# Bounds from Binary Decision Diagrams

David Bergman
Joint Work With: Andre Cire, Willem-Jan van Hoeve, J.N. Hooker

Carnegie Mellon Tepper School of Business

## Maximum Independent Set Problem

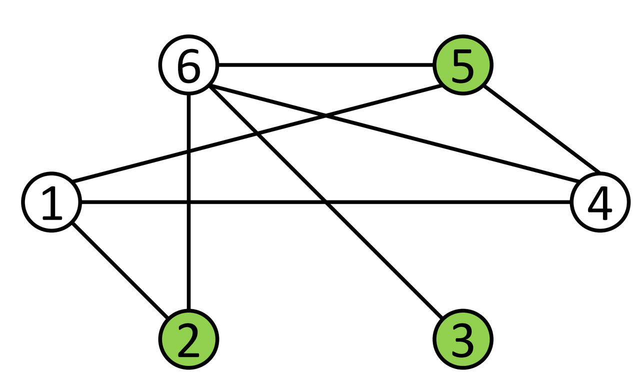
- Input: Undirected graph : $G = (V, E)$
- Problem: Find maximum cardinality subset of vertices, no two of which are adjacent
- Integer Programming Formulation:

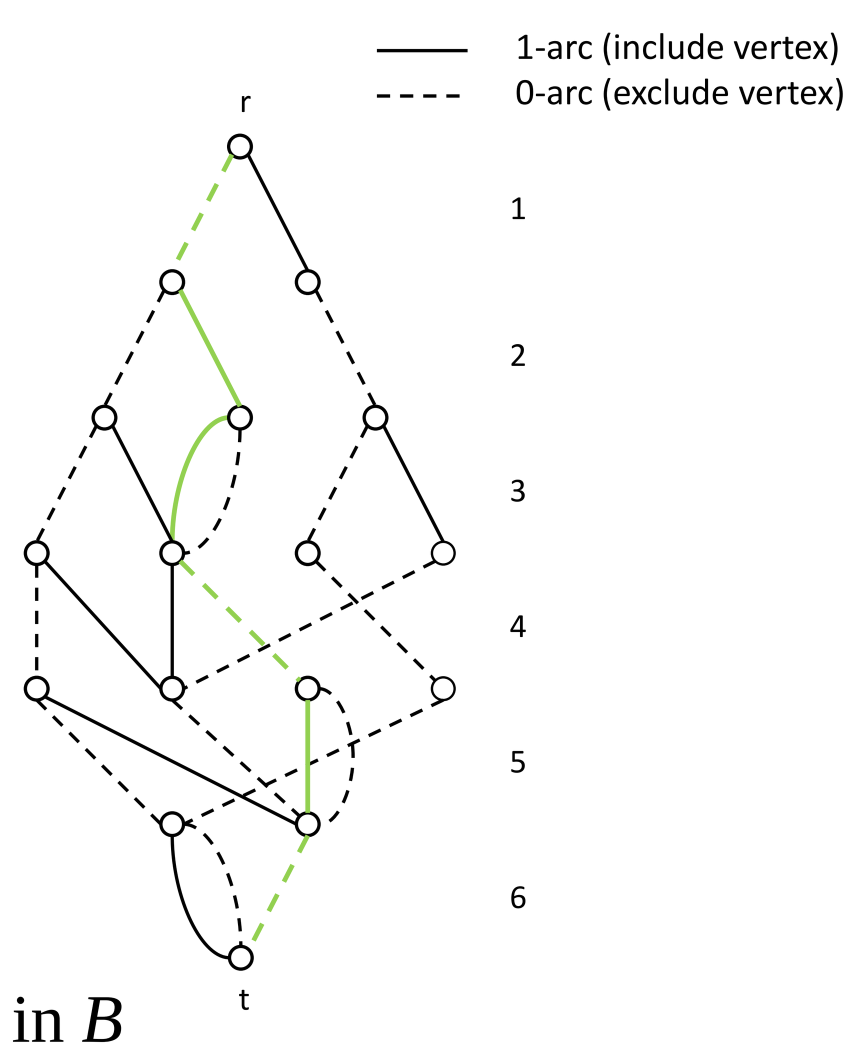
$$\begin{aligned} \max \quad & \sum_{j=1}^{n} x_j \\ \text{s.t.} \quad & x_i + x_j \leq 1 \quad \forall\, (i,j) \in E \\ & x_j \in \{0,1\} \quad \forall\, j \in V \end{aligned}$$

- Let $I(G)$ be the family of independent sets and $z^*(G)$ be the size of the maximum cardinality independent set in $G$

## Binary Decision Diagrams (BDDs)

- A BDD $B = (U,A)$ is a layered directed acyclic multi-graph used to represent a set of solutions to binary optimization problems



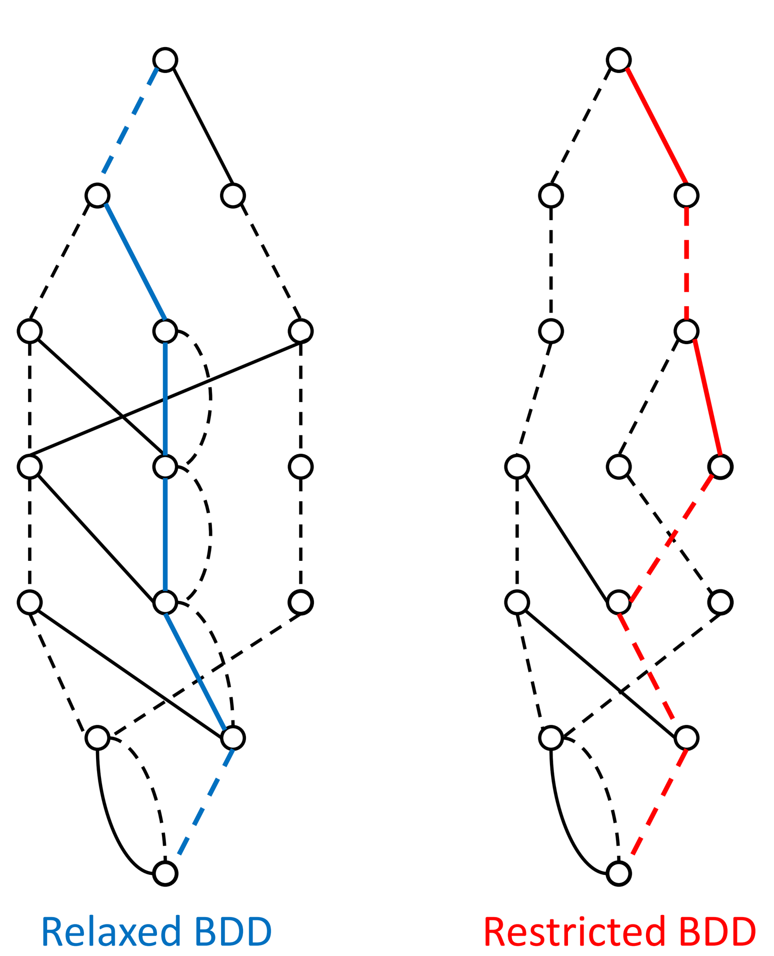

- Let Sol($B$) be the set of solutions in $B$
- $B$ is called *exact* if Sol($B$) = $I(G)$
- For nodes $u,v$ in $B$, we let $B_{uv}$ be the subgraph of $B$ induced by the nodes in $U$ that lie on some directed path from $u$ to $v$
- Assigning *arc-costs* 1 to *1-arcs* and 0 to *0-arcs*, the longest path LP($B_{rt}$) in an exact BDD corresponds to a largest cardinality independent set: LP($B_{rt}$) = $z^*(G)$

## Approximate BDDs

- Exact BDDs can have exponential size
- Limit the width of BDD to create *approximate* BDDs which provides *bounds* on $z^*(G)$

- *Relaxed* BDD $B$
  - Sol($B$) $\supset I(G)$
  - LP($B_{rt}$) $\geq z^*(G)$

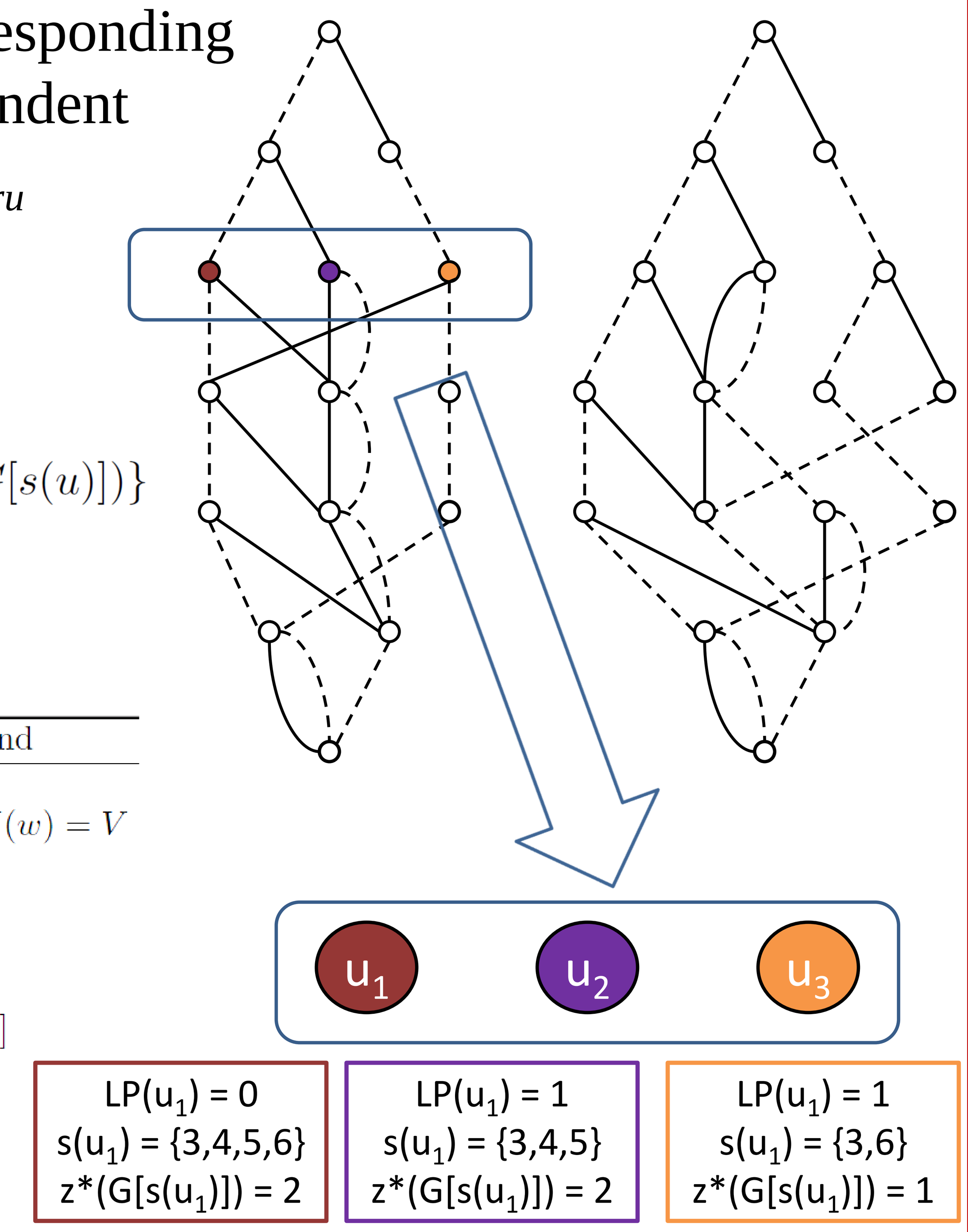
- *Restricted* BDD $B$
  - Sol($B$) $\subset I(G)$
  - LP($B_{rt}$) $\leq z^*(G)$




## Branch and Bound

- For a relaxed BDD $B$, the *Last Exact Layer* $L^*(B)$ is the deepest layer that coincides exactly with the exact BDD
- For every node $u$ in $L^*(B)$, associate the state $s(u)$ corresponding to the set of vertices independent of all independent sets in $B_{ru}$
- Theorem:

$$z^*(G) = \max_{u \in L^*(G)} \{\mathrm{LP}(B_{r,u}) + z^*(G[s(u)])\}$$




z*(G) = max {0 + 2, 1 + 2 , 1 + 1} = 3

**Algorithm 1** BDD-Based Branch and Bound

$z_{UB} = 0$
create search tree node $w$, $\mathrm{LP}(w) = 0, V(w) = V$
$S = \{w\}$
**while** $S \neq \emptyset$ **do**
  select node $w \in S$
  $S \leftarrow S \setminus \{w\}$
  build restricted BDD $B^{\text{res}}$ on $G[V(w)]$
  **if** $\mathrm{LP}(B^{\text{res}}_{r,t}) > z_{UB}$ **then**
    $z_{UB} = \mathrm{LP}(B^{\text{res}}_{r,t})$
  build relaxed BDD $B^{\text{rel}}$ on $G[V(w)]$
  **if** $\mathrm{LP}(B^{\text{rel}}_{r,t}) > z_{UB}$ **then**
    **for all** $u \in L^*(B^{\text{rel}})$ **do**
      $\mathrm{LP}(u) = \mathrm{LP}(B^{\text{rel}}_{r,u}), V(u) = s(u)$
      $S \leftarrow S \cup \{u\}$
return $z_{UB}$

## Computational Results

- Compared BDD-based branch and bound algorithm with state-of-the-art integer programming solver CPLEX
- Instances are complements of maximum clique DIMACS benchmark set taken from http://cs.hbg.psu.edu/txn131/clique.html