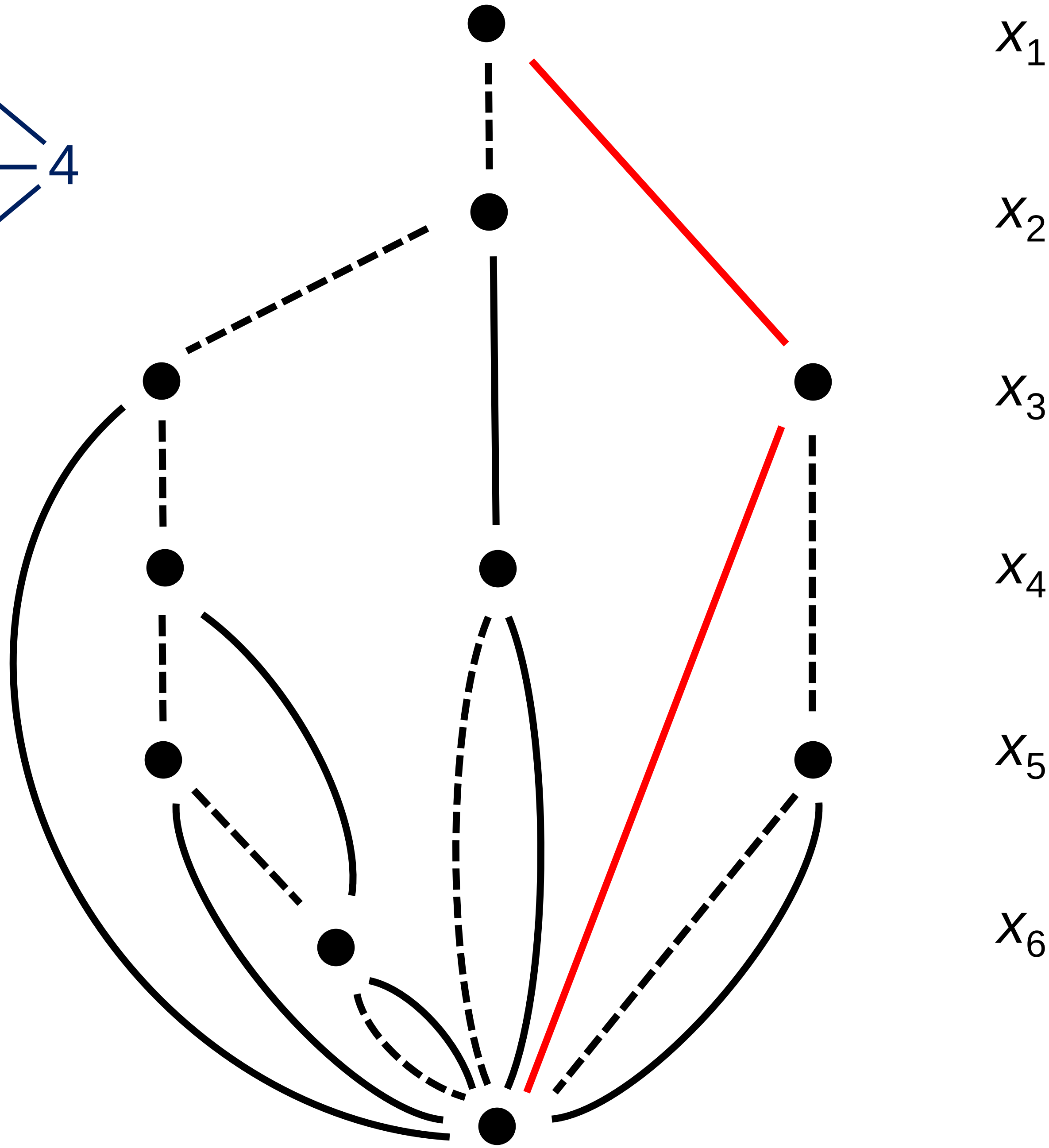
Exact BDD for independent set problem

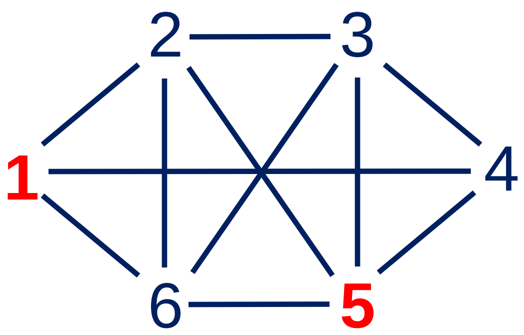
"zero-suppressed" BDD



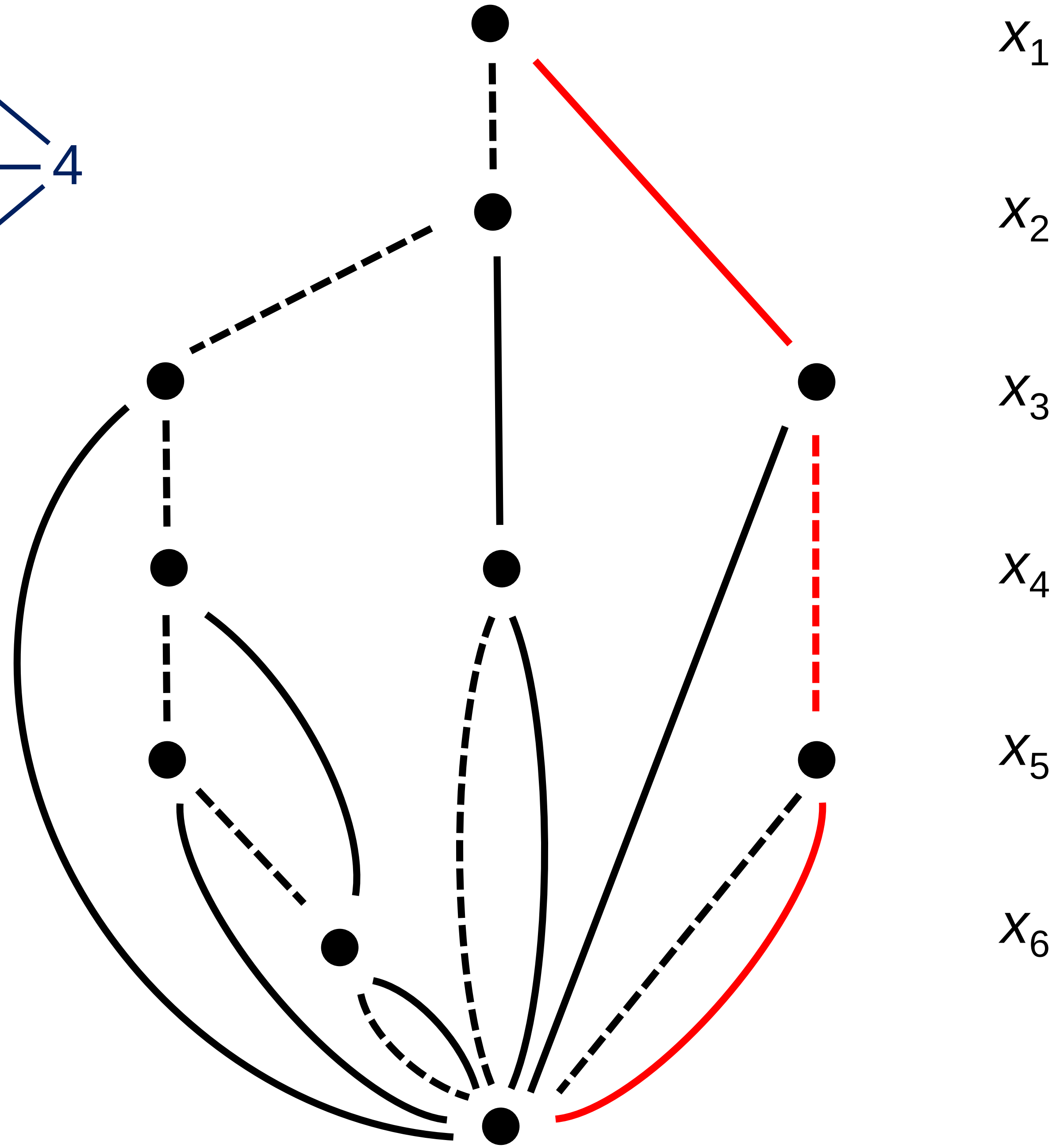


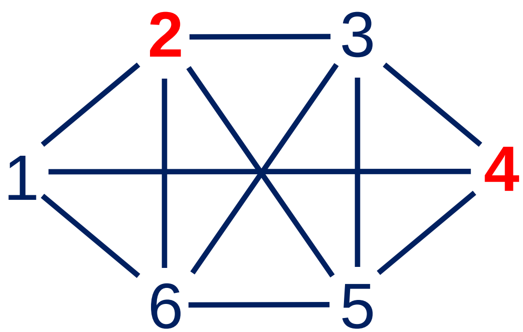
Paths from top to bottom correspond to the 11 feasible solutions



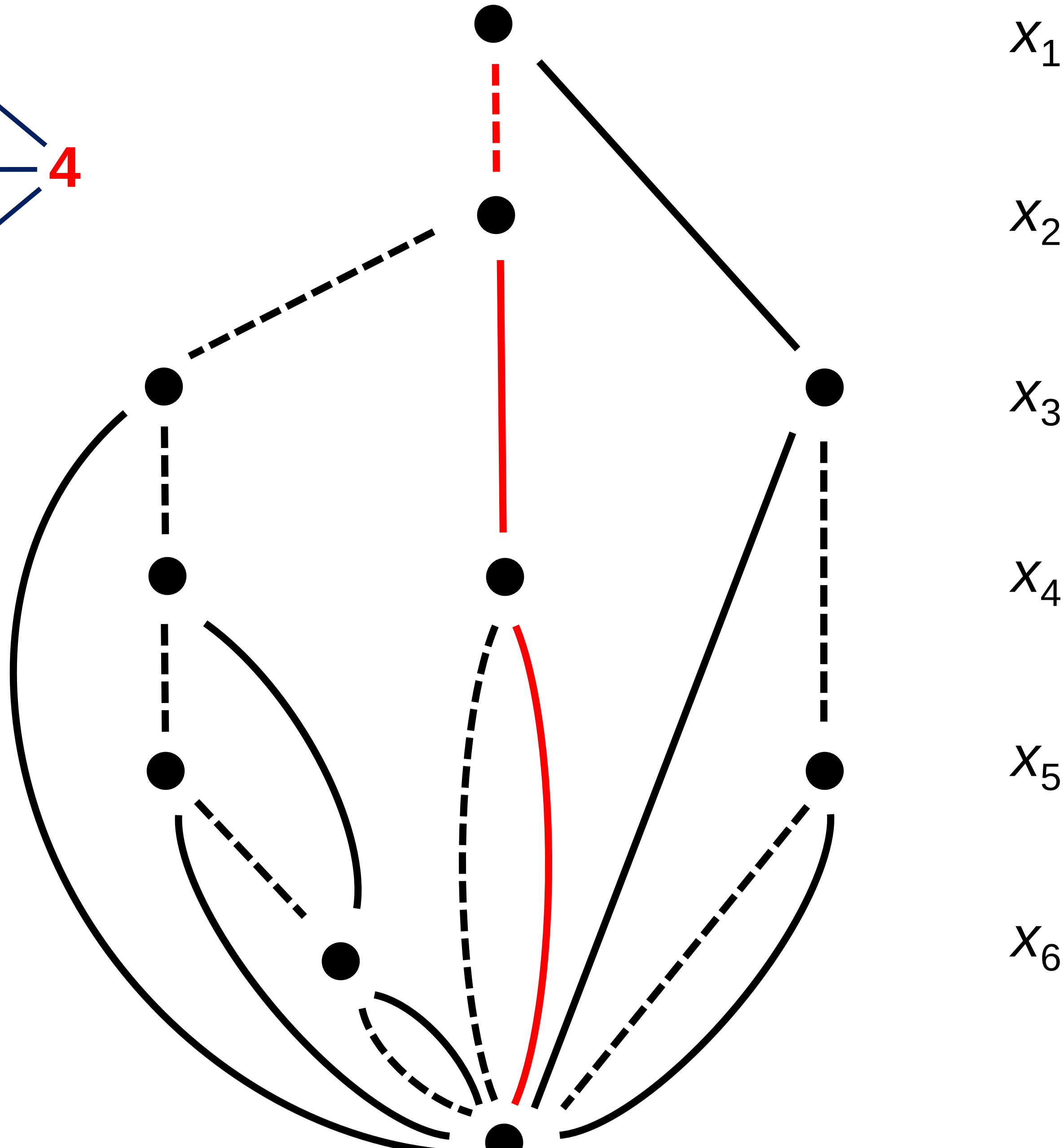


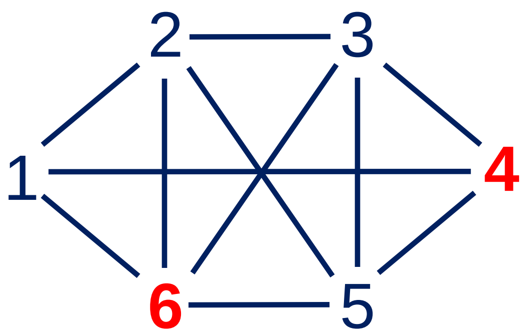
Paths from top to bottom correspond to the 11 feasible solutions



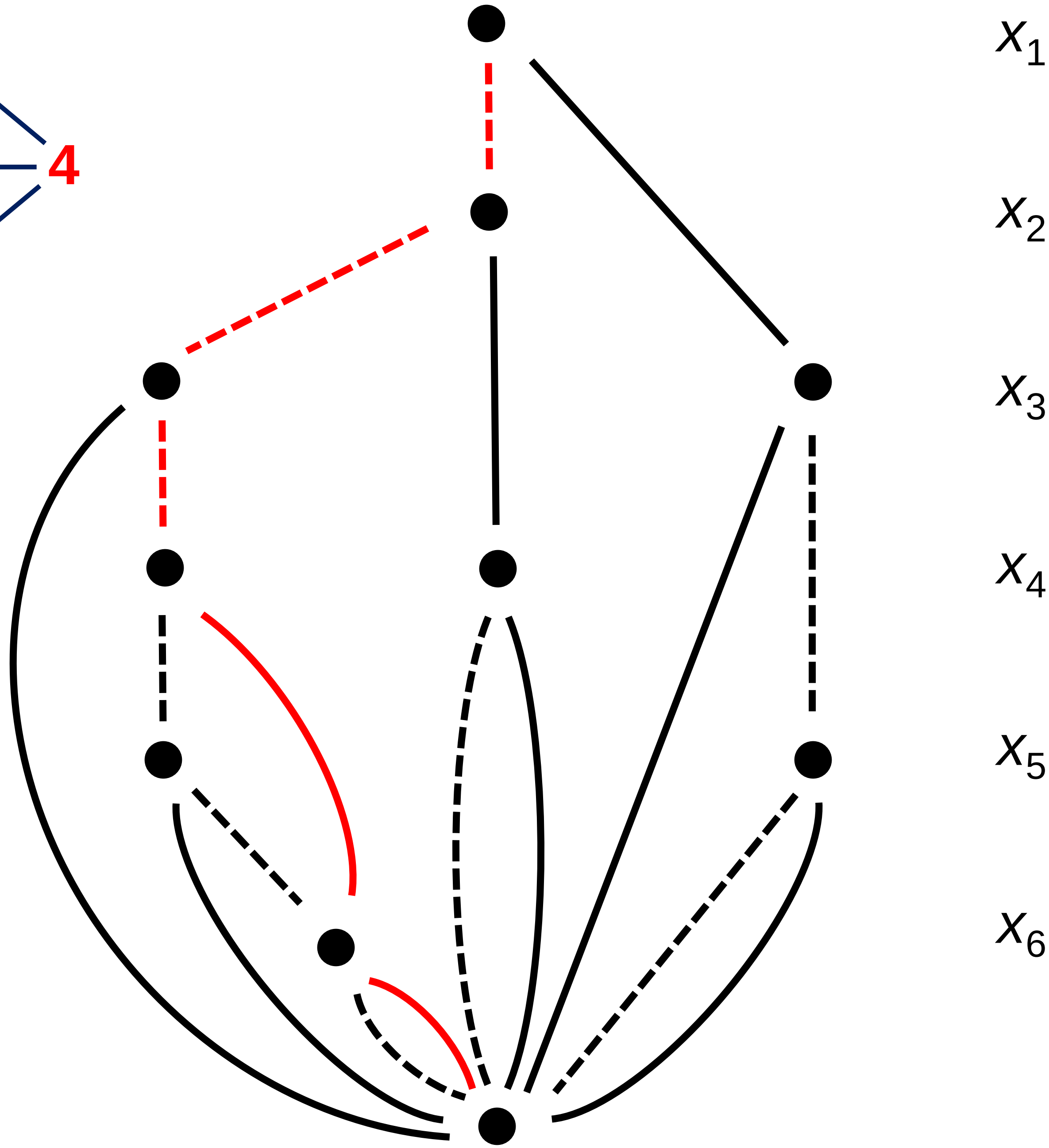


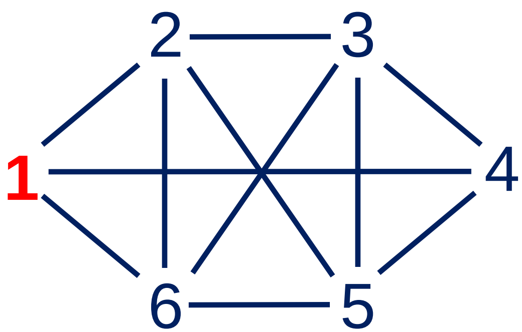
Paths from top to bottom correspond to the 11 feasible solutions

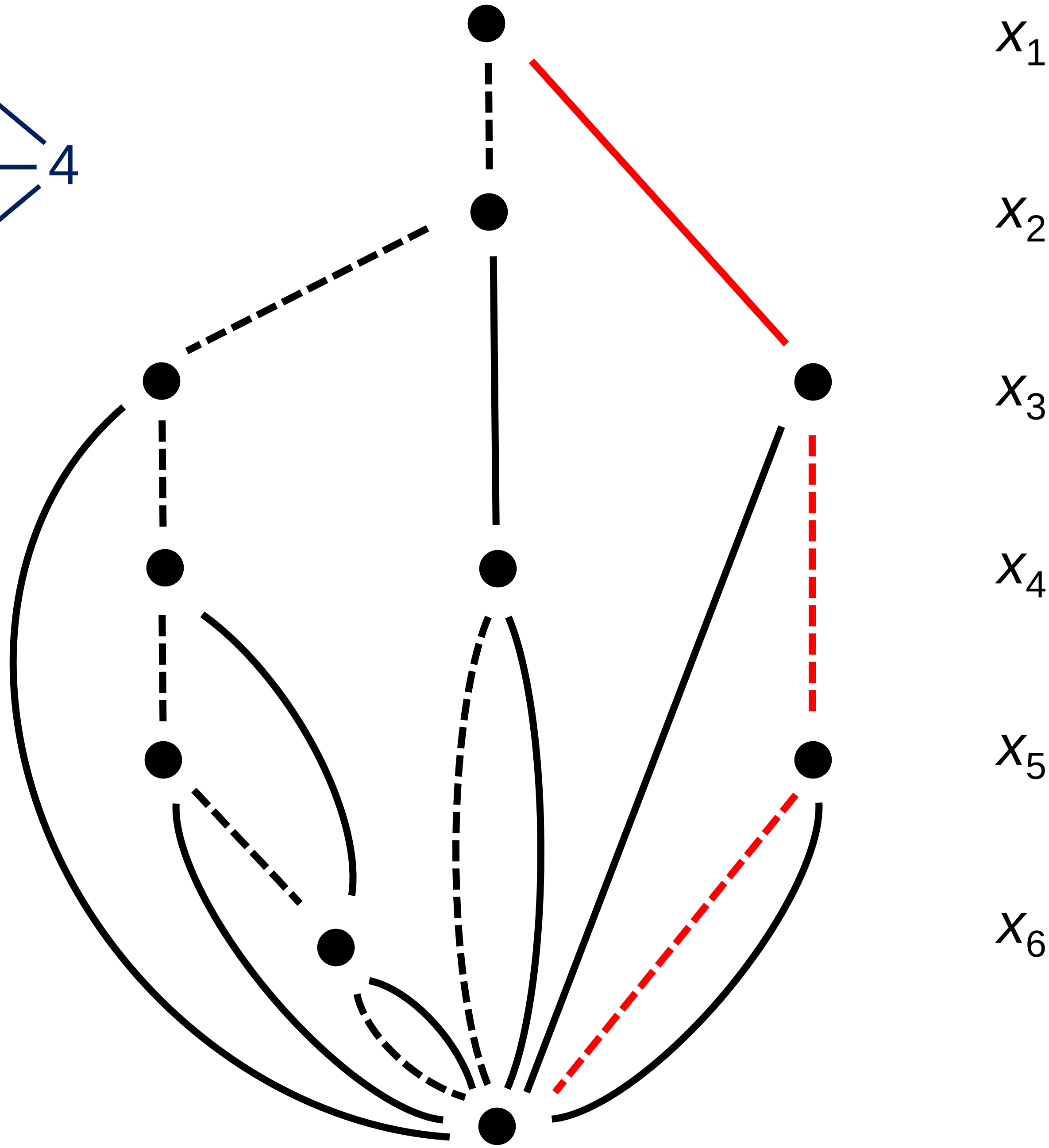




Paths from top to bottom correspond to the 11 feasible solutions

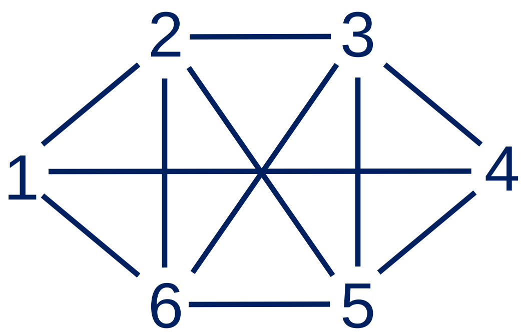




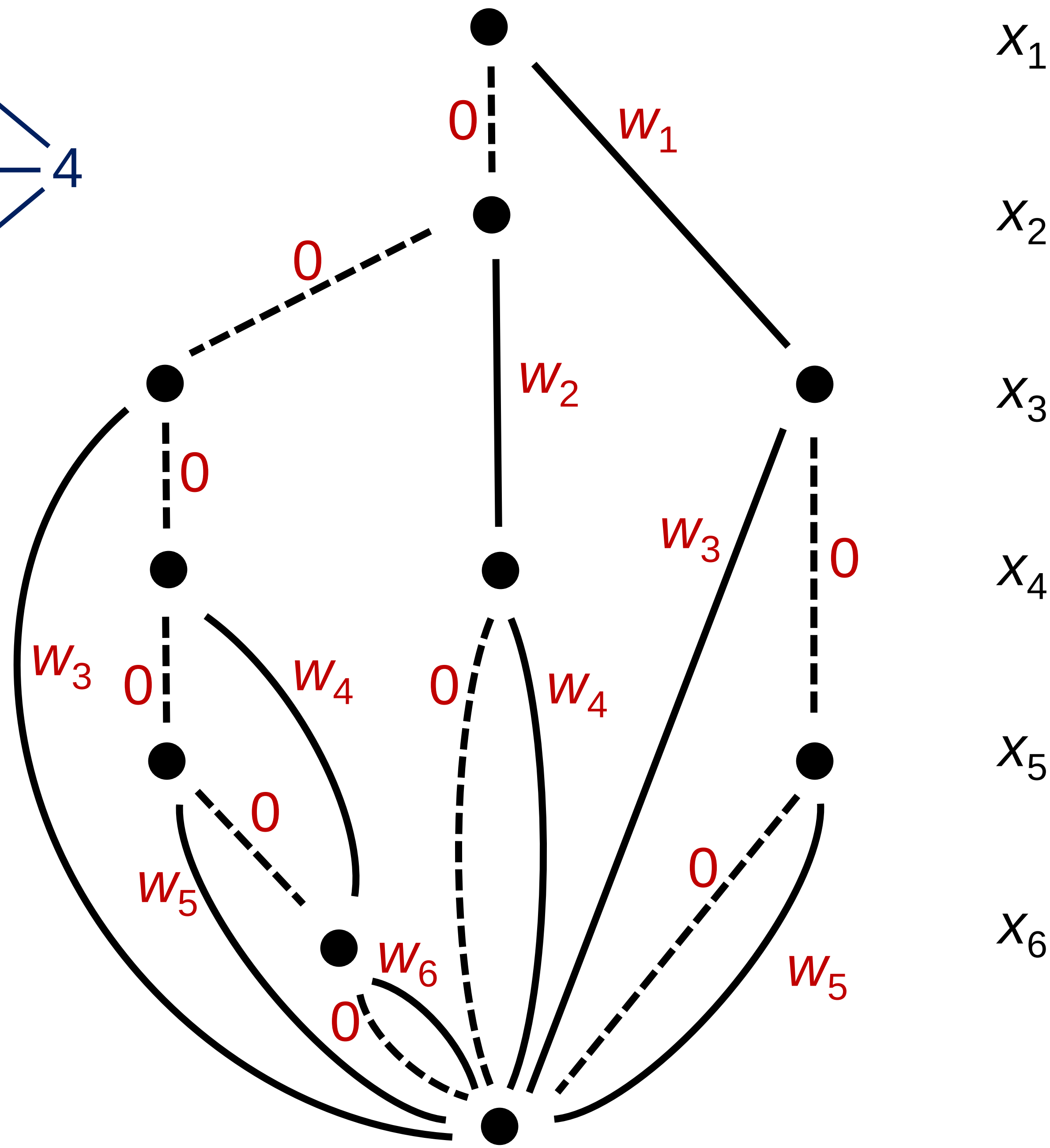


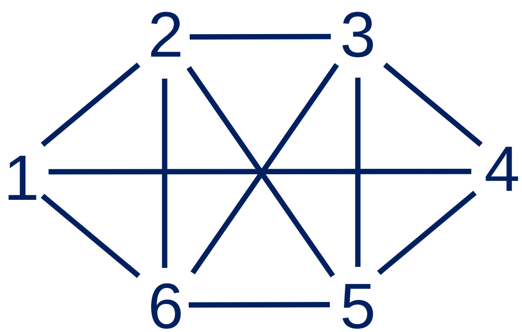
Paths from top to bottom correspond to the 11 feasible solutions

…and so forth



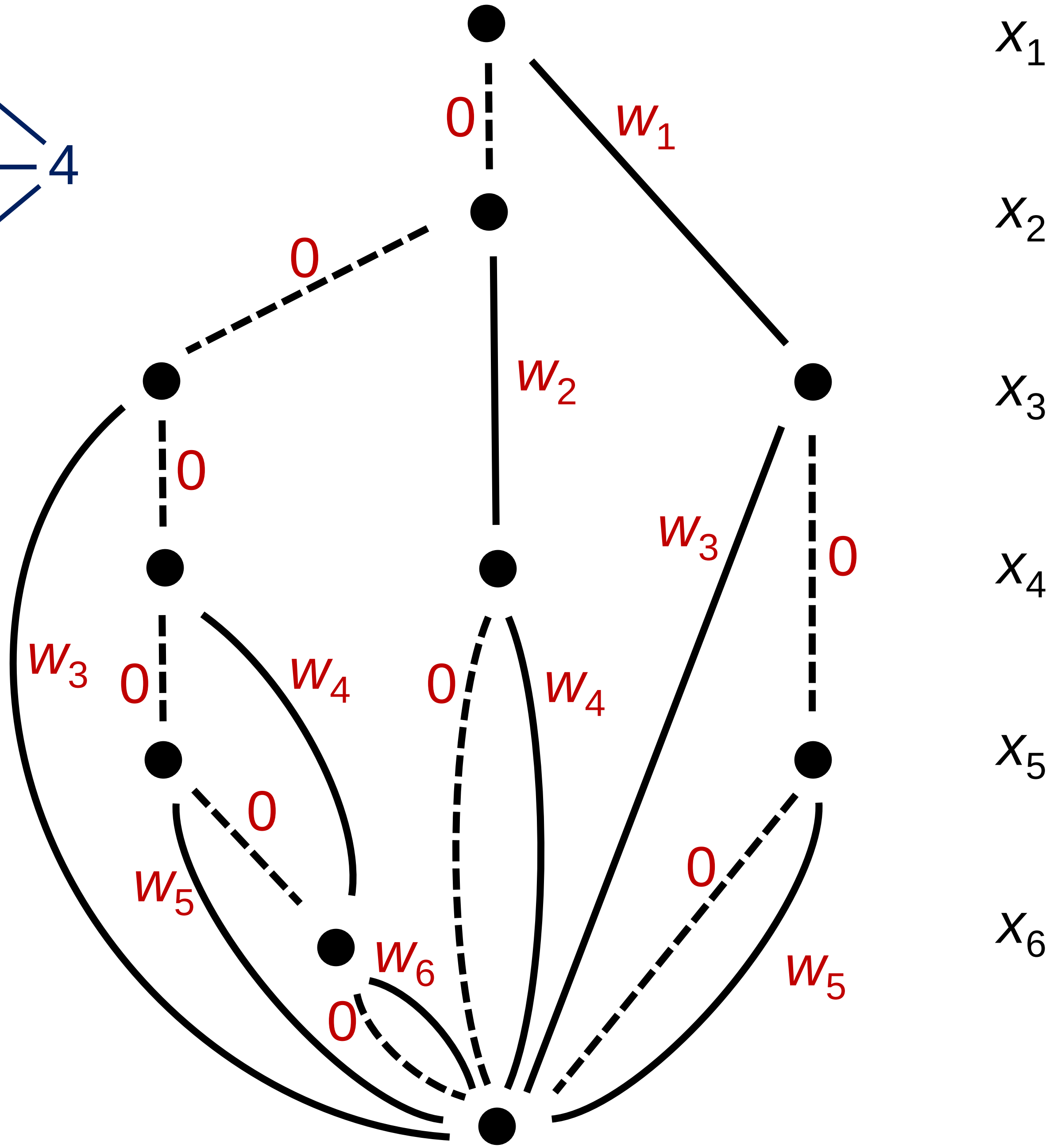
For objective function, associate weights with arcs





For objective function, associate weights with arcs

Optimal solution is **longest path**

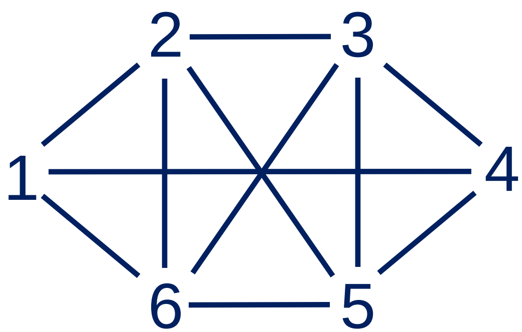


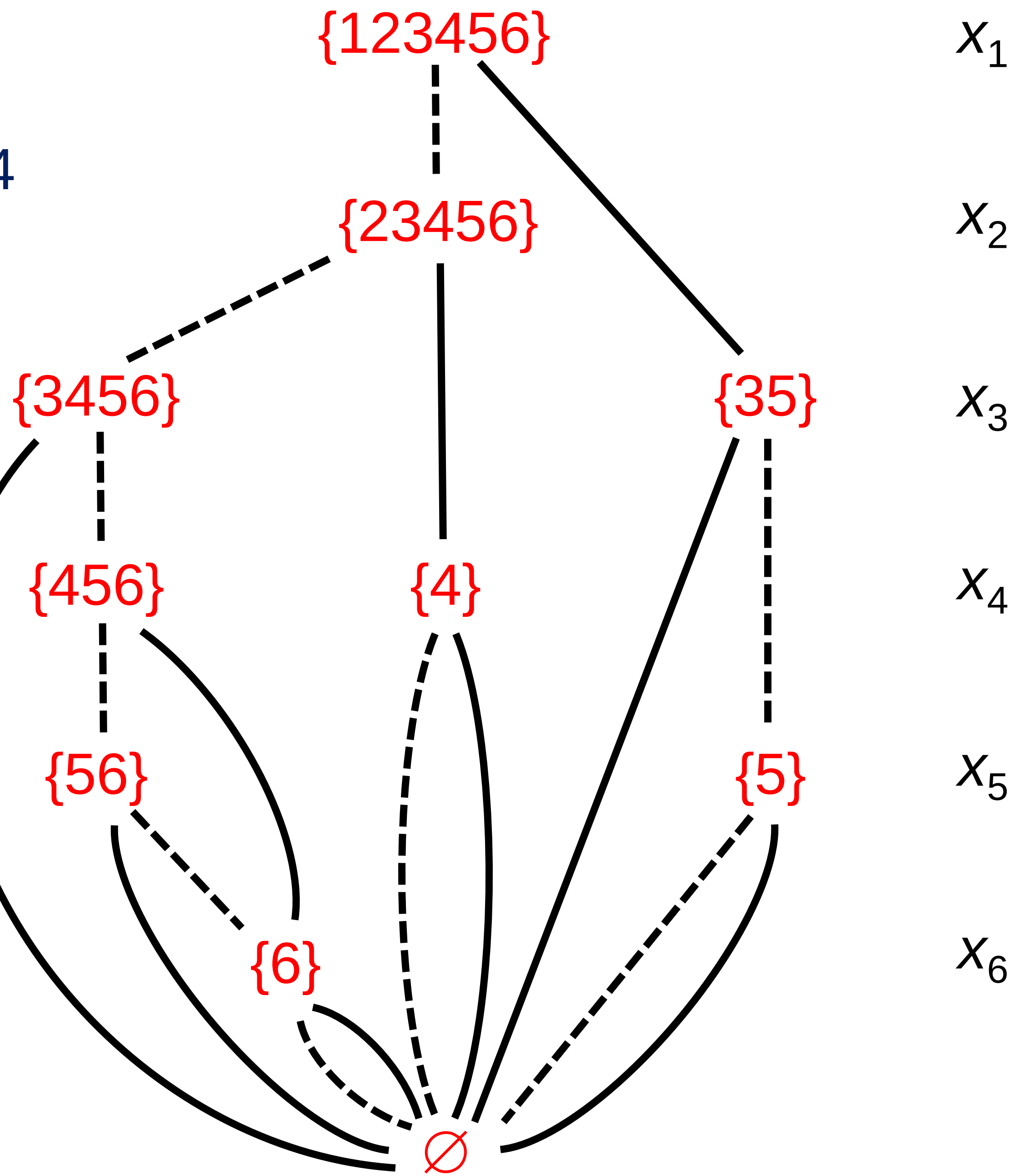
Objective Function

- In general, objective function can be any **separable function.**
  - Linear or nonlinear, convex or nonconvex
- BDDs can be generalized to **nonseparable** objective functions**.**
  - There is a unique reduced BDD with **canonical** edge costs.

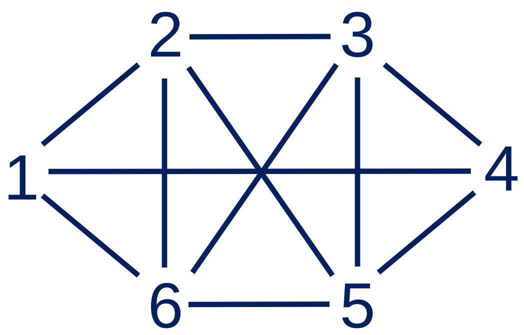
DP-Style Modeling

- To build a decision diagram, we need information about the problem**.**
  - We model the problem with **state variables**.
  - Rather than inequalities as in MIP.
  - Or global constraints as in CP.





To build BDD, associate **state** with each node



{123456}

x_1

x_2

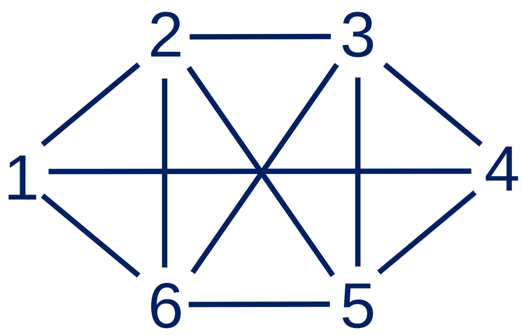
x_3

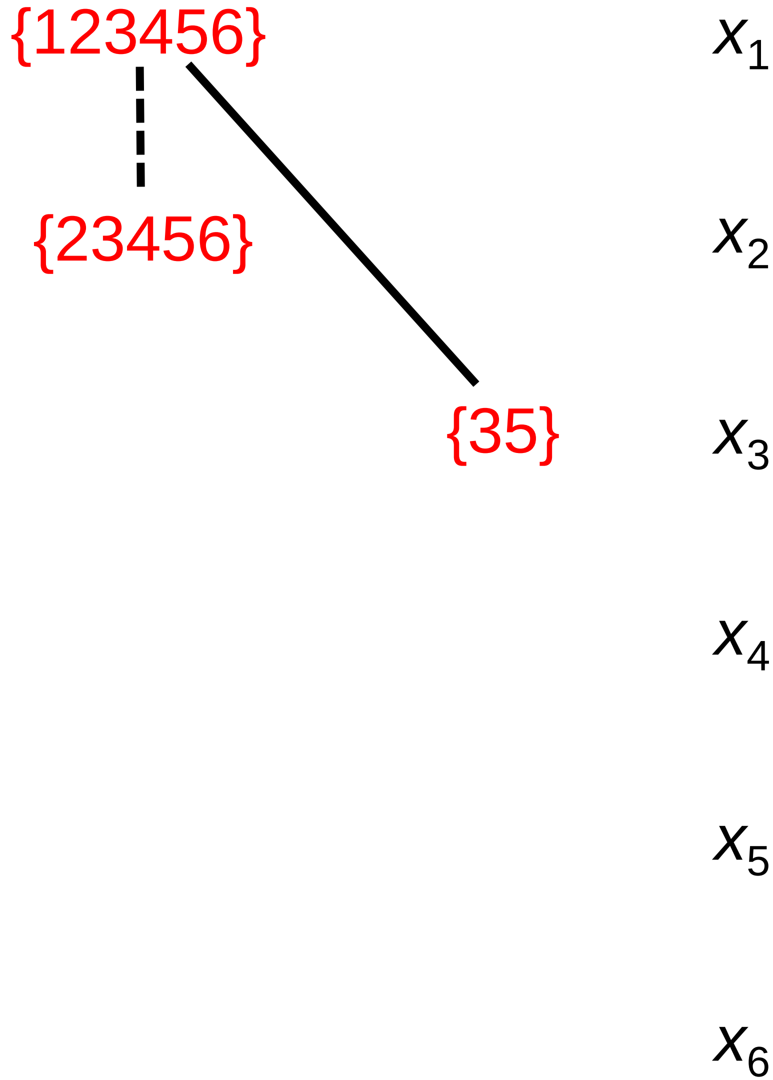
x_4

x_5

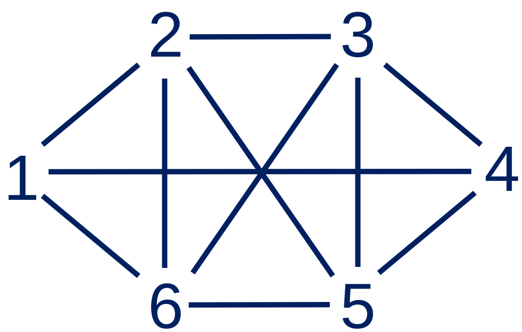
To build BDD,
associate **state**
with each node

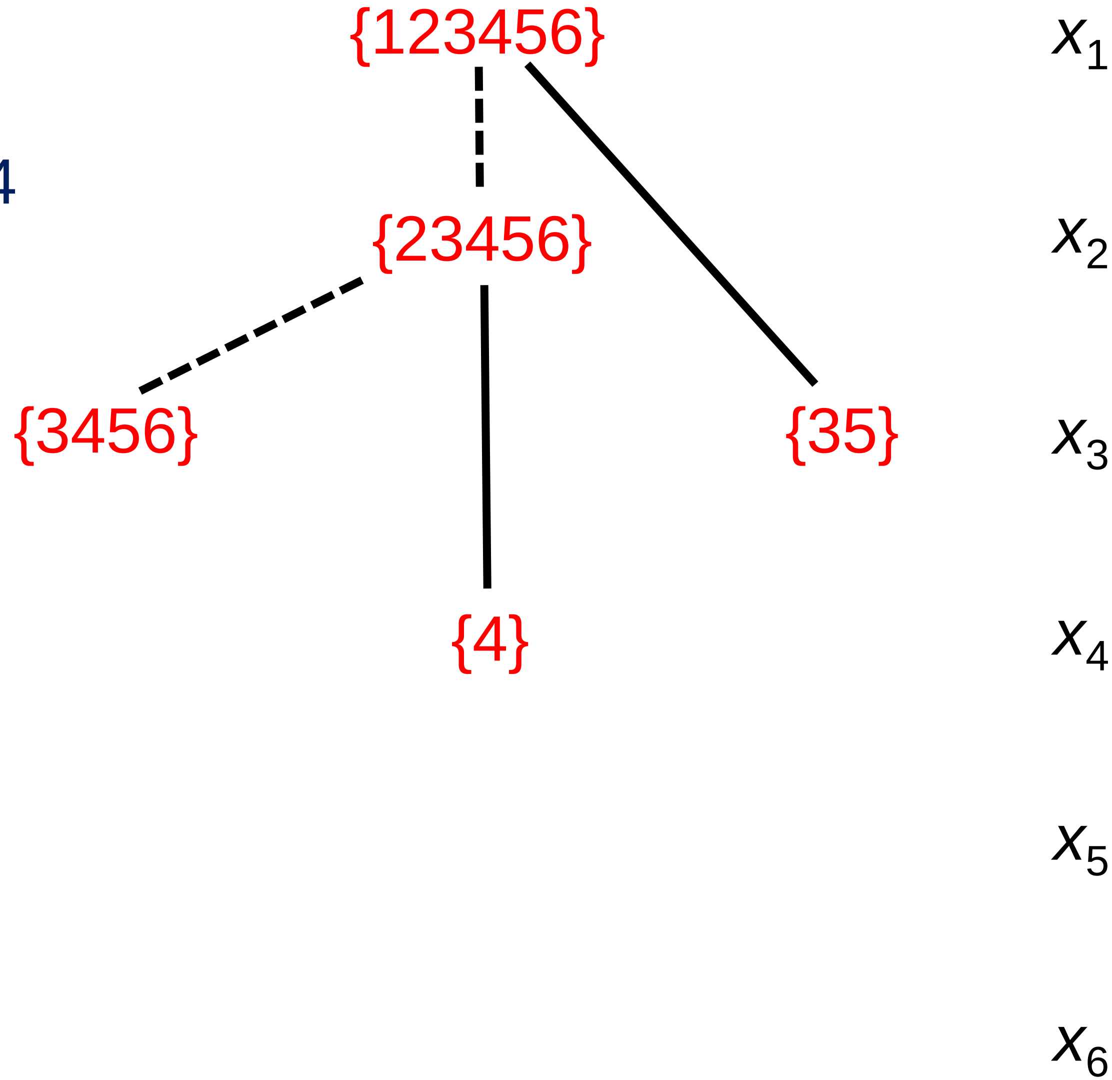
x_6



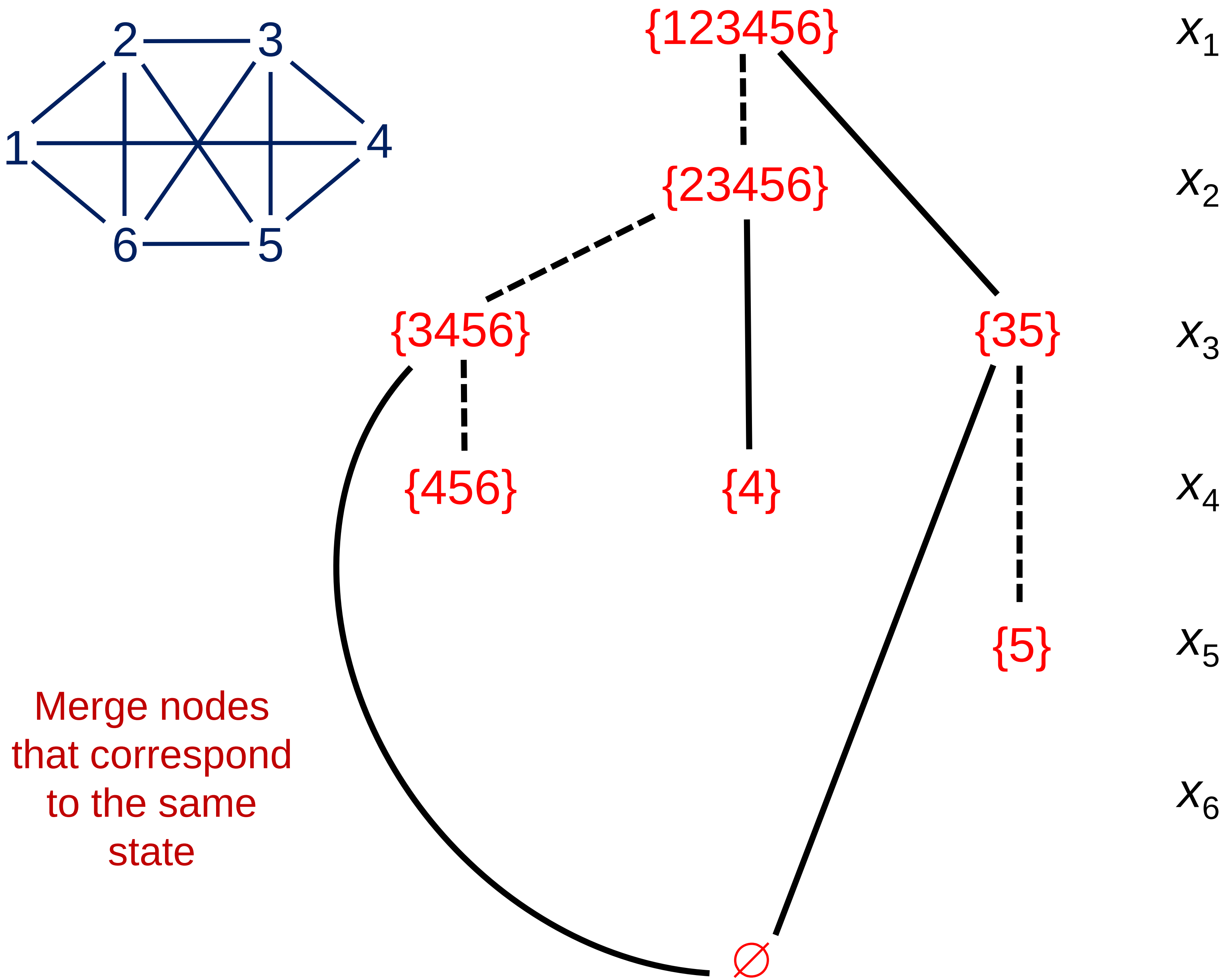


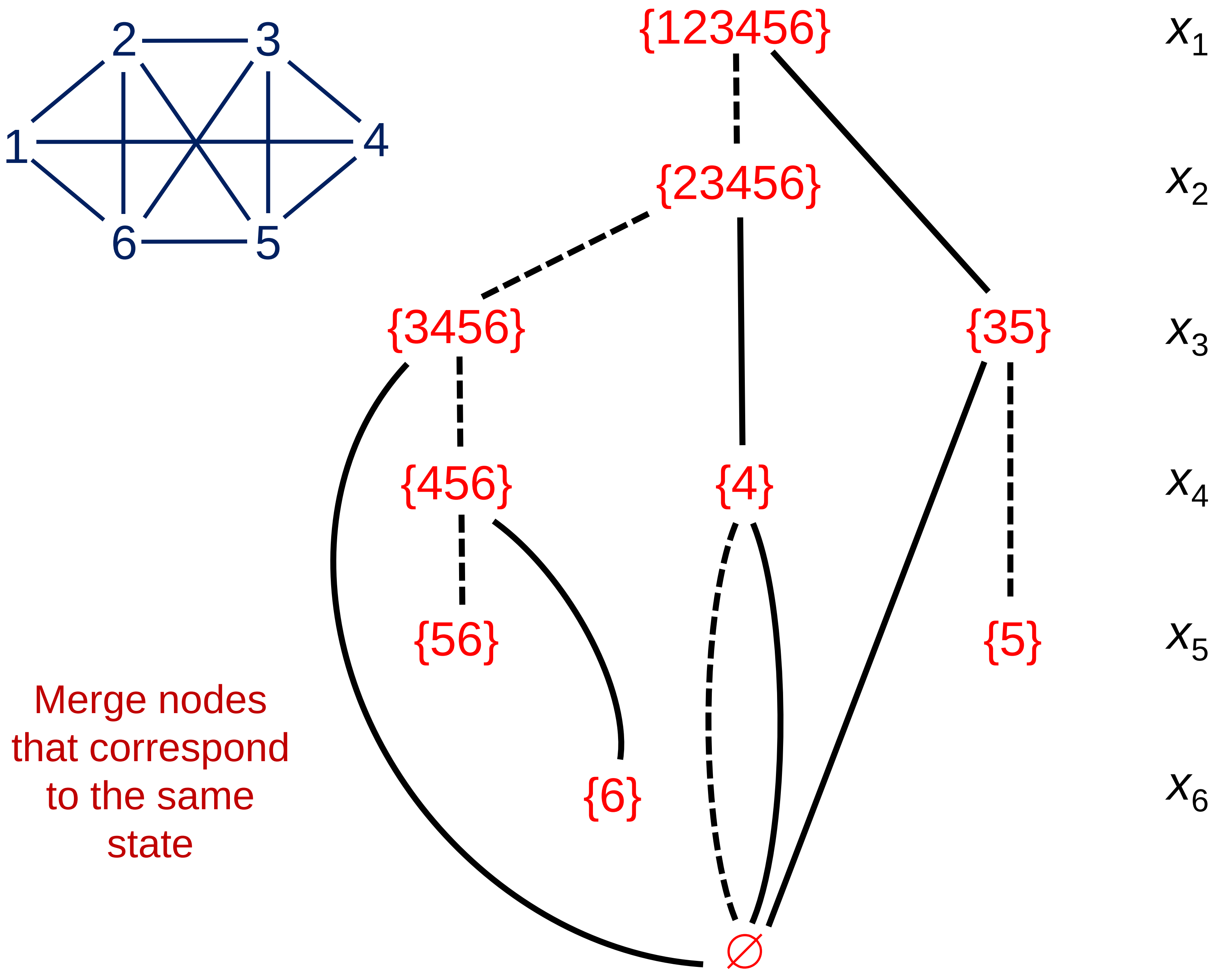
To build BDD,
associate **state**
with each node



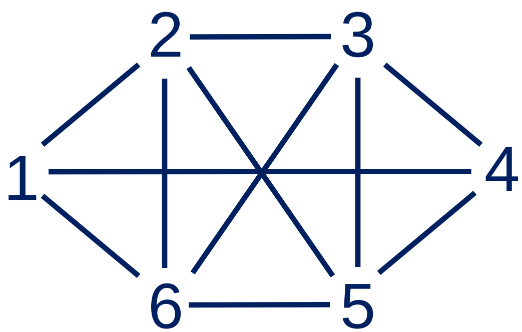


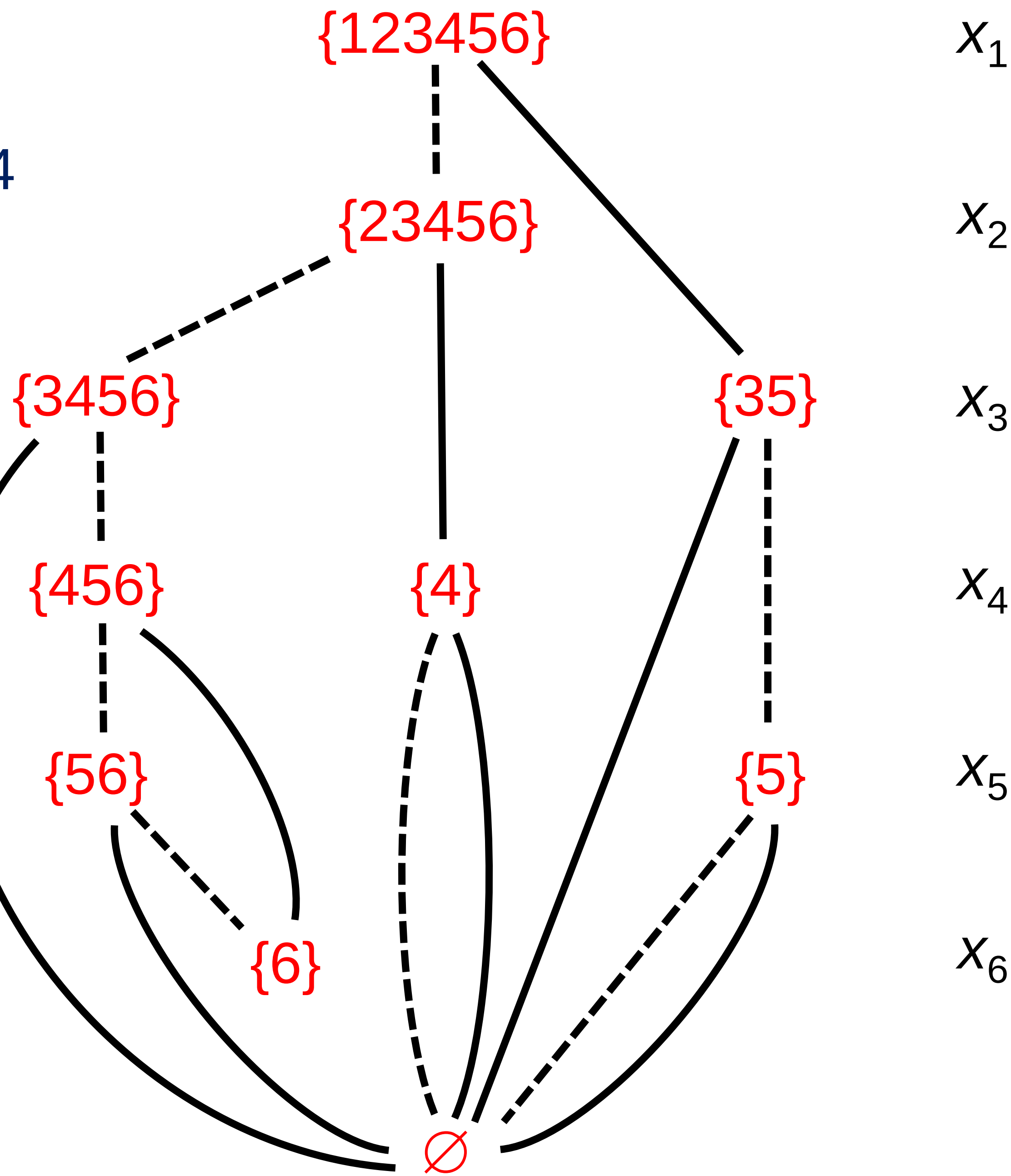
To build BDD, associate **state** with each node


1
2
3
4
5
6
{123456}
{23456}
{3456}
{35}
{456}
{4}
{5}
∅
x_1
x_2
x_3
x_4
x_5
x_6
Merge nodes that correspond to the same state

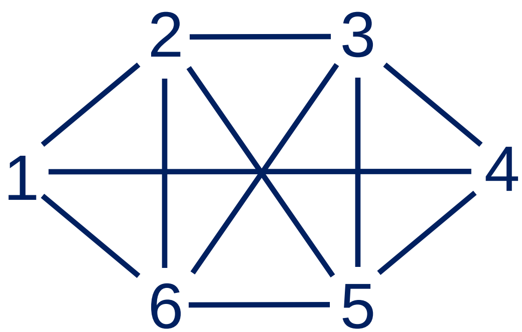


1
2
3
4
5
6
{123456}
{23456}
{3456}
{35}
{456}
{4}
{56}
{5}
{6}
∅
x_1
x_2
x_3
x_4
x_5
x_6
Merge nodes that correspond to the same state

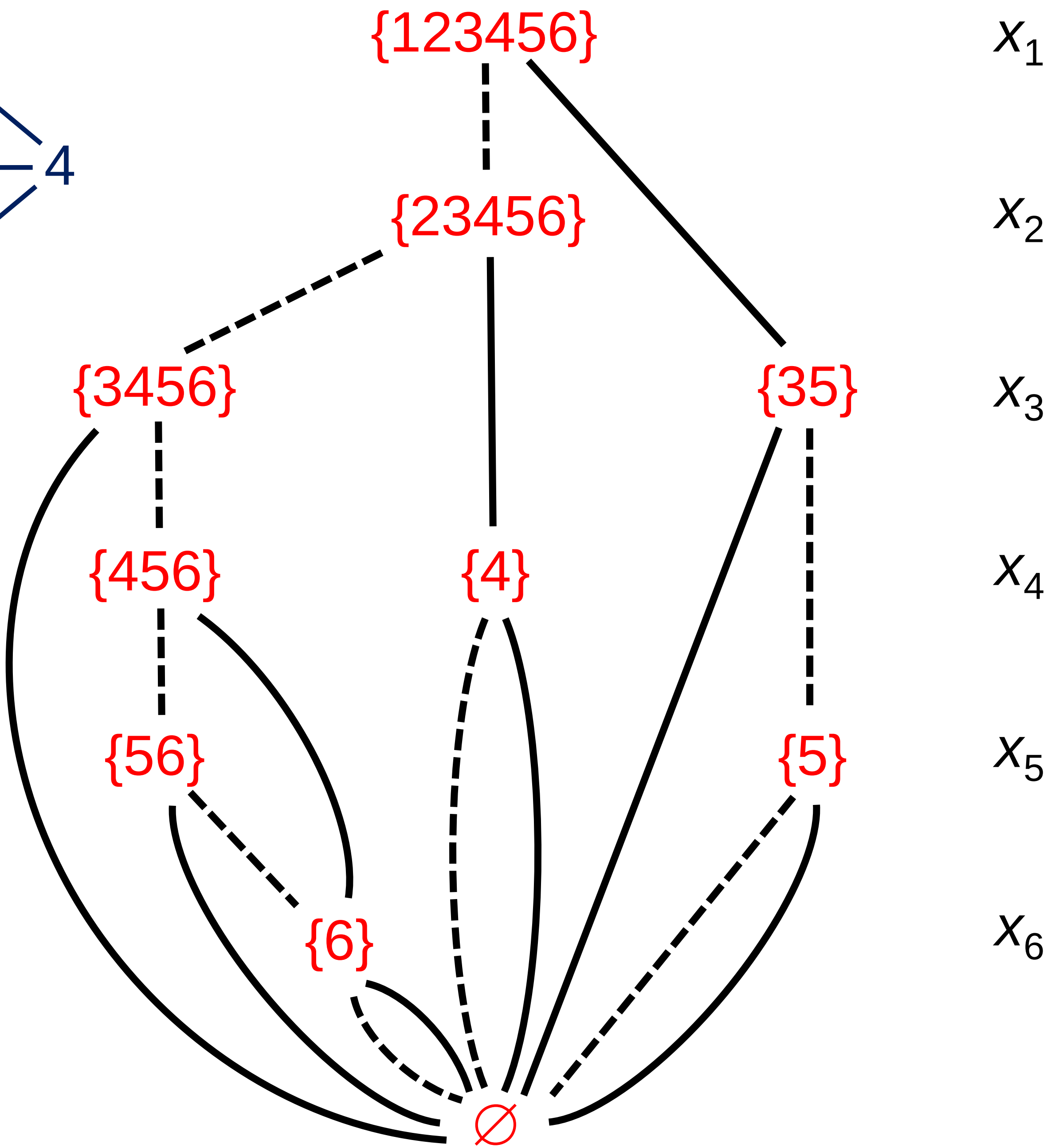




Merge nodes that correspond to the same state



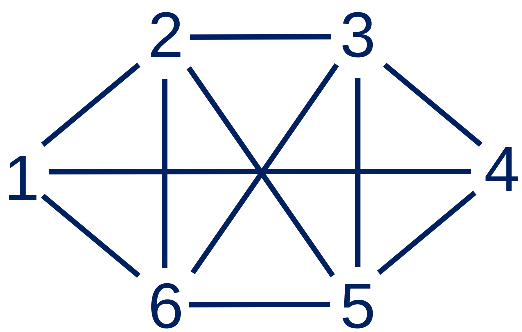
Width = 2



Merge nodes
that correspond
to the same
state

Relaxation Bounding

- To obtain a bound on the objective function:
  - Use a **relaxed** decision diagram
  - Analogous to linear programming relaxation in MIP
  - This relaxation is **discrete**.
  - Doesn't require the linear inequality formulation of MIP.



{123456}

x_1

x_2

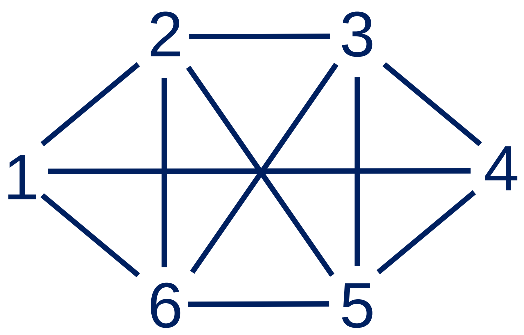
x_3

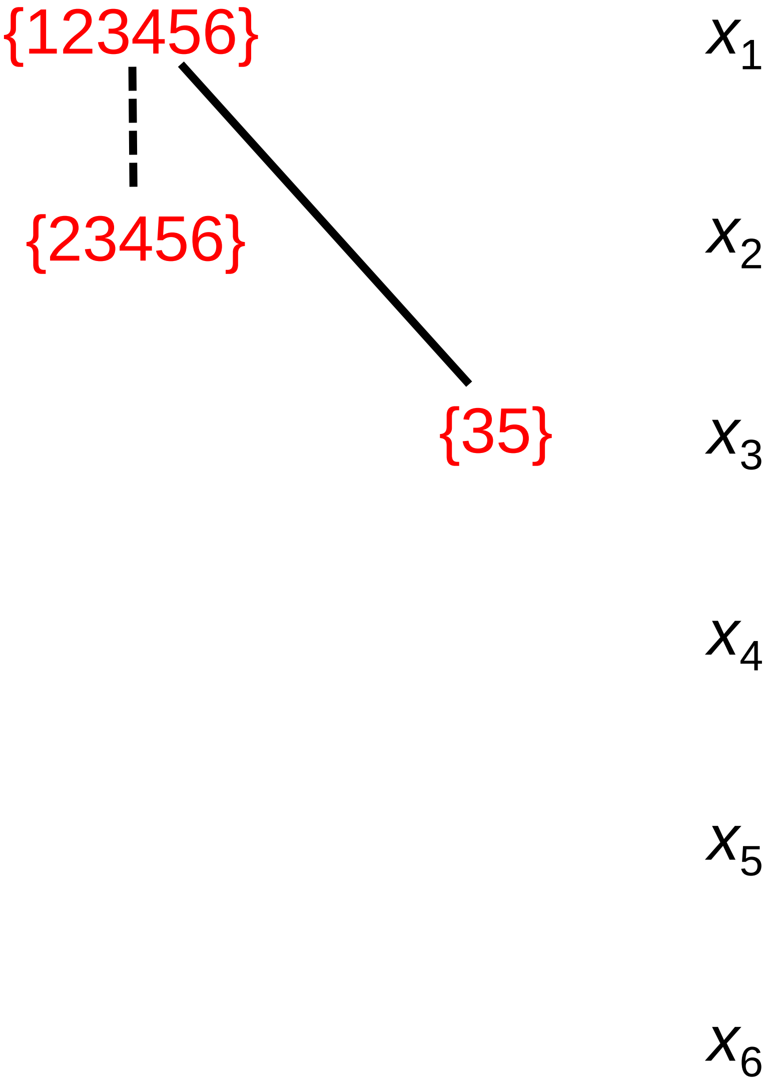
x_4

x_5

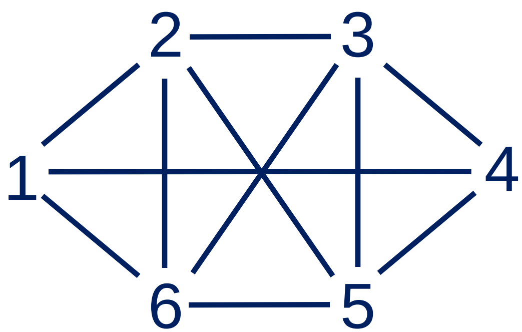
x_6

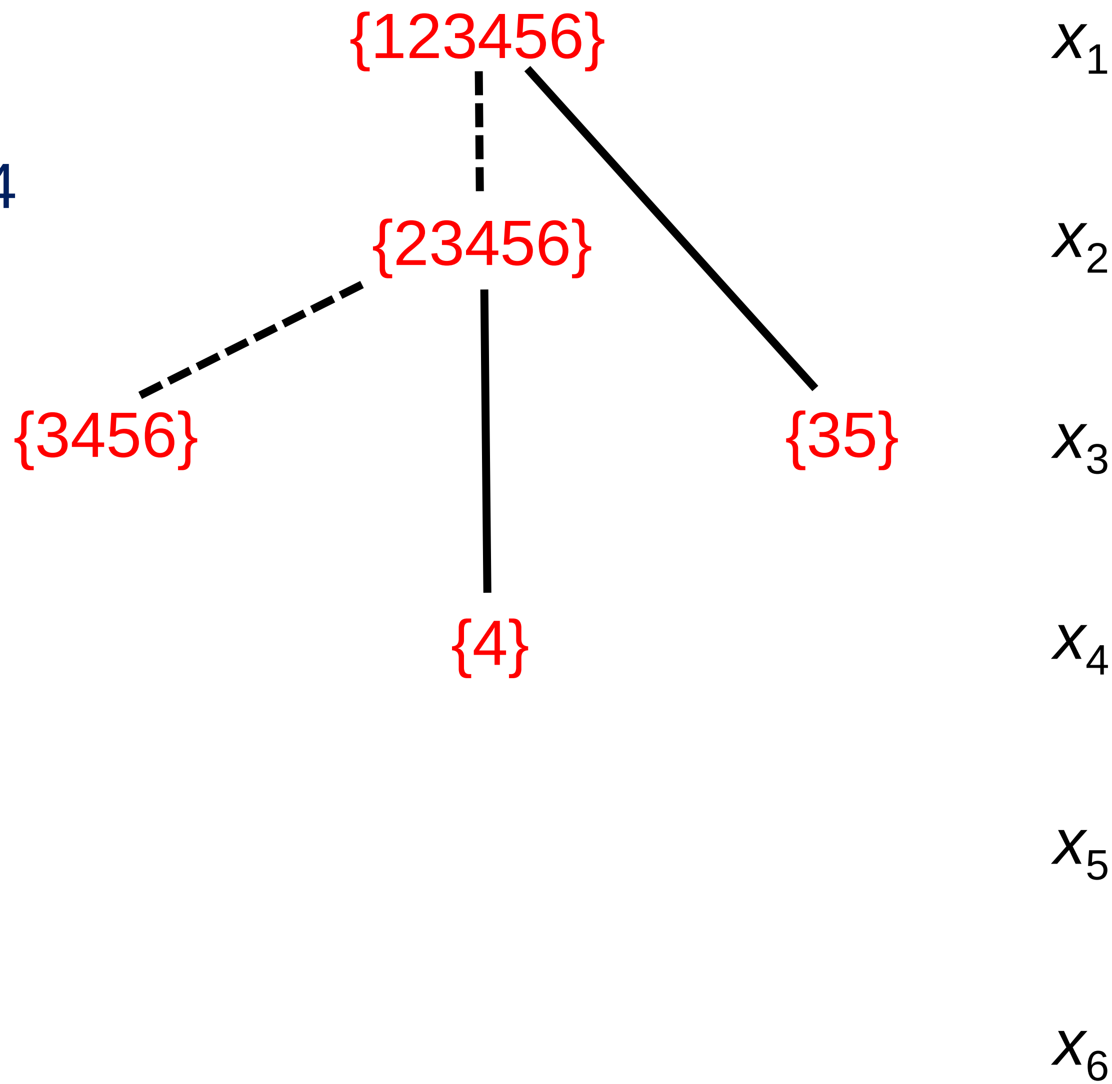
To build **relaxed** BDD, merge some additional nodes as we go along



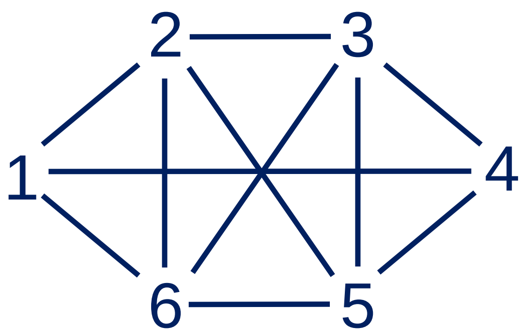


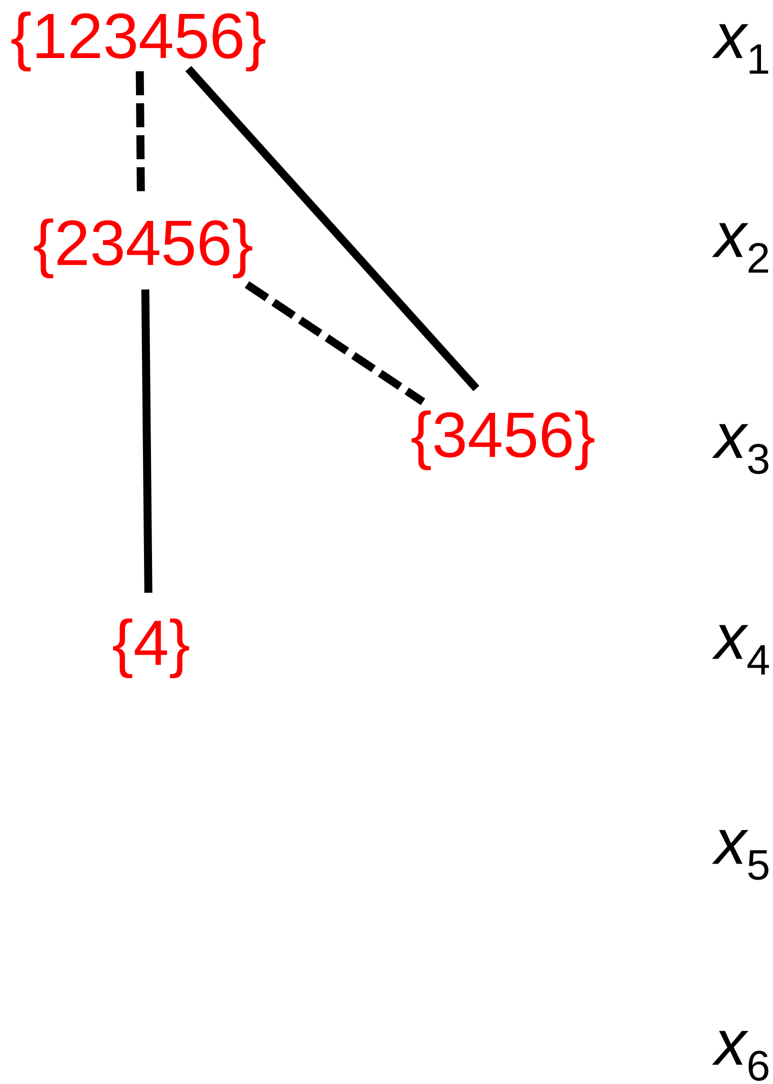
To build **relaxed** BDD, merge some additional nodes as we go along



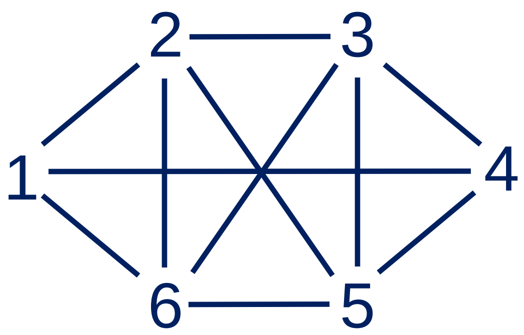


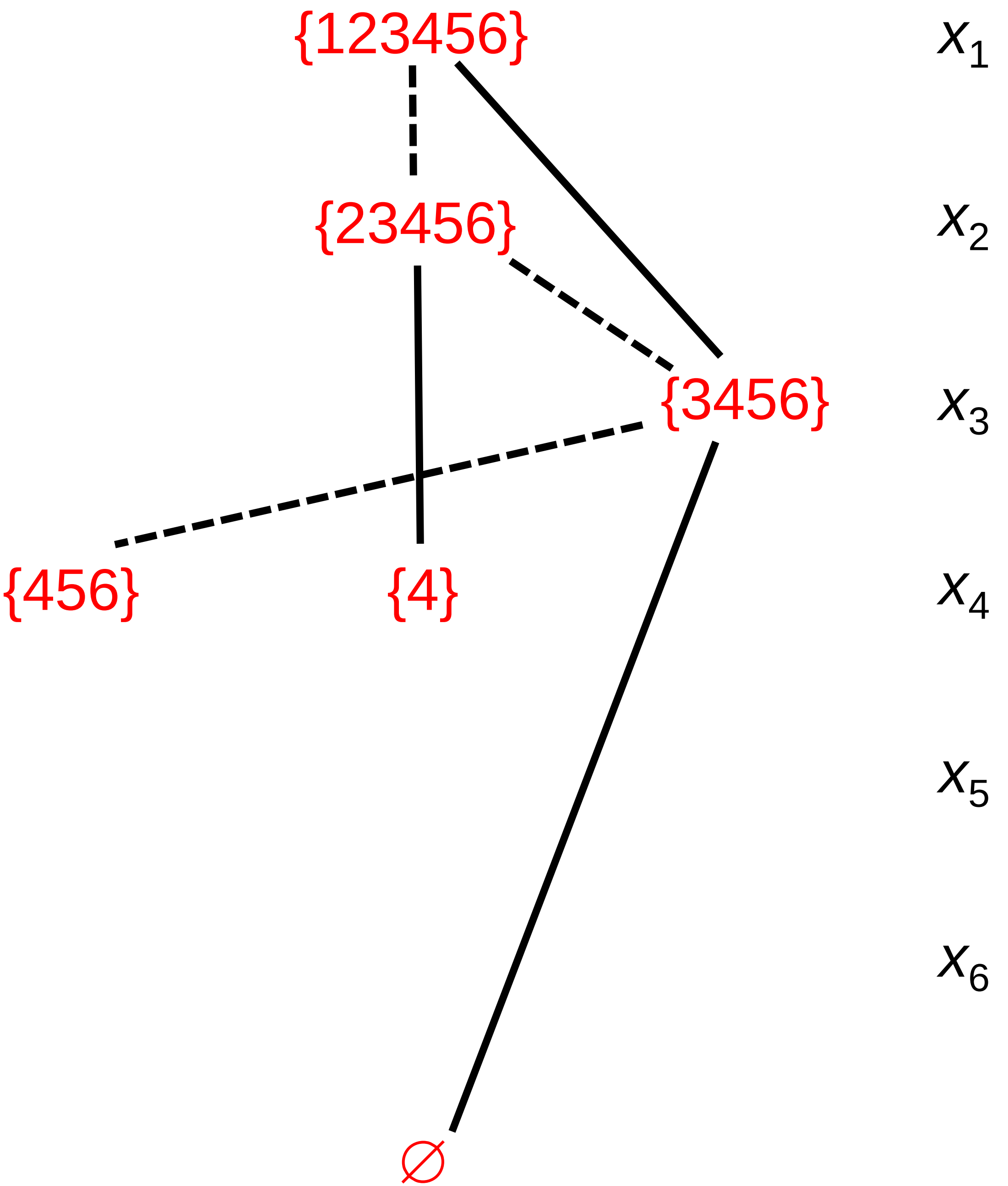
To build **relaxed** BDD, merge some additional nodes as we go along



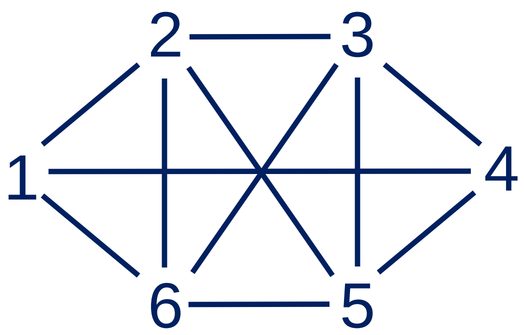


To build **relaxed** BDD, merge some additional nodes as we go along

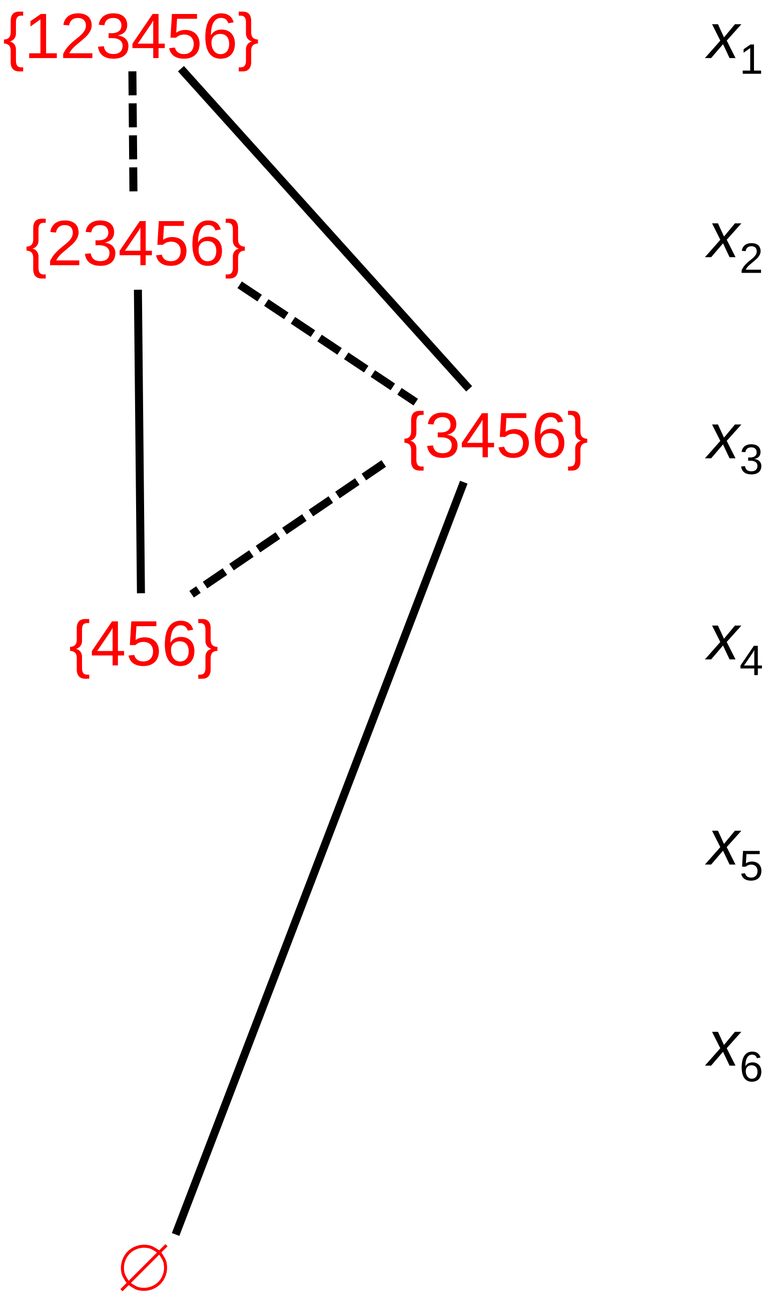


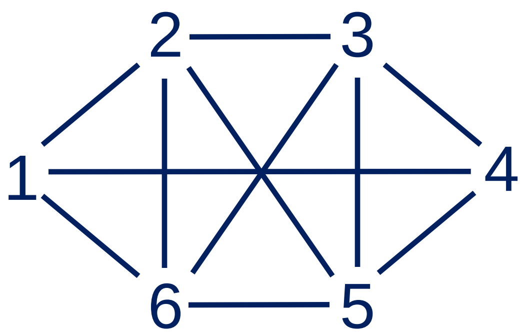


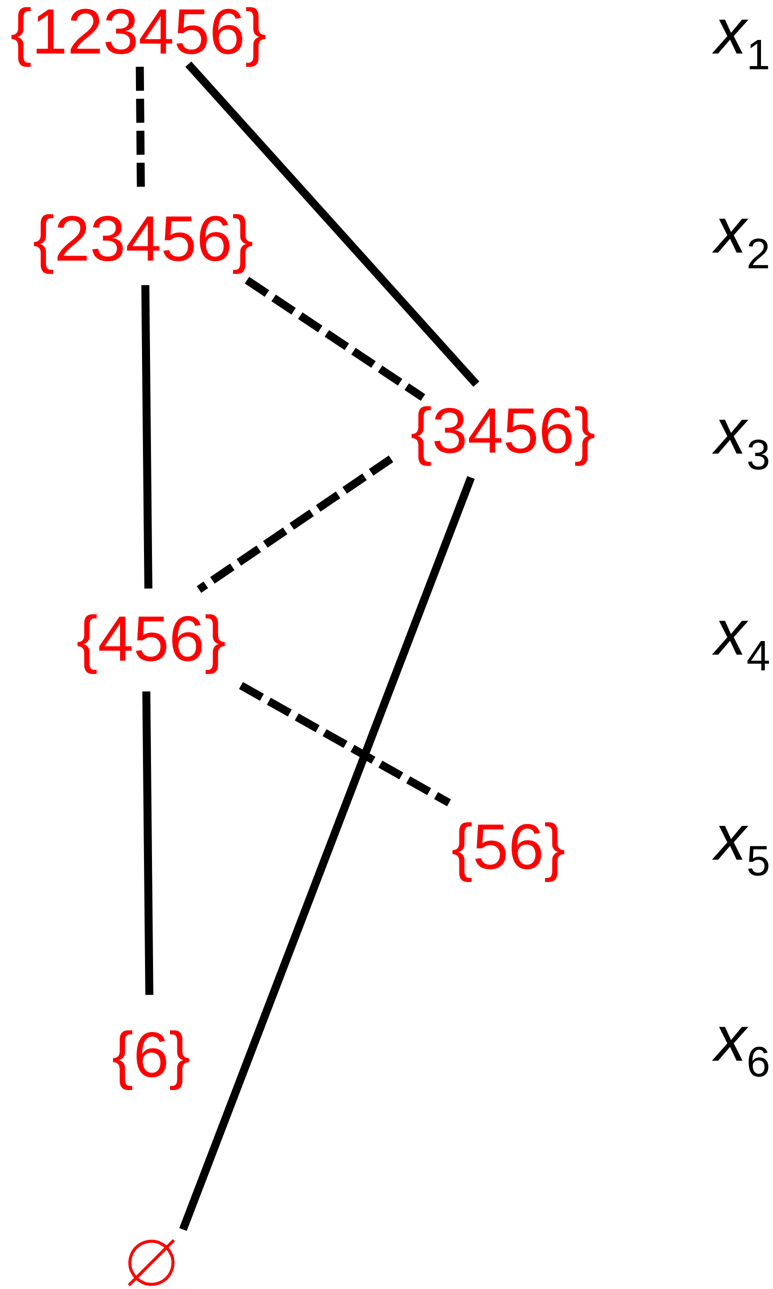
To build **relaxed** BDD, merge some additional nodes as we go along



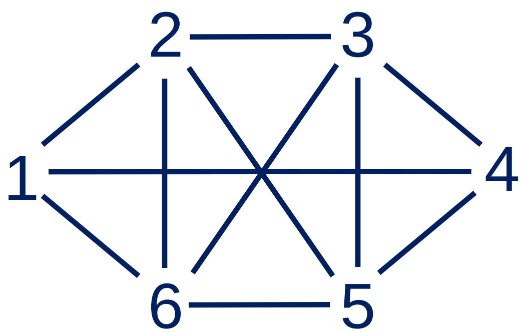
To build **relaxed** BDD, merge some additional nodes as we go along



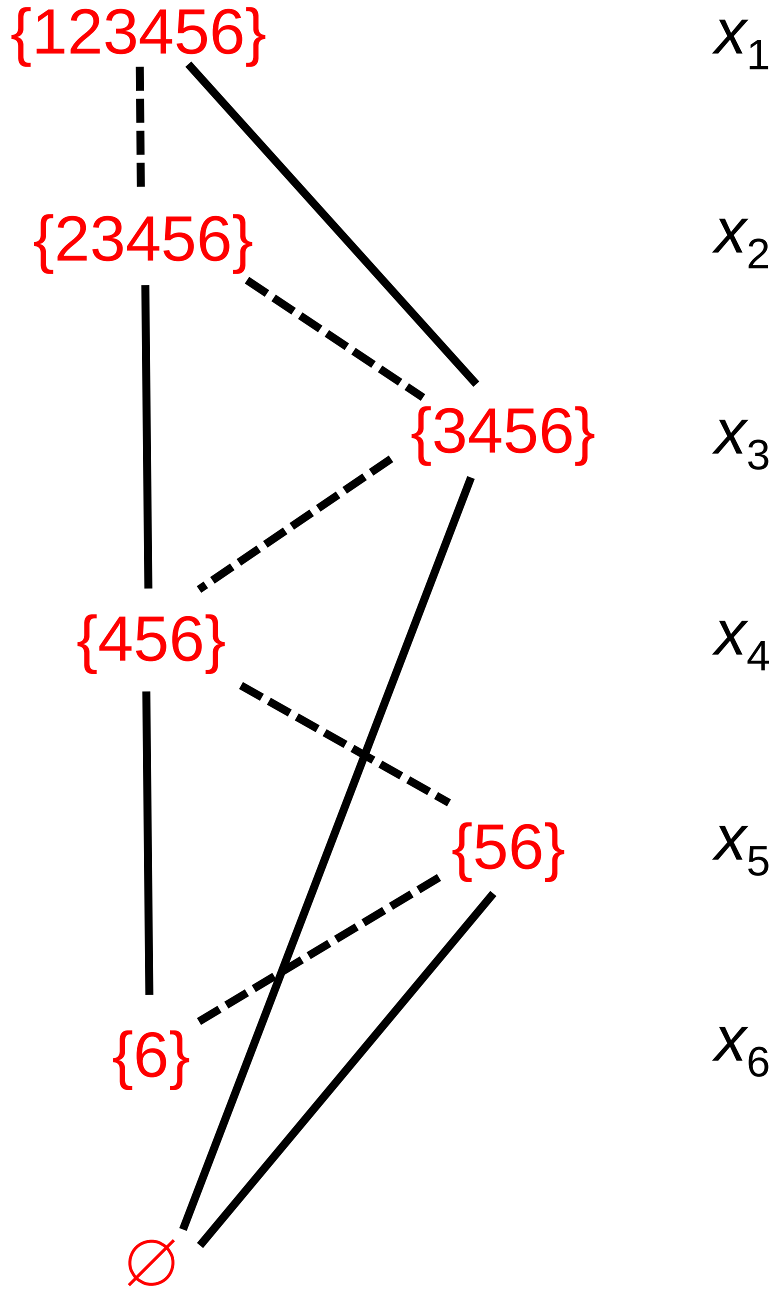


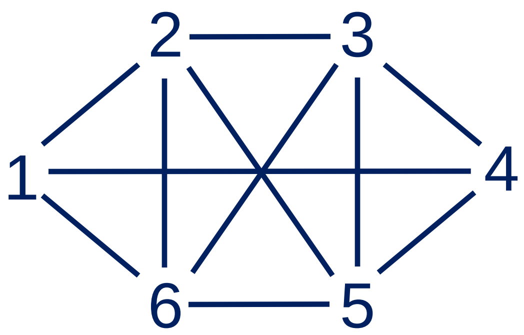


To build **relaxed** BDD, merge some additional nodes as we go along



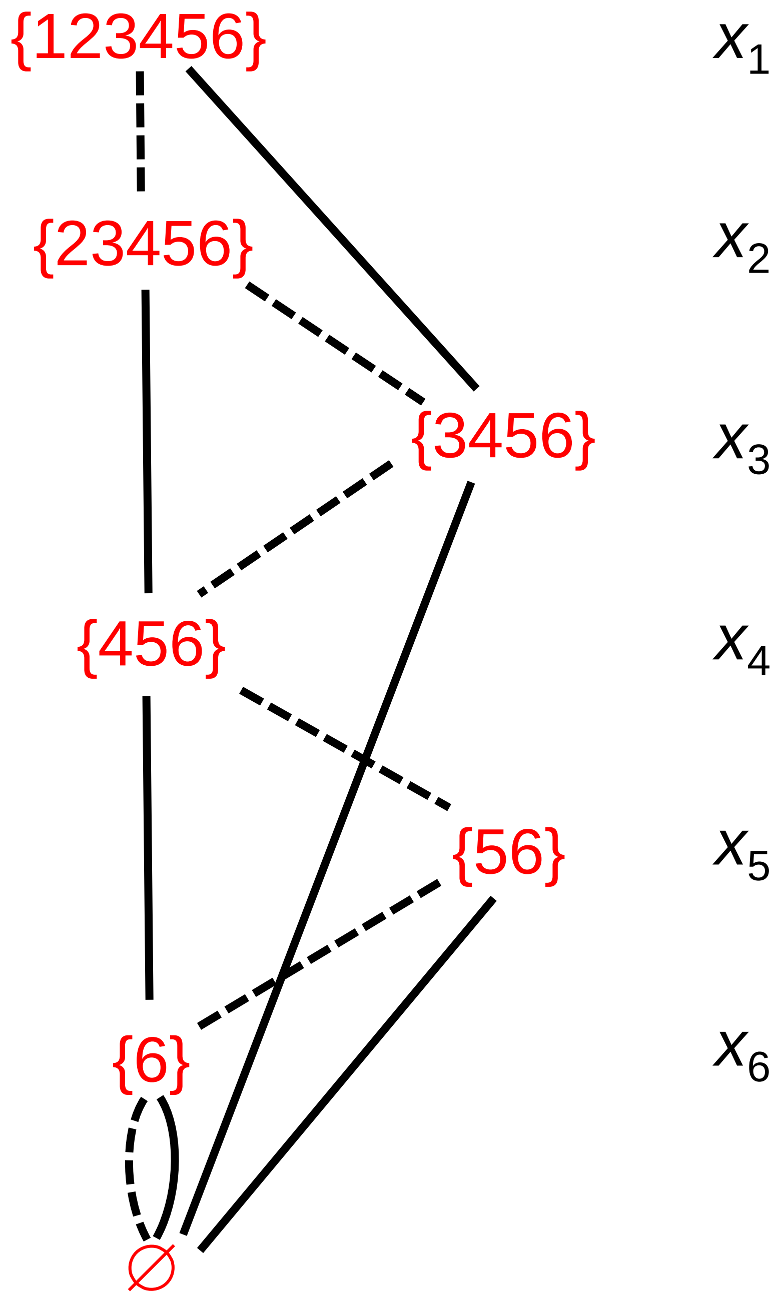
To build **relaxed** BDD, merge some additional nodes as we go along

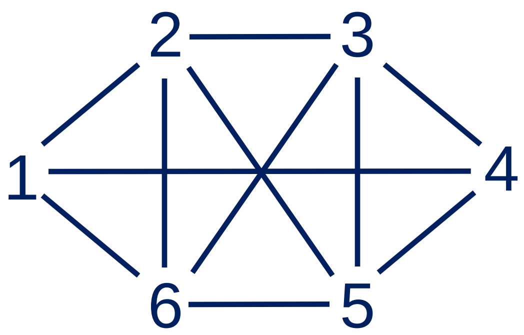




Width = 1

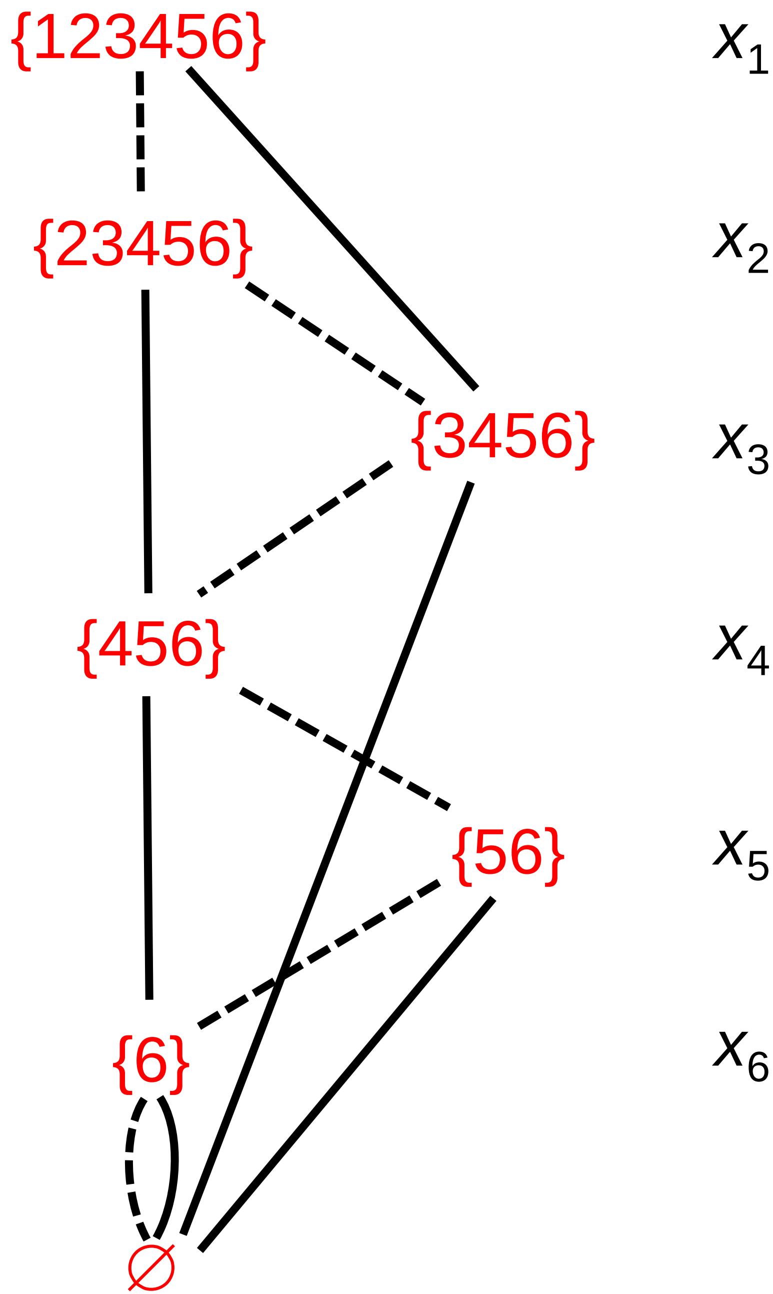
To build **relaxed** BDD, merge some additional nodes as we go along

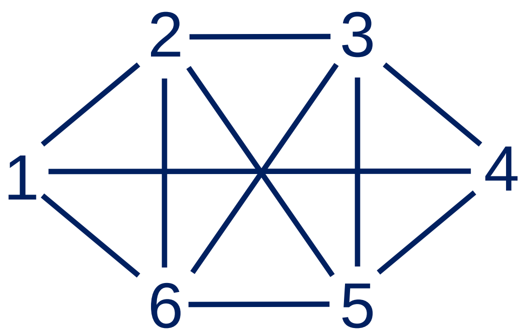




Width = 1

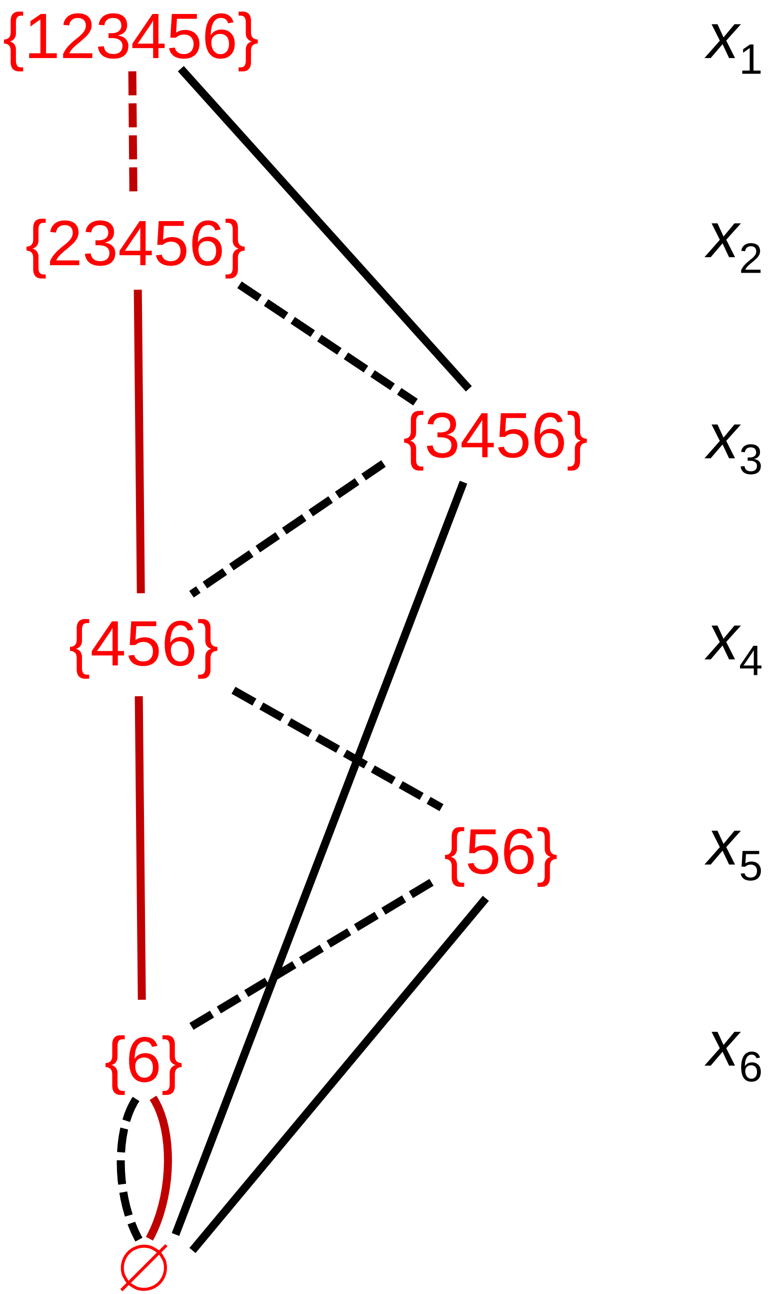
Represents 18 solutions, including 11 feasible solutions





Width = 1

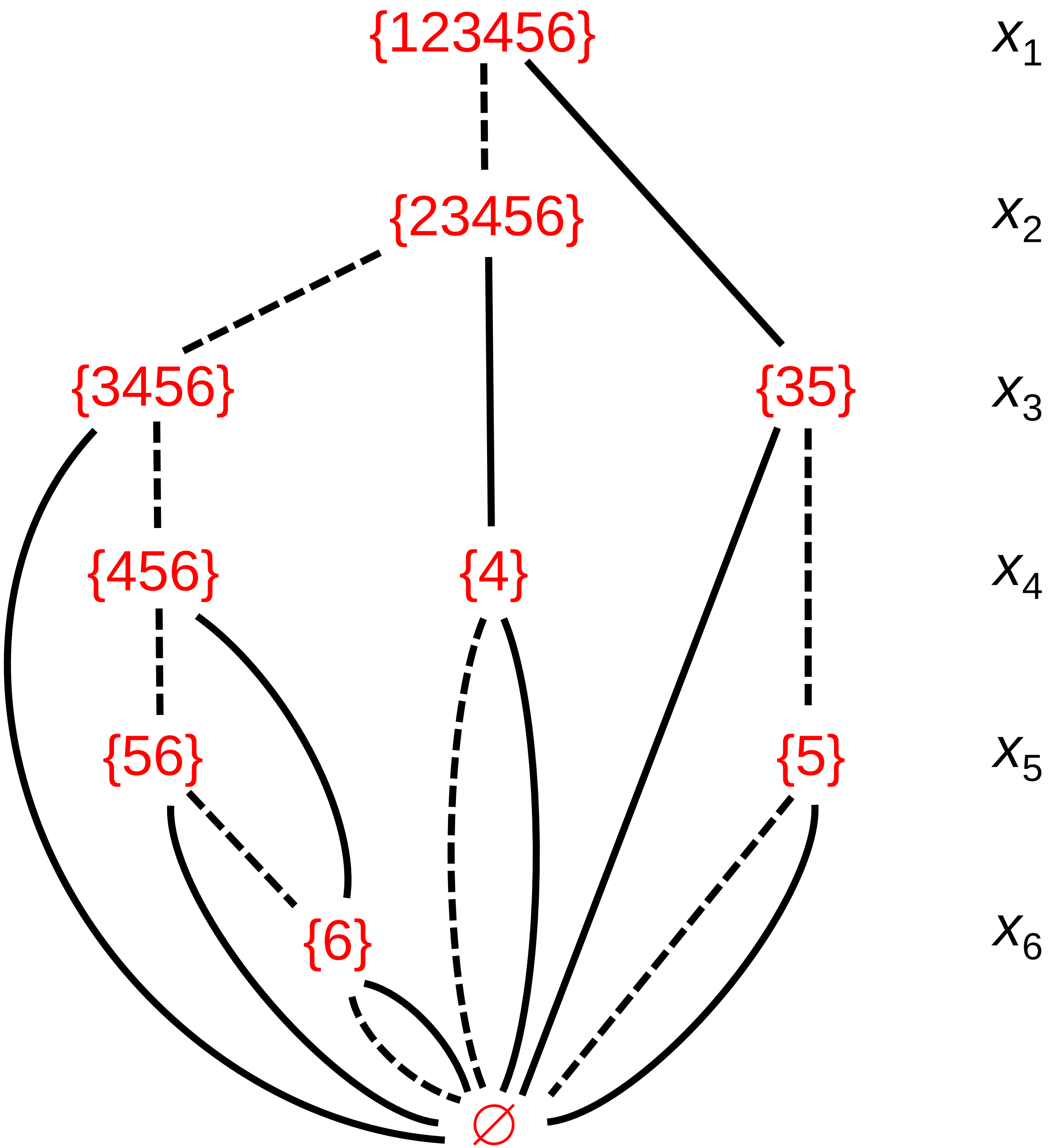
Longest path gives bound of 3 on optimal value of 2



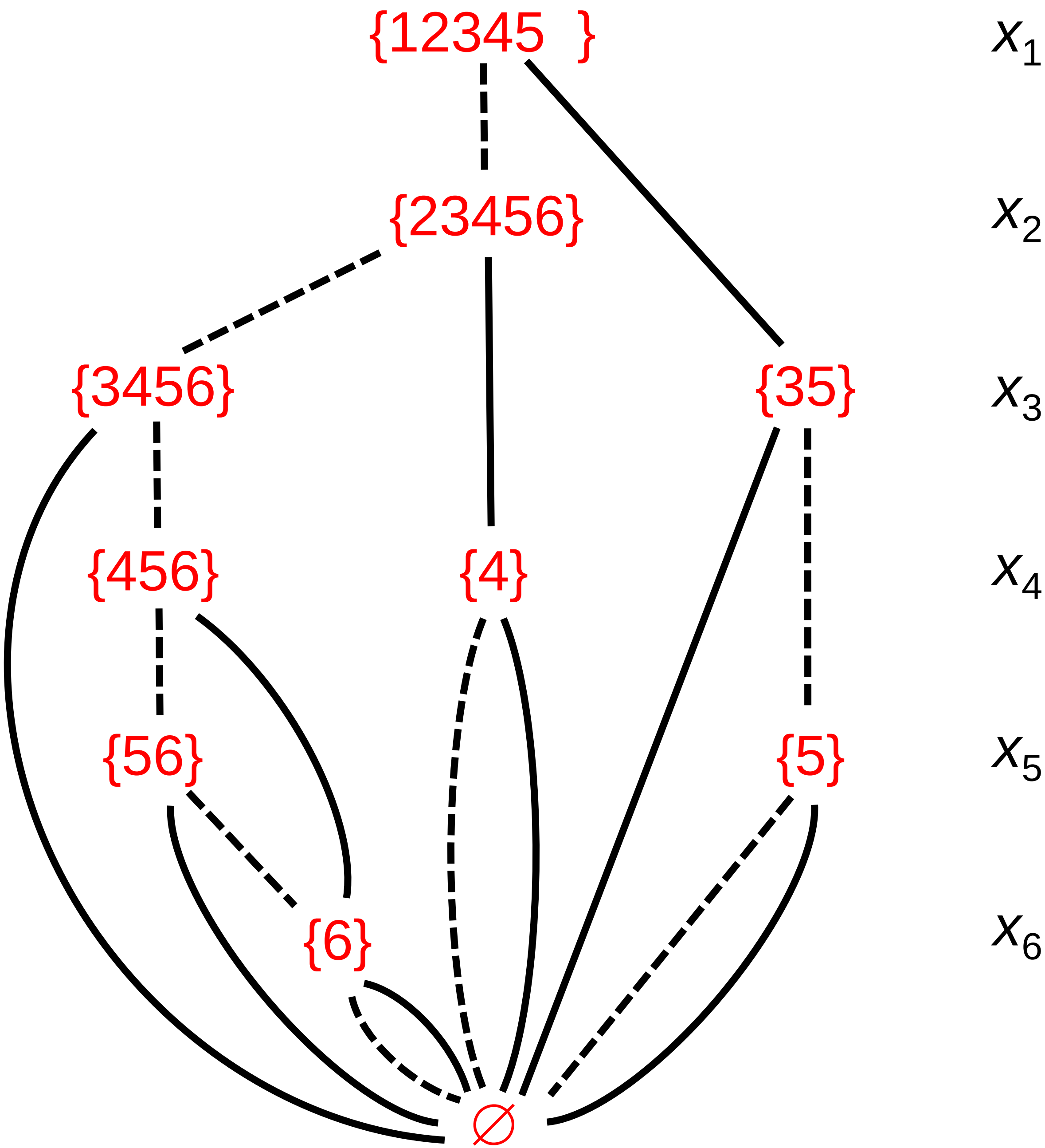
Propagation

- We can propagate by removing arcs from the decision diagram.
  - Rather than removing elements from variable spains.
  - More effective than traditional domain filtering.
  - More information propagated from one constraint to the next.

Suppose this is the relaxed decision diagram

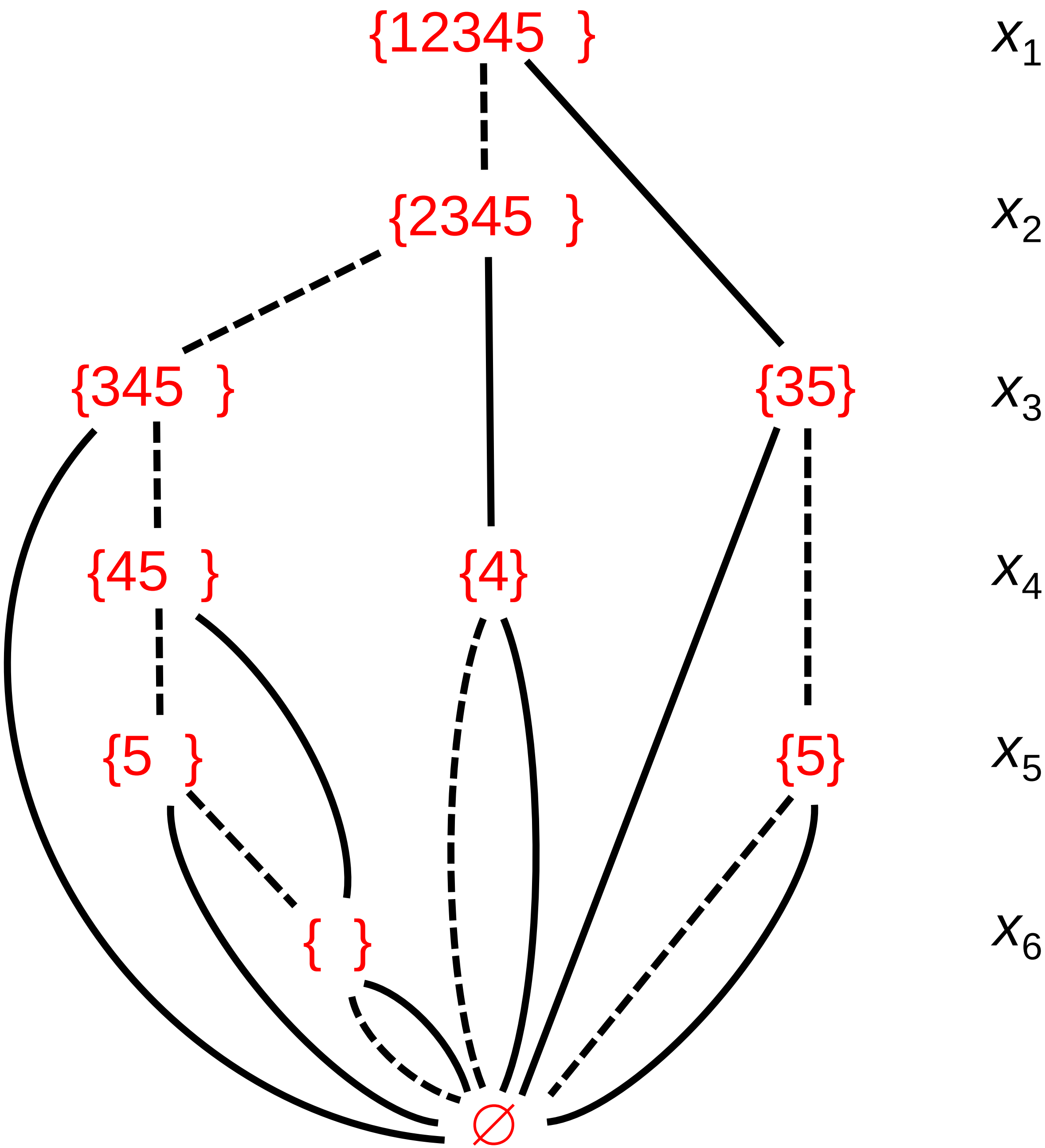


Suppose this is the relaxed decision diagram



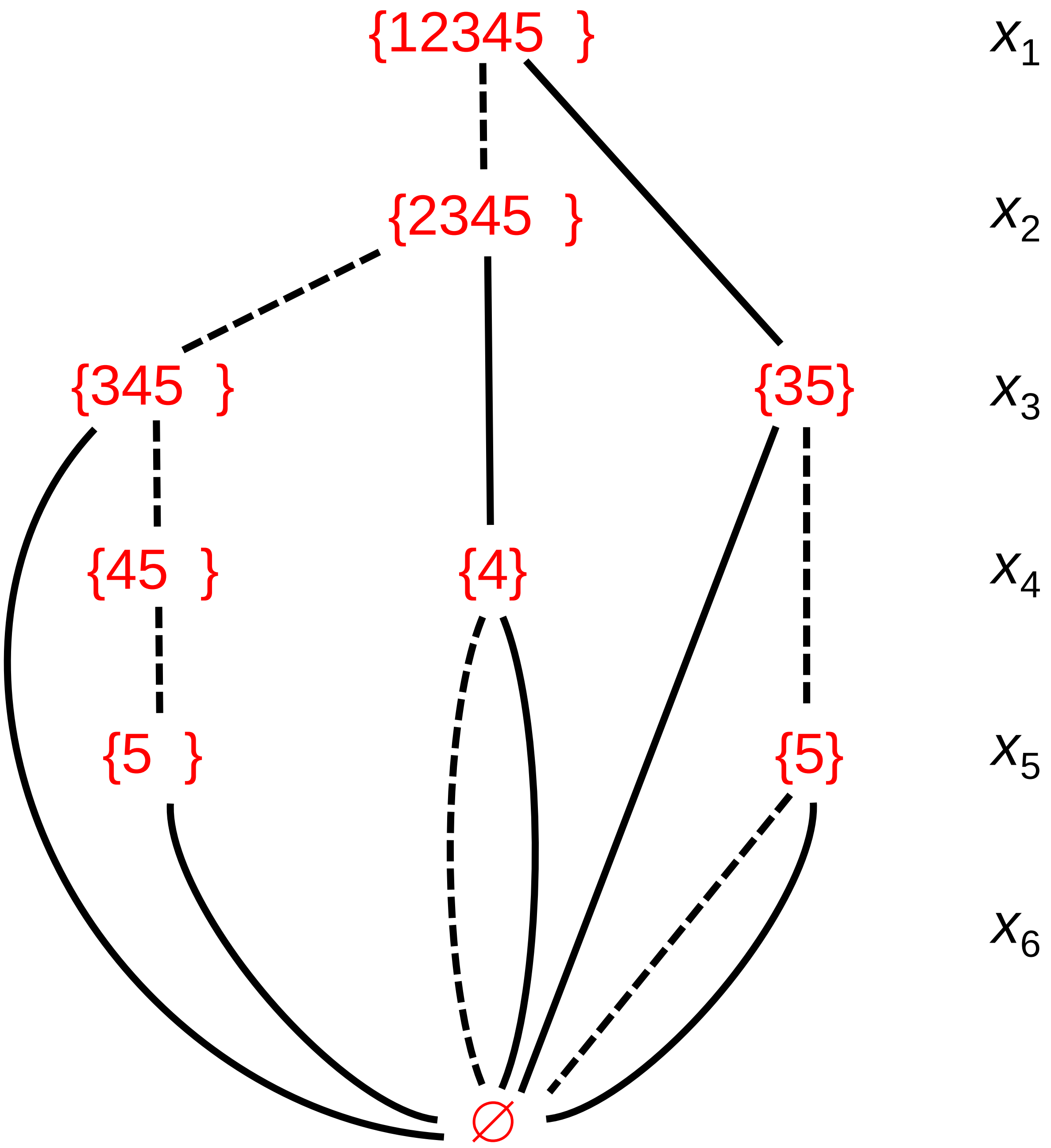
Suppose other constraints remove 6 from domain of x_1

Suppose this is the relaxed decision diagram



This propagates through the states and removes some arcs.

Suppose this is the relaxed decision diagram



This propagates through the states and removes some arcs.

Variable Ordering

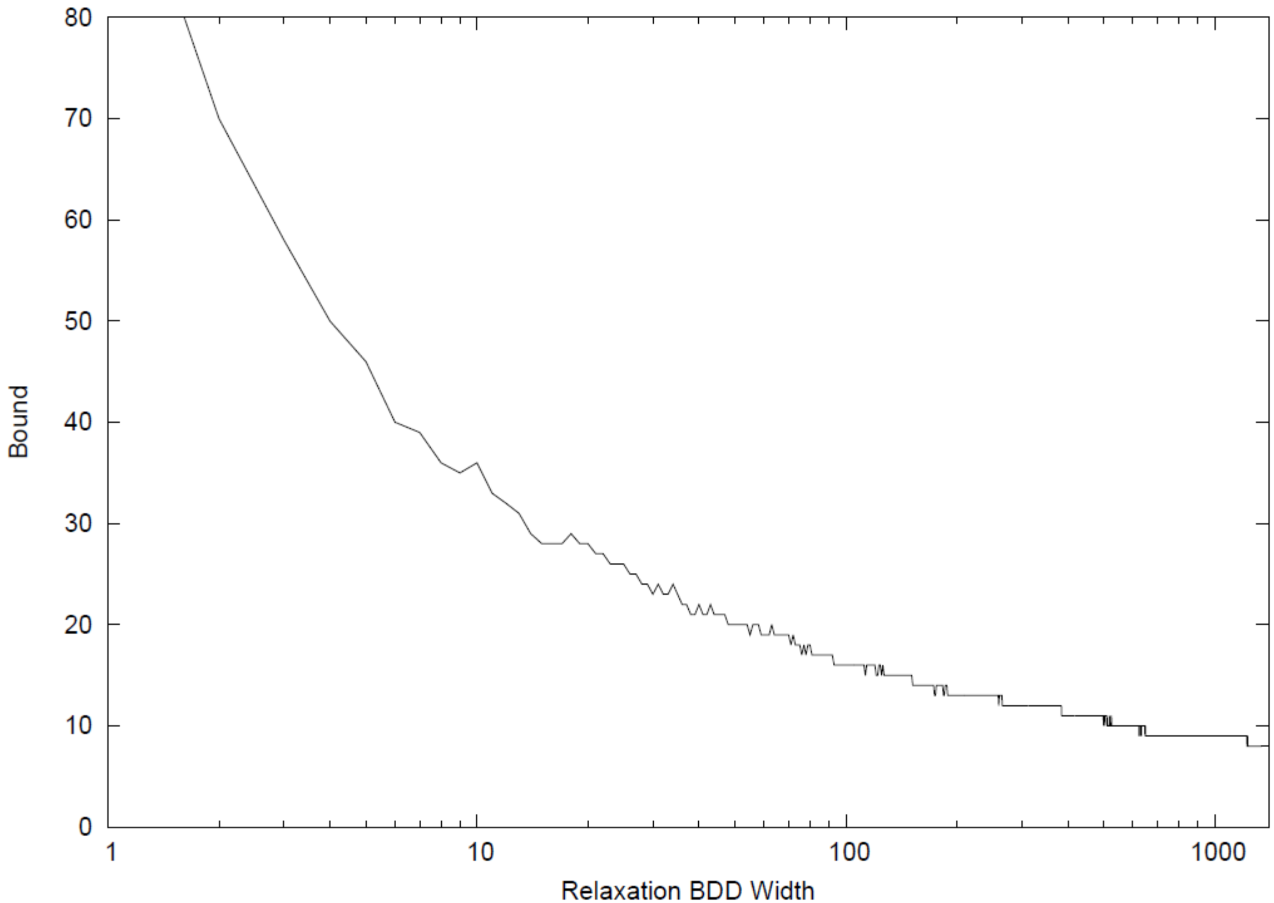
- Variable ordering is key.
  - Just as in branching methods.
  - Some orderings provide much tighter bounds.
- Width of exact decision diagram is bounded by Fibonacci numbers.
  - For an ordering induced by maximal path decomposition of the graph.
- We used a dynamic ordering heuristic.
  - Next variable is the one appearing in the smallest number of states on the current level.
    - Better than maximal path ordering.

Merging Heuristic

- Which nodes to merge when building relaxation?
- A longest path heuristic seems by far the best.
  - Order nodes on current layer by increasing length of longest path into each node.
  - Merge nodes in this order.
  - We lose information in areas not likely to be part of the solution.
- This is better than merging nodes with more vertices in the corresponding states.

Width of Decision Diagram

- Wider BDDs yield tighter bounds.
  - But take longer to build.



Comparison with LP Bound

- Random and benchmark instances
- Compare with LP bound at root node in CPLEX
  - Use clique cover IP formulation
    - Requires precomputing clique cover.

$$\max \sum_i w_i y_i$$

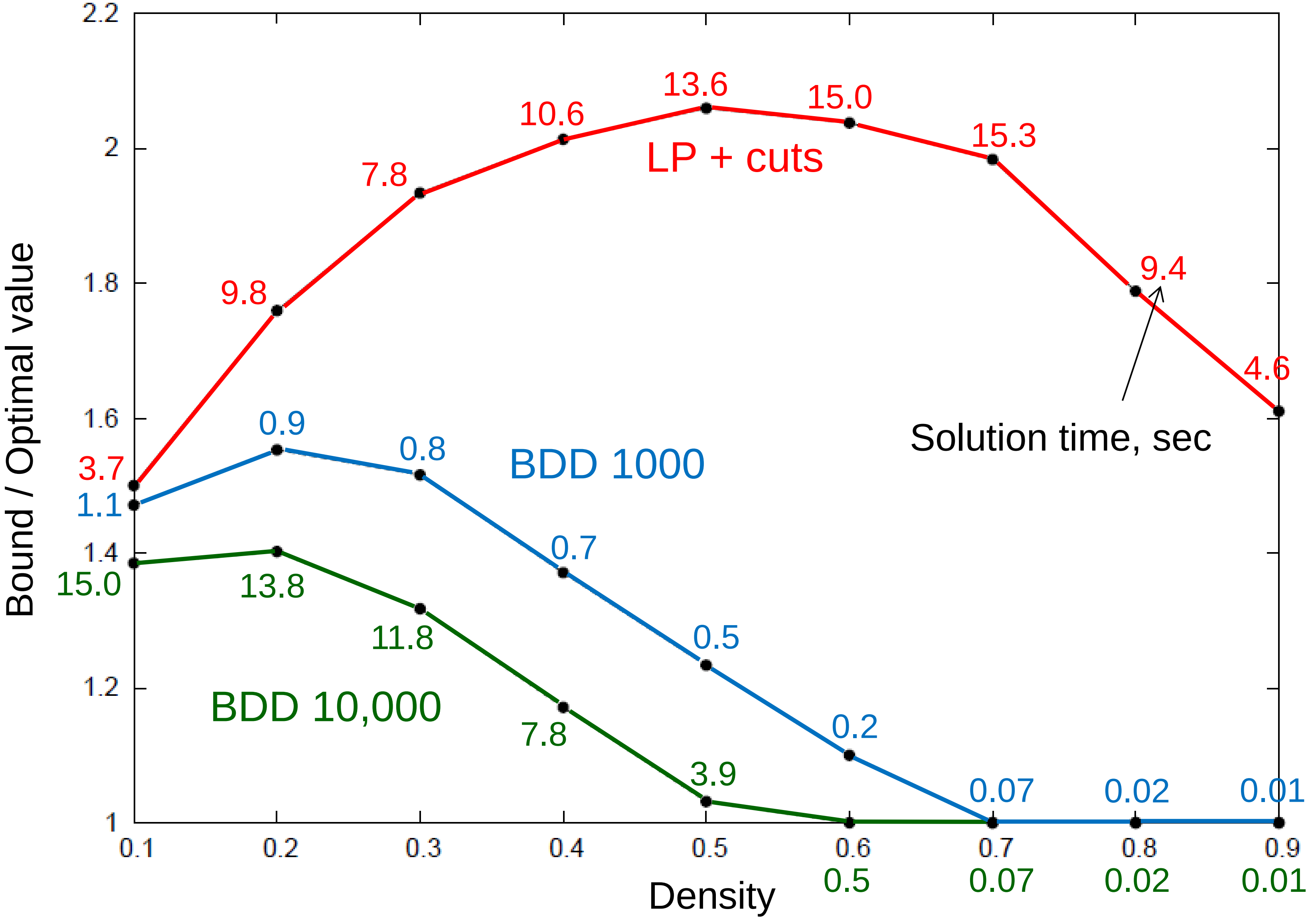
$$\sum_{i \in C_k} y_i \leq 1, \quad \text{all cliques } C_k \text{ in clique cover}$$

$$y_i \in \{0,1\}, \quad \text{all } i$$

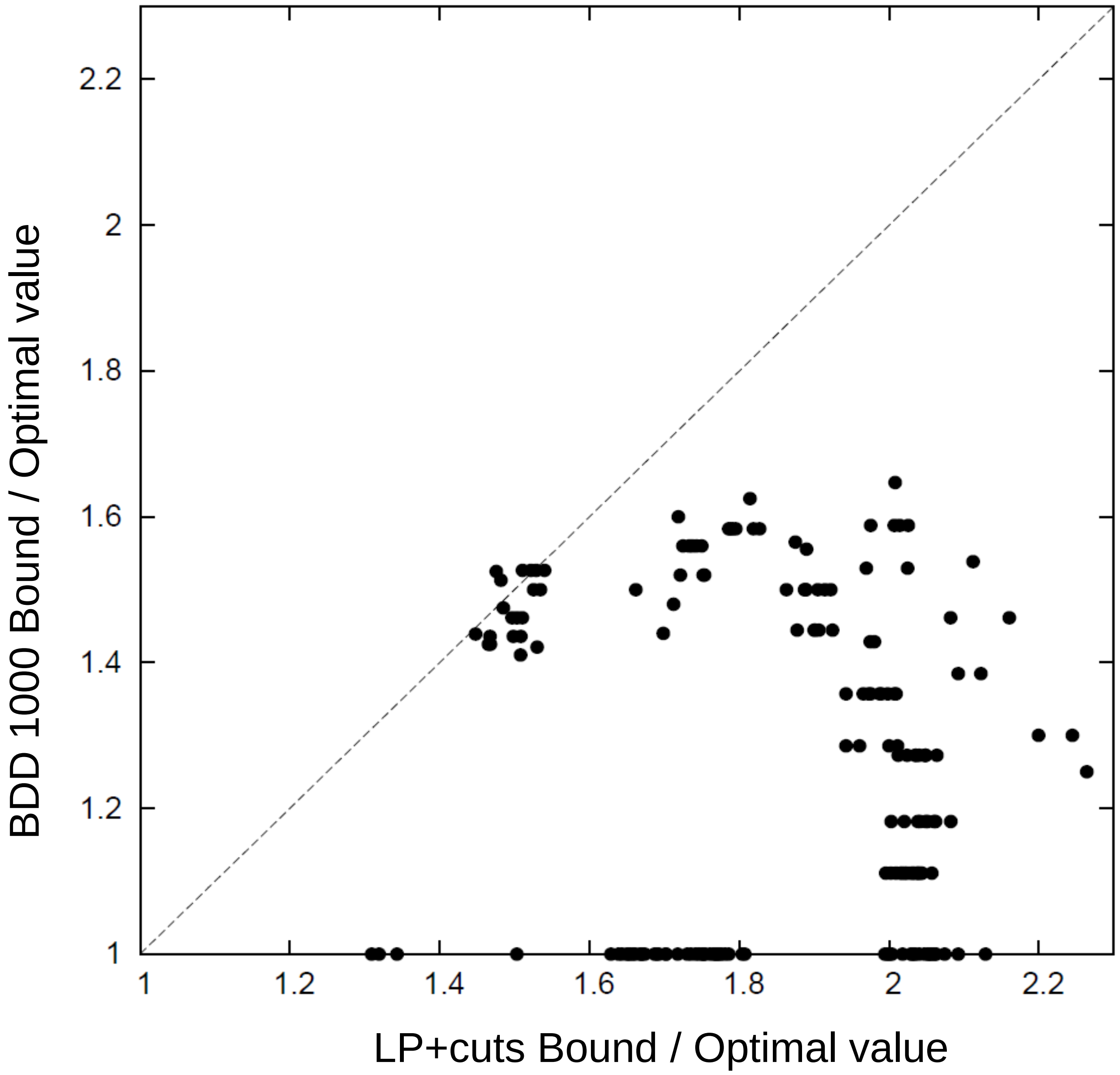
Comparison with LP Bound

- CPLEX settings
  - Turn off presolve
    - It makes CPLEX bound worse or at most 1 better.
    - BDD bounds don’t use presolve.
  - Use full cutting plane resources
  - Use interior point (barrier) LP solver
    - Faster than simplex on these instances.

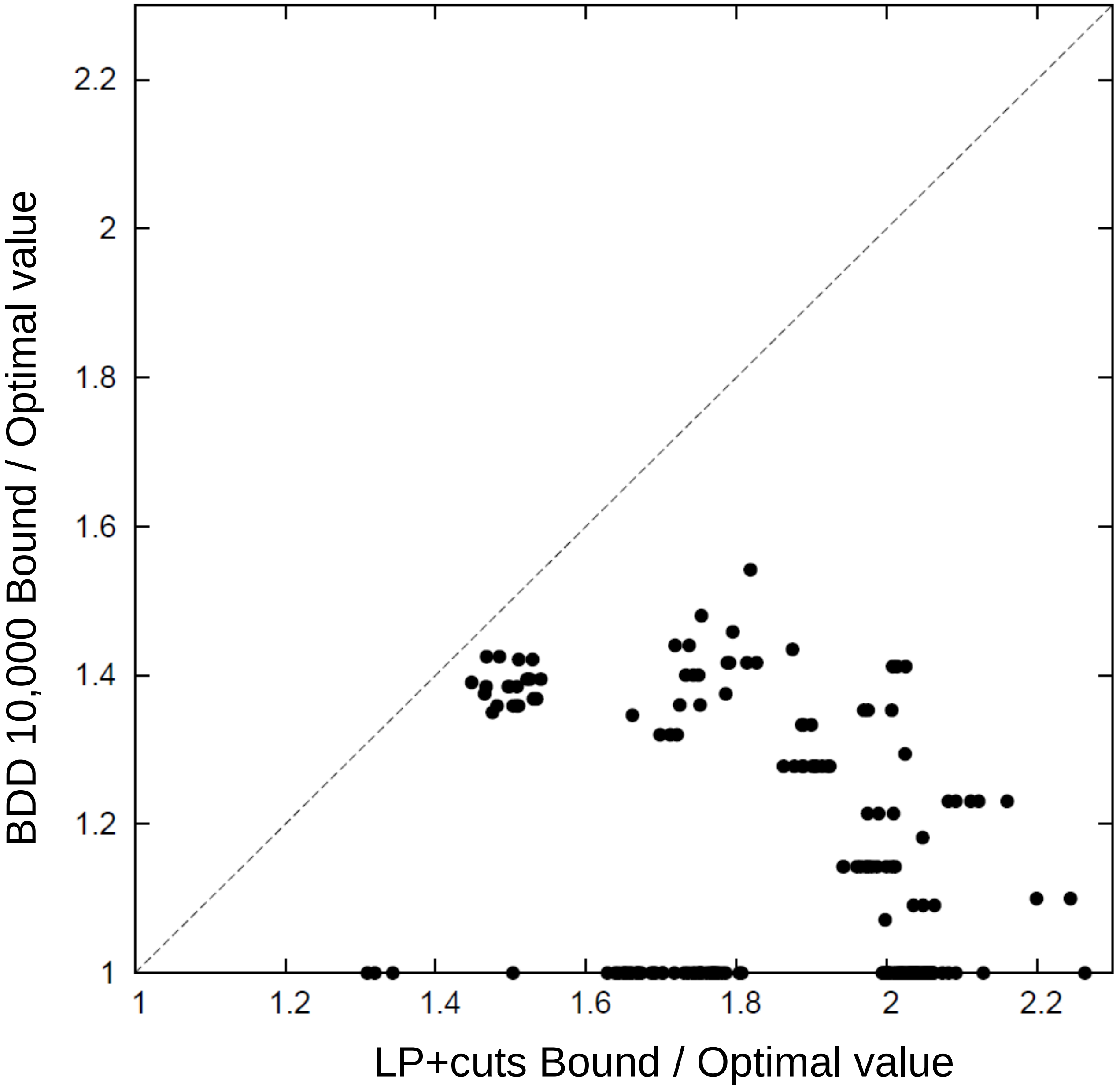
Random instances



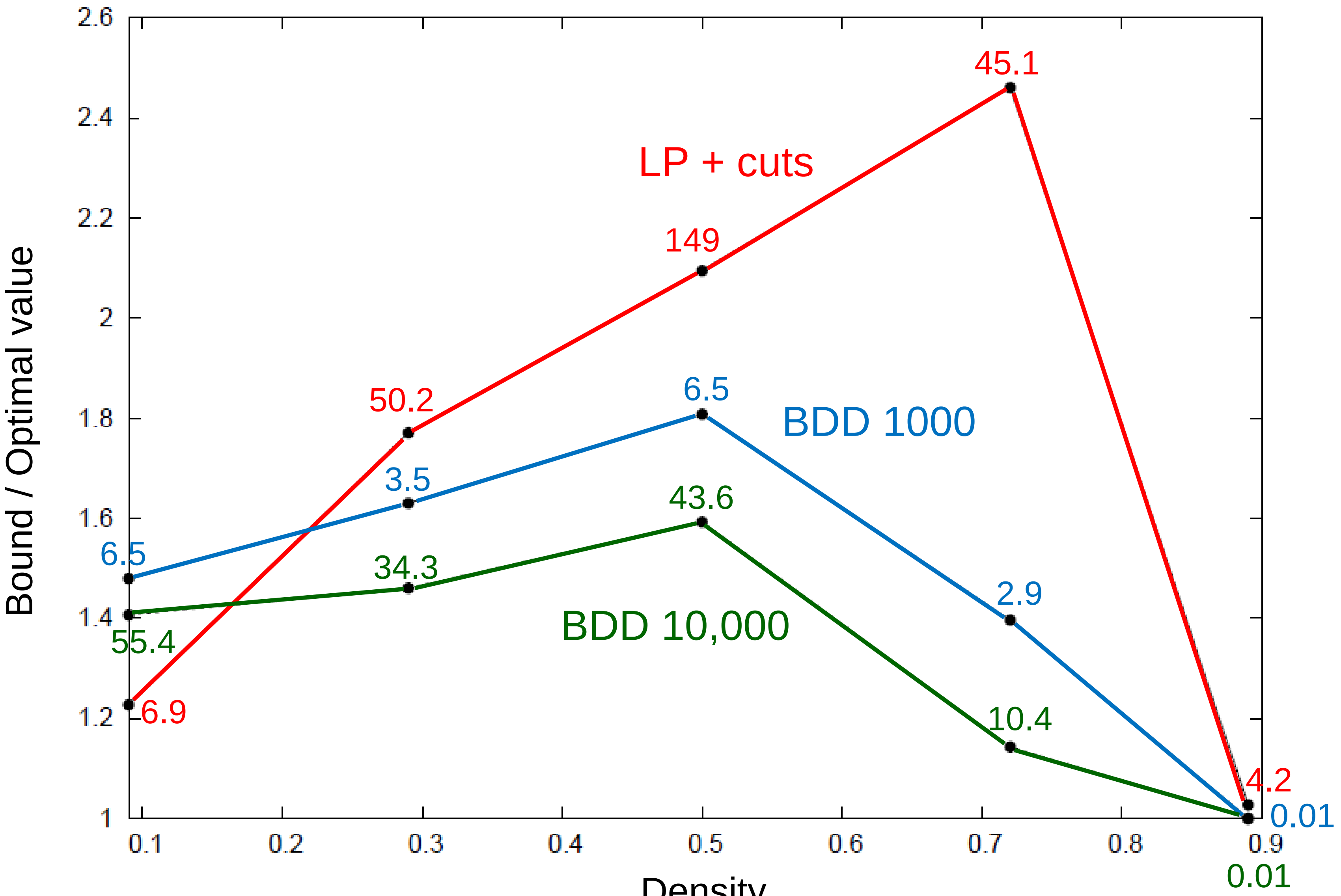
Random instances



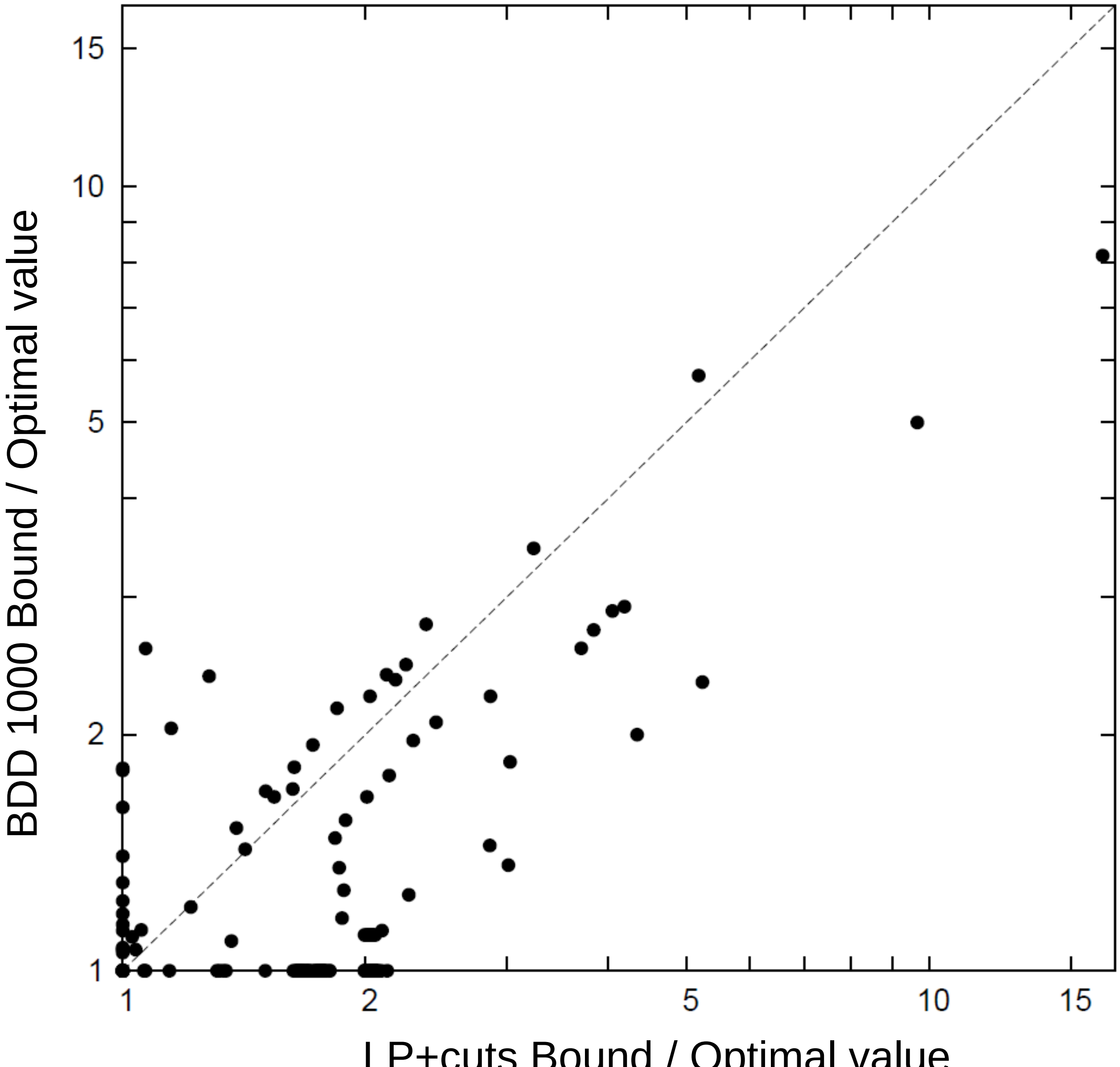
Random instances



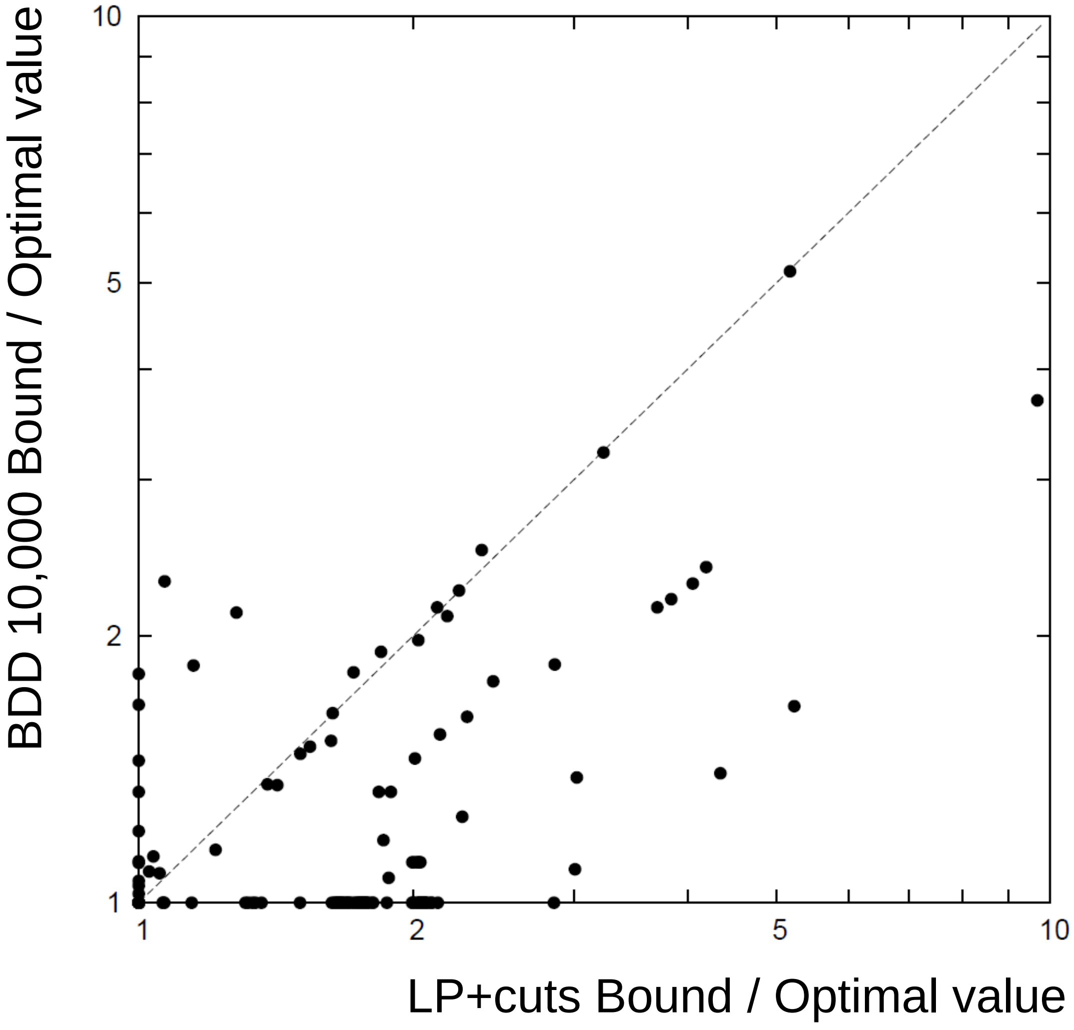
DIMACS instances



DIMACS instances



DIMACS instances



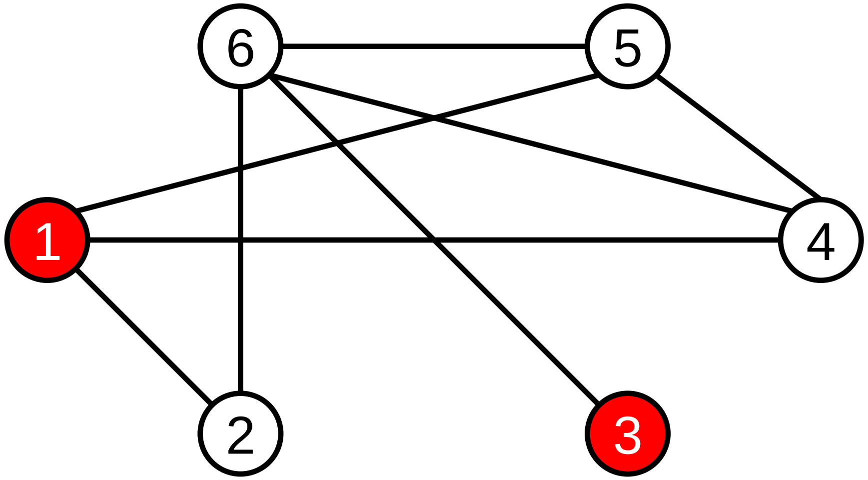
Incrementality

- Fast incremental calculation of bound.
  - Modify relaxed BDD after variable is fixed in branching tree (fast).
  - Recompute longest path (fast).
- However, may be better to rebuild relaxation from scratch.
  - Research issue.

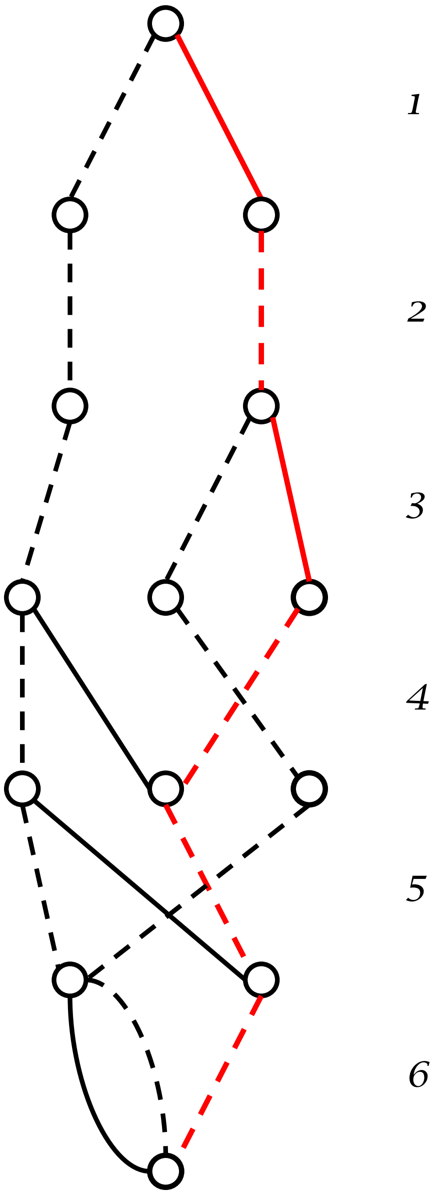
Primal Heuristic

- Use **restricted** decision diagram**.**
  - All paths are feasible.
  - Longest path is a good feasible solution
  - Reduce width by intelligently removing nodes.

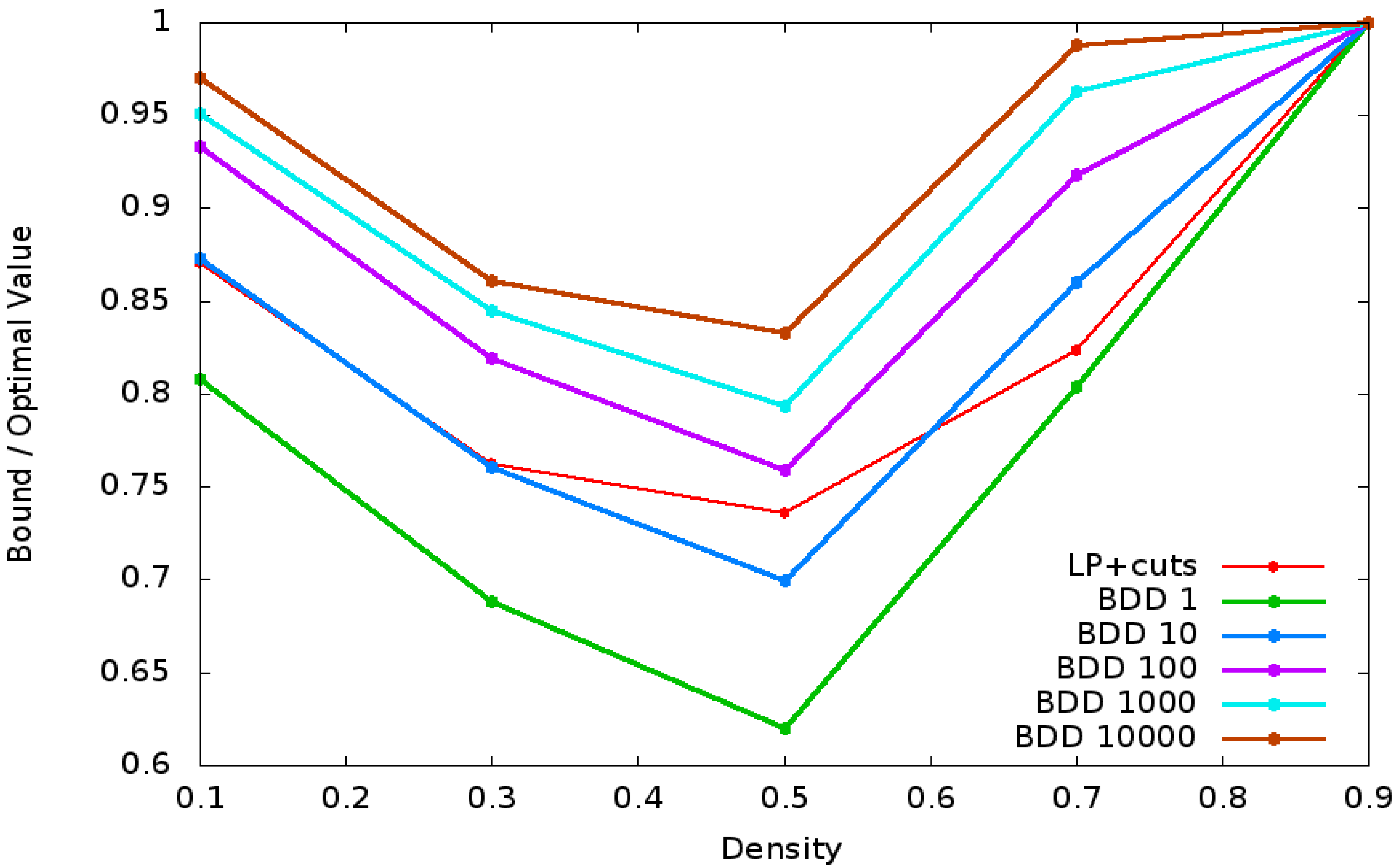
Restricted BDD
Width = 3



Longest path is
feasible solution
and gives upper bound
of **2** on optimal value of **3**



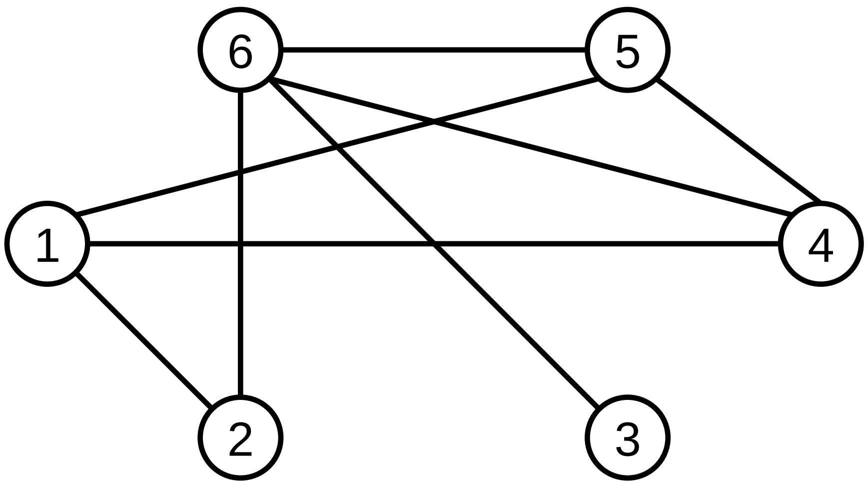
DIMACS instances

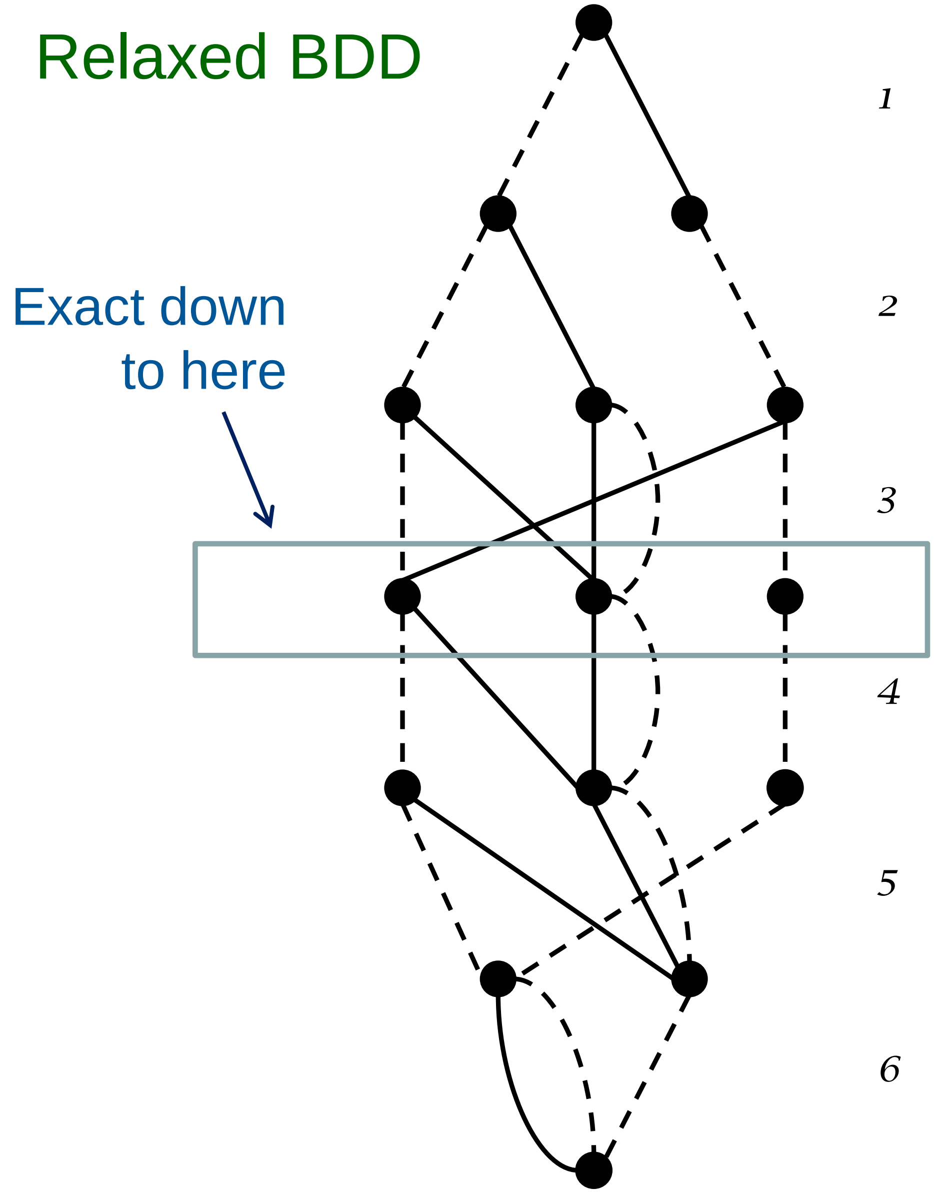


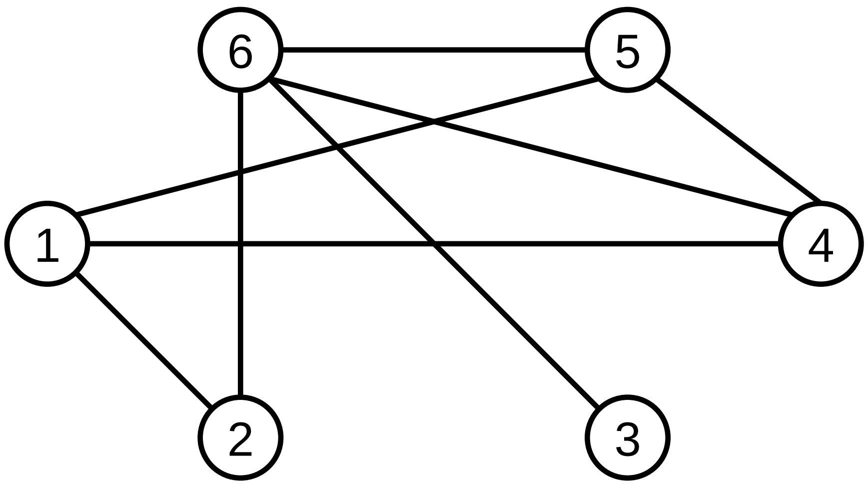
Branching

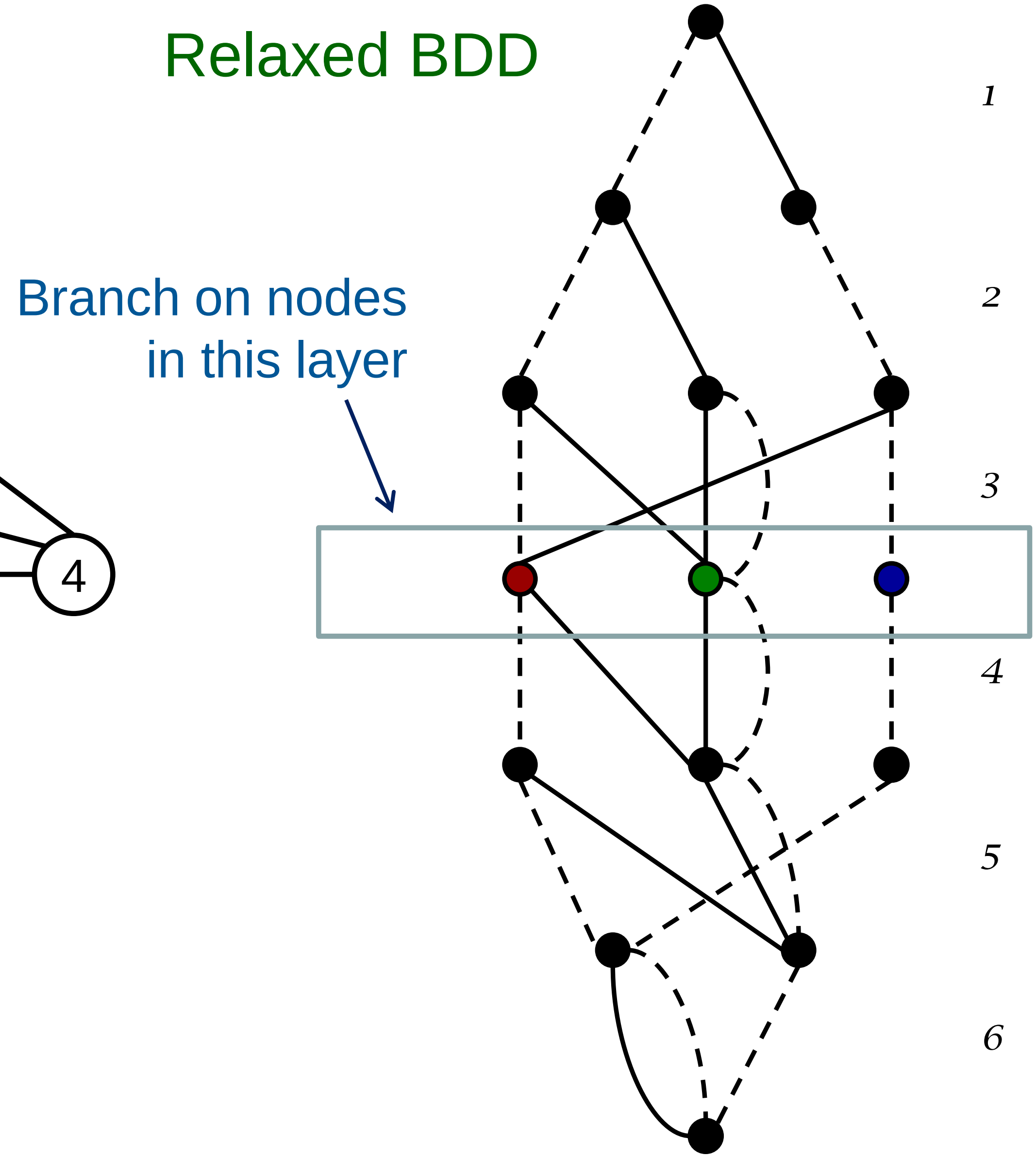
- Novel branching scheme
  - Branch in relaxed decision diagram.
  - Branch on pools of partial solutions.
  - Reduce symmetry.

Relaxed BDD

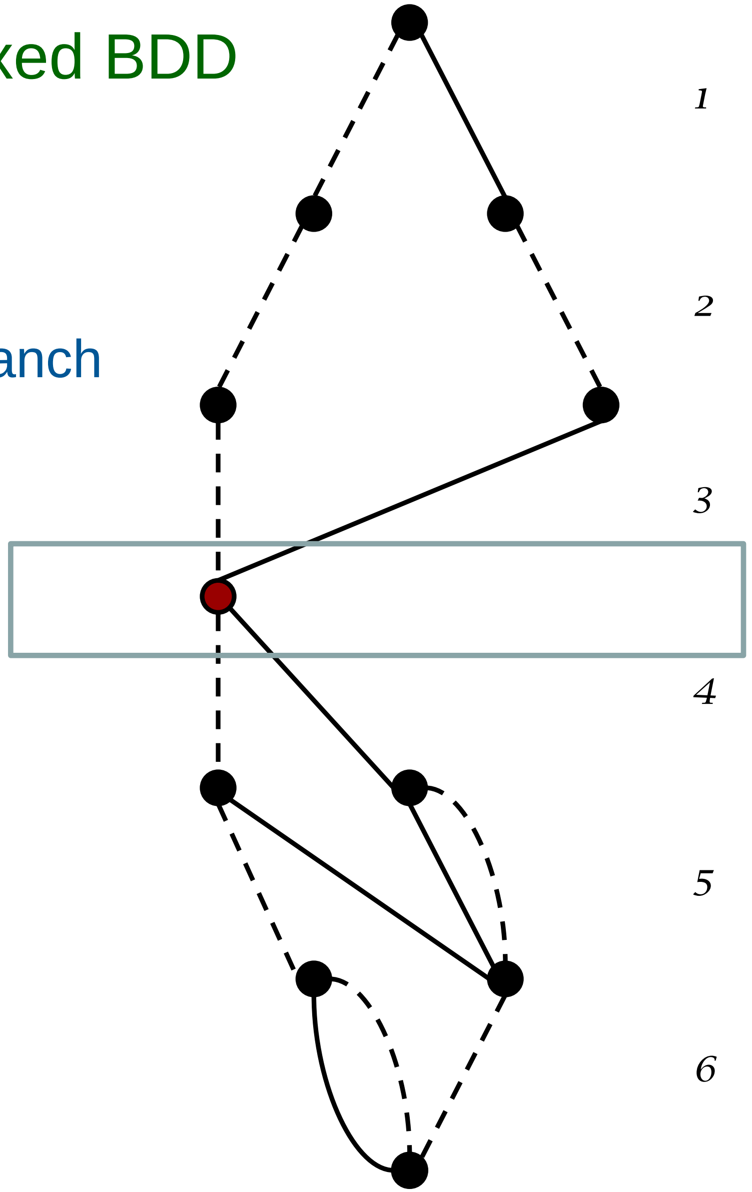




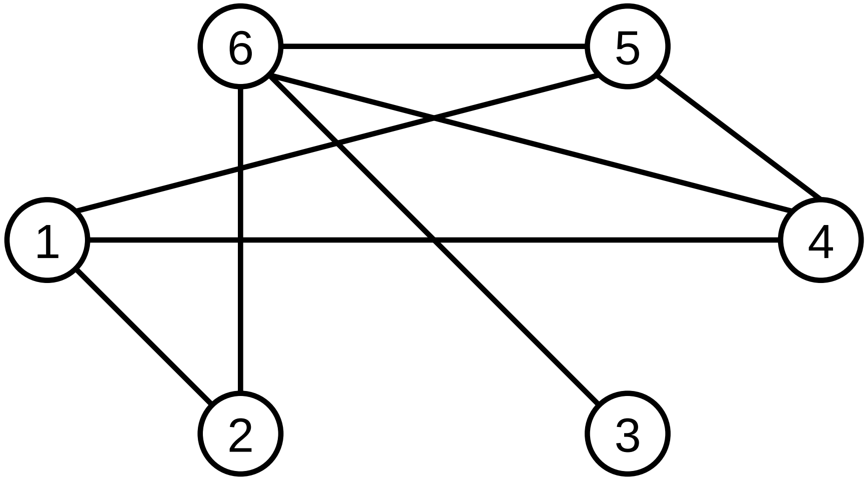

6
5
1
4
2
3


Relaxed BDD
Branch on nodes
in this layer
1
2
3
4
5
6

Relaxed BDD

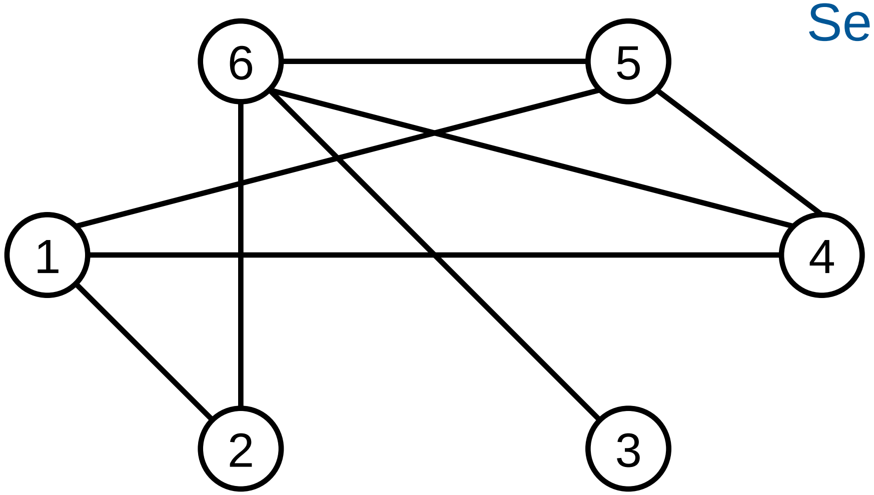


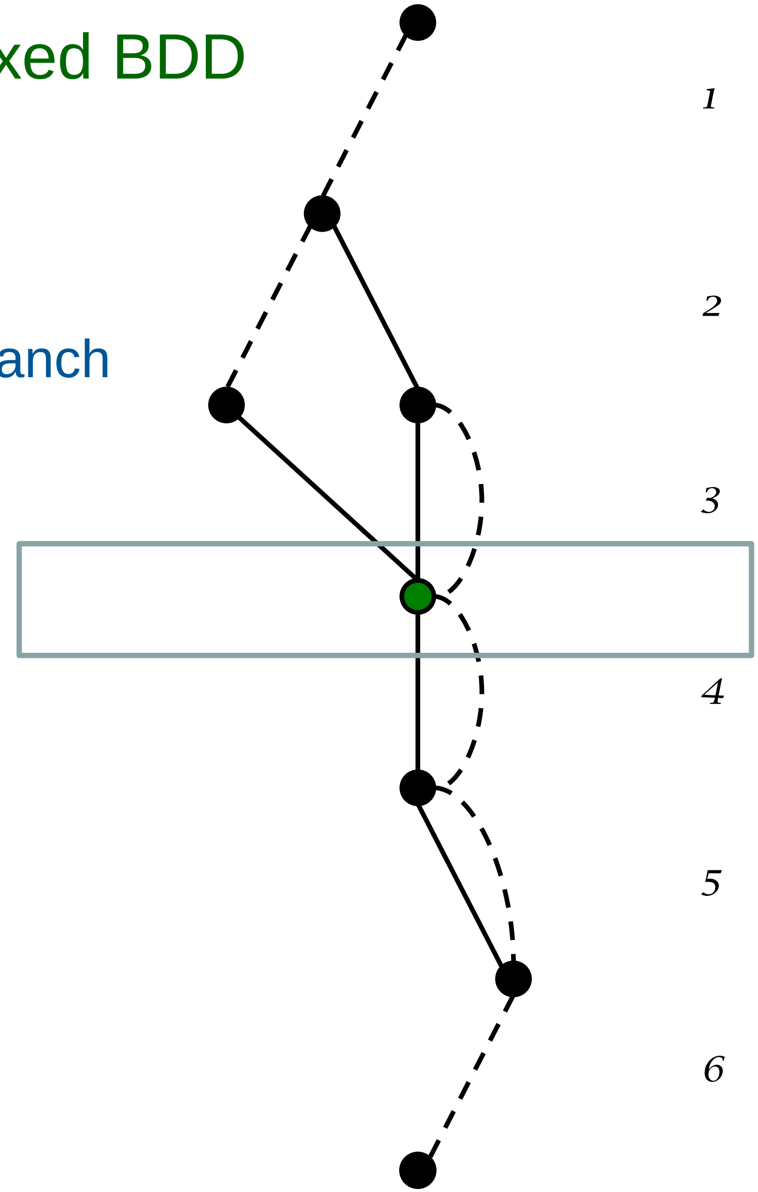
First branch

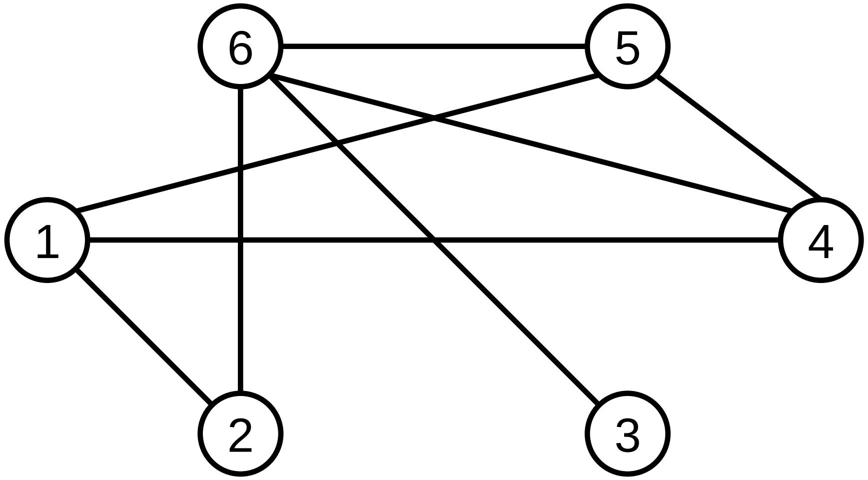


Relaxed BDD

Second branch

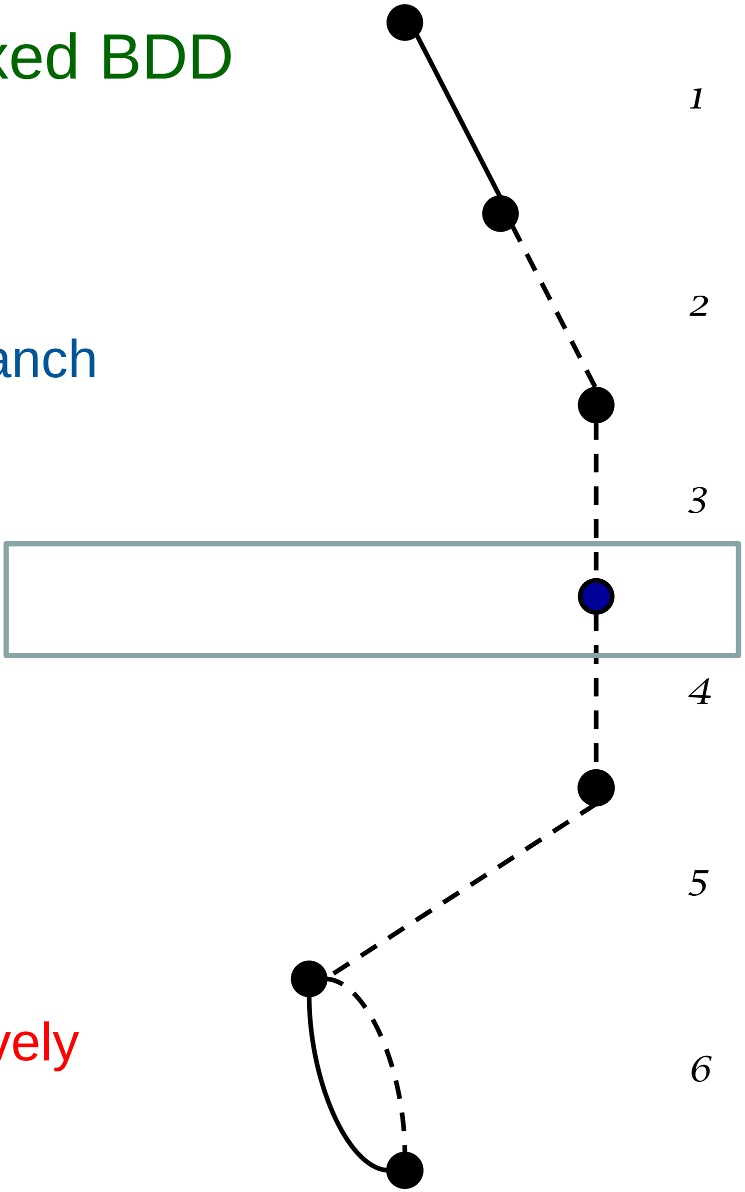






Relaxed BDD

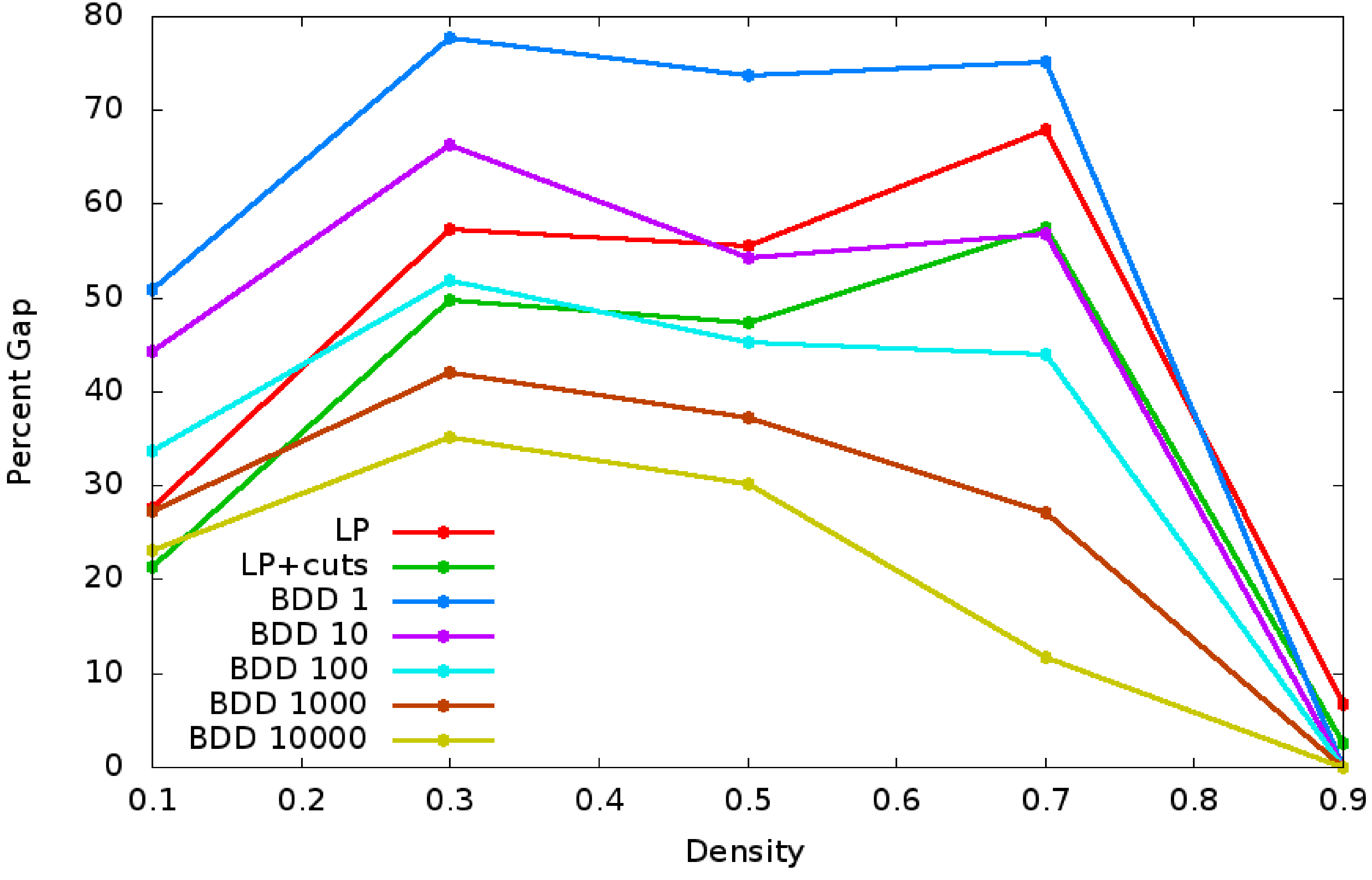
Third branch



Continue recursively

DIMACS instances

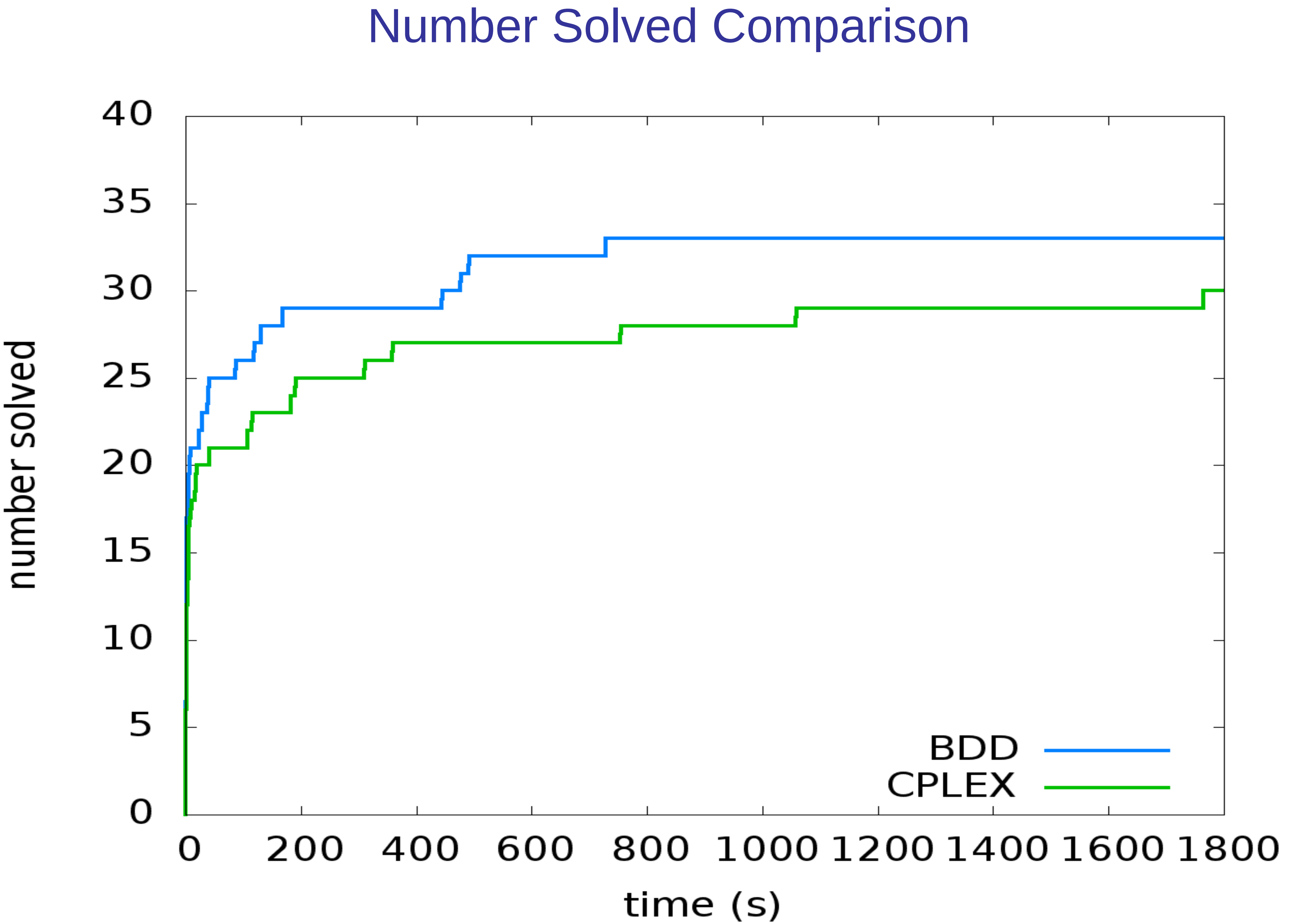
Root Node Gap Comparison

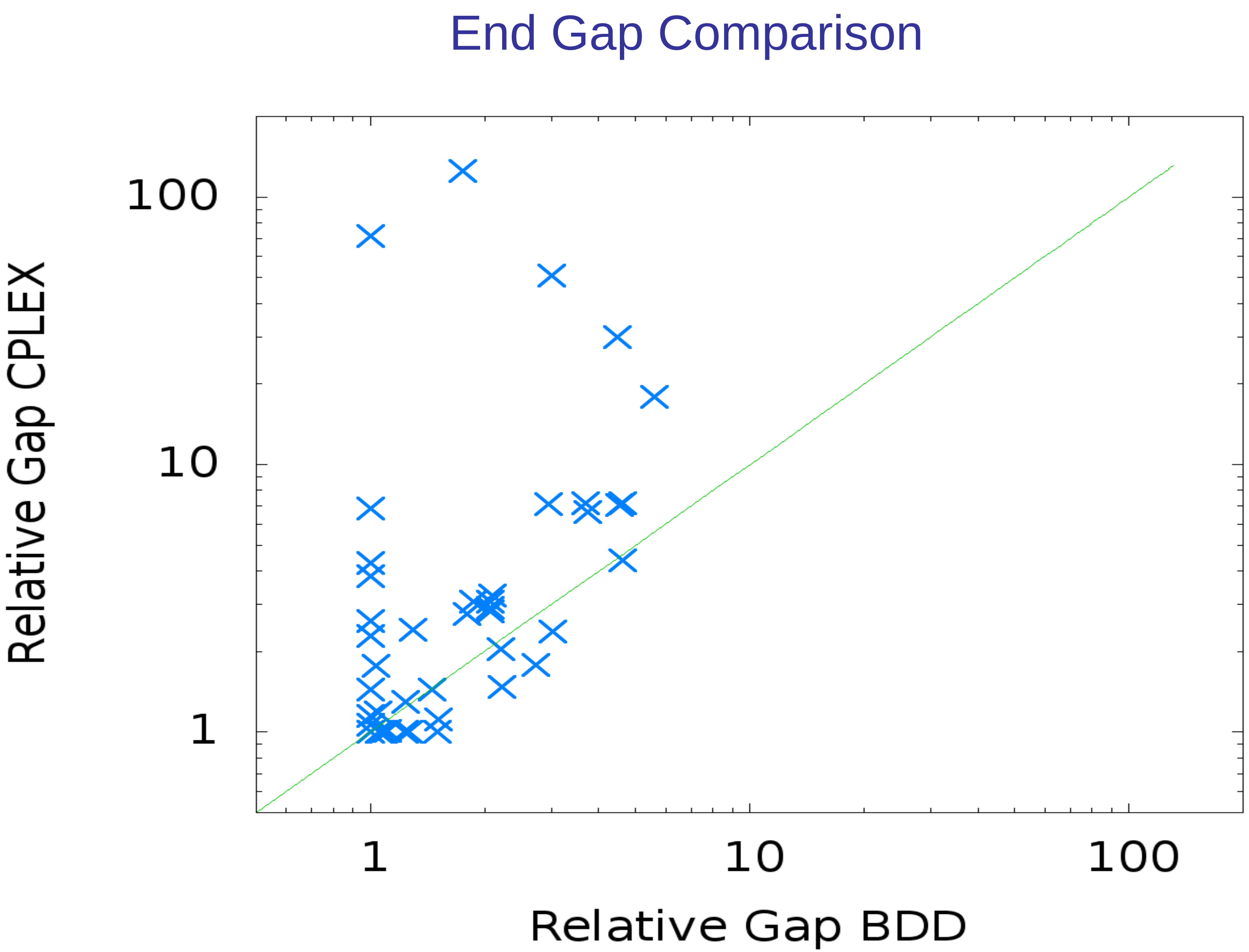


Put it All Together

- General solver
  - Bounds from relaxed decision diagram
  - Primal heuristic from restricted decision diagram
  - Propagation in relaxed decision diagram (not used here)
  - Formulation with state variables
  - Branching in decision diagram
- Compare with CPLEX
  - Independent set problem

DIMACS instances




End Gap Comparison
Relative Gap CPLEX
100
10
1
1
10
100
Relative Gap BDD

Ongoing Research

- Additional problems
  - Set covering
  - MaxSAT
  - Minimum bandwidth (linear arrangement)
- Omit complicating variables from relaxed decision diagram
  - Reduce coupling from constraints
  - Decision diagram becomes much tighter relaxation