Mixed Integer Programming

Mixed Integer Programming (MIP)

Changing the rules of business™

Minimize $c^T x$

Subject to $Ax = b$

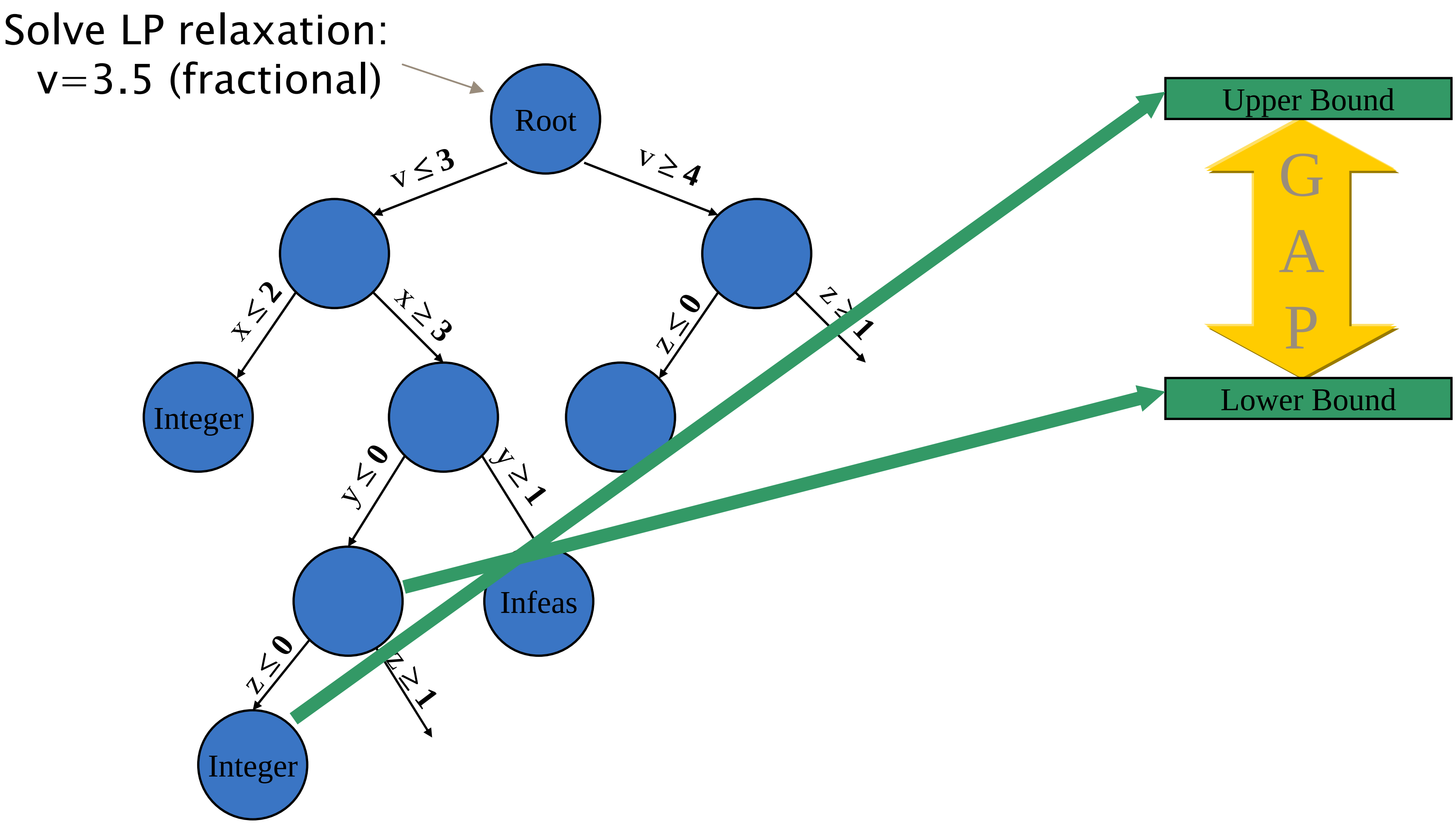
$l \leq x \leq u$

Some x_j are integer

Integrality Restriction

Branch and Bound

Solving MIPs: Branch-and-Bound



Remarks:

(1) GAP = 0 $\Rightarrow$ Proof of optimality

(2) In practice: Often good enough to have good Solu

Important Steps

Important Steps

Changing the rules of business™

The branch and bound loop

- Choose an unexplored node in the tree
- Solve continuous relaxation
- Strengthen formulation
  - Generate *cutting planes*
  - Perform variable fixing
- Find integer feasible solutions that are "similar" to the relaxation solution ("Heuristics")
- Choose a variable on which to branch
- Explore logical implications of branch
- Repeat

Important Steps

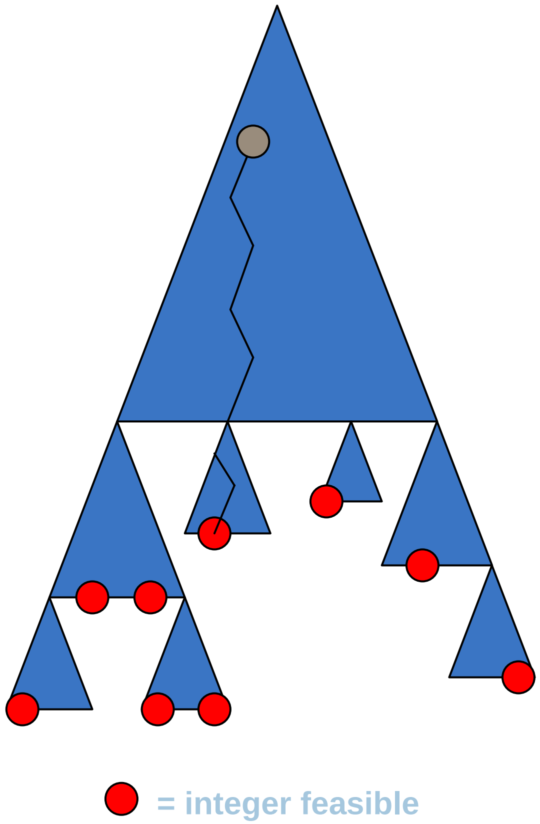
Changing the rules of business™

The branch and bound loop

- **Choose an unexplored node in the tree**
- Solve relaxation
- Generate *cutting planes*
- Perform variable fixing
- Find integer feasible solutions that are "similar" to the relaxation solution
- Choose a variable on which to branch
- Explore logical implications of branch
- Repeat

Node Selection

Tradeoff: feasibility versus optimality



- **When exploring nodes deep in the search tree…**
  - **More likely to find integer feasible solutions**
  - **More likely to explore nodes that would be pruned by later feasible solutions**

Node Selection Options

Changing the rules of business™

Options

- Depth first
- Breadth first
- Best first
  - Weight objective and # integer infeasibilities
- Best estimate
- Plunging (combined with above)
  - Always choose a child of the previously explored node
  - Probed dive

Important Steps

The branch and bound loop

- Choose an unexplored node in the tree
- **Solve relaxation**
- Generate *cutting planes*
- Perform variable fixing
- Find integer feasible solutions that are “similar” to the relaxation solution
- Choose a variable on which to branch
- Explore logical implications of branch
- Repeat

Node Relaxation Solution

Ideally suited to dual simplex

- Change from parent relaxation is small : a new bound on the branching variable
  - Previous basis remains dual feasible
  - Solution likely to be “close” to previous basis
- A few iterations of dual simplex typically suffice to restore optimality
- Cost per node quite low

Important Steps

Changing the rules of business™

The branch and bound loop

- Choose an unexplored node in the tree
- Solve relaxation
- **Generate *cutting planes***
- Perform variable fixing
- Find integer feasible solutions that are "similar" to the relaxation solution
- Choose a variable on which to branch
- Explore logical implications of branch
- Repeat

Important Steps

The branch and bound loop

- Choose an unexplored node in the tree
- Solve relaxation
- Generate *cutting planes*
- Perform variable fixing
- Find integer feasible solutions that are "similar" to the relaxation solution
- Choose a variable on which to branch
- Explore logical implications of branch
- Repeat

Reduced Cost Fixing

Use reduced costs to fix variables

- Recall: reduced cost D_N is the marginal cost of moving a variable off of its bound
- If $z_{lp} + |D_j| \geq z^*$
  - z^* = objective of best known feasible solution (incumbent)
- Then x_j can be fixed to its current value in this subtree (“region”)

Important Steps

Changing the rules of business™

The branch and bound loop

- Choose an unexplored node in the tree
- Solve relaxation
- Generate *cutting planes*
- Perform variable fixing
- ***Find integer feasible solutions that are "similar" to the relaxation solution***
- Choose a variable on which to branch
- Explore logical implications of branch
- Repeat

Important Steps

The branch and bound loop

- Choose an unexplored node in the tree
- Solve relaxation
- Generate *cutting planes*
- Perform variable fixing
- Find integer feasible solutions that are "similar" to the relaxation solution
- **Choose a variable on which to branch**
- Explore logical implications of branch
- Repeat

Variable Selection

Greatly affects search tree size

- **Guiding principles:**
  - **Make important decisions early**
  - **Both directions of branch should have an impact**
- **Example:**
  - **Decide whether or not to build a factory first**
  - **Decide how many lines to place in the factory later**

Variable Selection

Predicting impact

- **Question:**
  - **How to predict impact of a branch?**
- **Possible answers:**
  - **Find variables that are furthest from their bounds**
    - Maximum infeasibility
  - **Measure the impact for each branching candidate**
    - Strong branching [Applegate, Bixby, Chvátal, Cook]
  - **Use historical information**
    - Pseudo-costs

Important Steps

Changing the rules of business™

The branch and bound loop

- Choose an unexplored node in the tree
- Solve relaxation
- Generate *cutting planes*
- Perform variable fixing
- Is the relaxation solution near-feasible?
- Choose a variable on which to branch
- **Explore logical implications of branch**
- Repeat

Logical Propagation

Propagate implications logically

- Simple example:
  - $x + 2y + 3z \leq 3$, all variables binary
  - $x = 1$ (e.g., fixed during tree exploration)
  - $z = 2/3$ still feasible in LP relaxation
- Use *bound strengthening* to tighten variable bounds

Next

- The MIP Landscape
- Heuristic details
- Cutting plane details