Mixed Integer Programming

A Linear Program

```
\ P1 = units of Product 1 produced
\ P2 = units of Product 2 produced
\ P3 = units of Product 3 produced
\ 500 = hours/week of milling machine time available
\ 350 = hours/week of lathe time available
\ 150 = hours/week of grinder time available

Maximize
 Profit: 30 P1 + 12 P2 + 15 P3
Subject To
 Milling_machine: 9 P1 + 3 P2 + 5 P3 <= 500
 Lathe:           5 P1 + 4 P2 <= 350
 Grinder:         3 P1 + 2 P3 <= 150
 Bound:                    P3 <= 20
End
```

Optimal Solution:

Profit = $1742.85

P1 = 26.1905 units

P2 = 54.7619 units

P3 = 20 units

Problem: It doesn't make sense to produce fractional units

Possible solution: Round the answer

P1 = 26, P2 = 54, P3 = 20

Profit = $1728

A Better Solution: Integer Programming

```
\Problem name: HW1_integer.lp

Maximize
 Profit: 30 P1 + 12 P2 + 15 P3
Subject To
 Milling_machine: 9 P1 + 3 P2 + 5 P3 <= 500
 Lathe:           5 P1 + 4 P2 <= 350
 Grinder:         3 P1 + 2 P3 <= 150
Bounds
 0 <= P3 <= 20
Generals
 P1 P2 P3
End
```

Rounded Solution:

Profit = $1728

P1 = 26 units

P2 = 54 units

P3 = 20 units

Optimal Solution:

Profit = $1740

P1 = 26 units

P2 = 55 units

P3 = 20 units

Mixed Integer Programming (MIP)

Minimize $c^T x$

Subject to $Ax = b$

$l \leq x \leq u$

Some x_j are integer

Integrality Restriction

Application of LP & MIP - I

- Transportation-airlines
  - Fleet assignment
  - Crew scheduling
  - Ground personnel scheduling
  - Yield management
  - Fuel allocation
  - Passenger mix
  - Booking control
  - Maintenance scheduling
  - Load balancing/freight packing
  - Airport traffic planning
  - Gate scheduling/assignment
  - Upset recover and management
- Transportation-other
  - Vehicle routing
  - Freight vehicle scheduling and assignment
  - Depot/warehouse location
  - Freight vehicle packing
  - Public transportation system operation
  - Rental car fleet management
- Process industries
  - Plant production scheduling and logistics
  - Capacity expansion planning
  - Pipeline transportation planning
  - Gasoline and chemical blending

Application of LP & MIP - II

- Financial
  - Portfolio selection and optimization
  - Cash management
  - Synthetic option development
  - Lease analysis
  - Capital budgeting and rationing
  - Bank financial planning
  - Accounting allocations
  - Securities industry surveillance
  - Audit staff planning
  - Assets/liabilities management
  - Unit costing
  - Financial valuation
  - Bank shift scheduling
  - Consumer credit delinquency management
  - Check clearing systems
  - Municipal bond bidding
  - Stock exchange operations
  - Debt financing
- Manufacturing
  - Product mix planning
  - Blending
  - Manufacturing scheduling
  - Inventory management
  - Job scheduling
  - Personnel scheduling
  - Maintenance scheduling and planning
  - Steel production scheduling
- Coal Industry
  - Coal sourcing/transportation logistics
  - Coal blending
  - Mining operations management
- Forestry
  - Forest land management
  - Forest valuation models
  - Planting and harvesting models

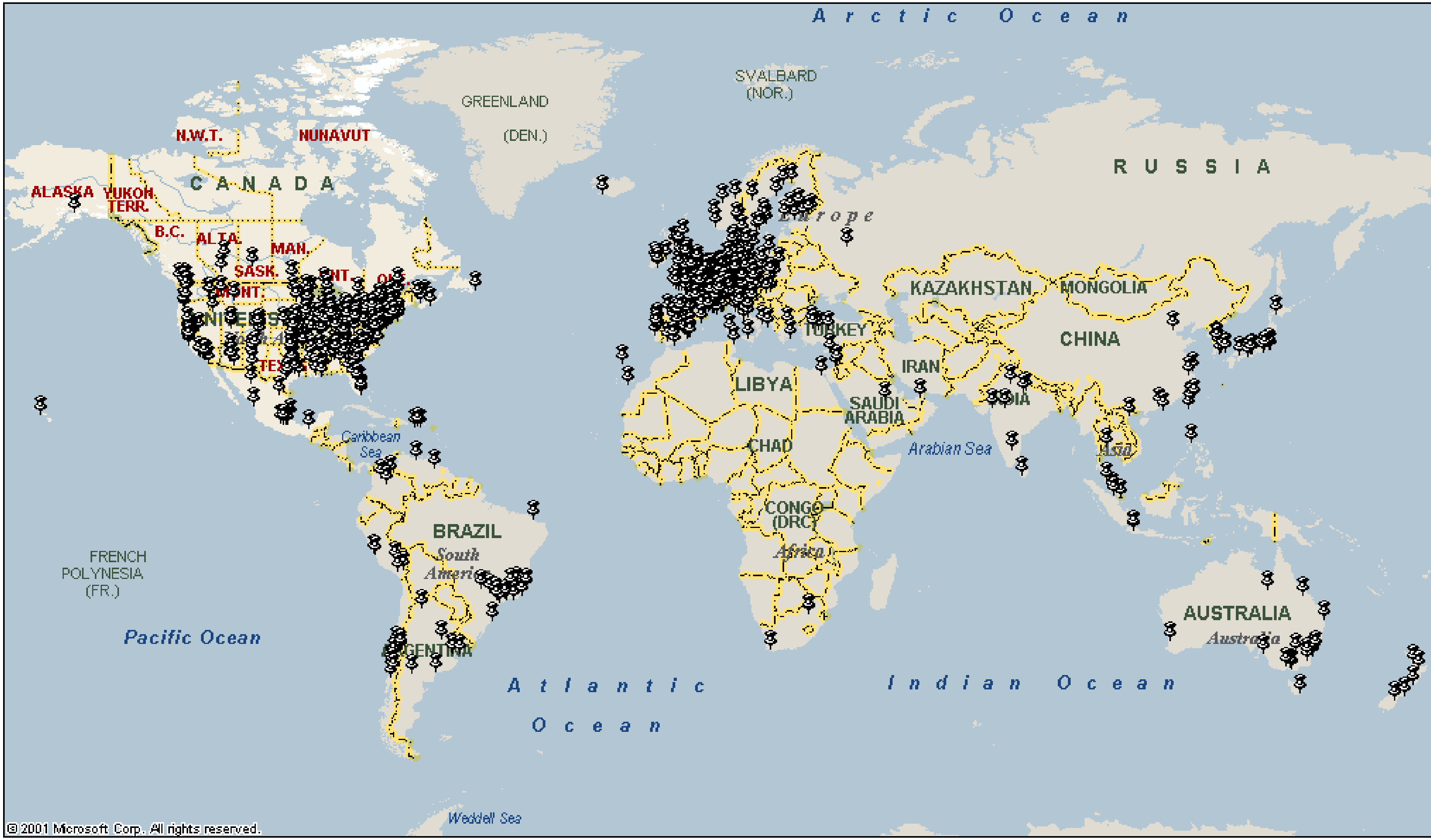
Application of LP & MIP - III

- Agriculture
  - Production planning
  - Farm land management
  - Agricultural pricing models
  - Crop and product mix decision models
  - Product distribution
- Public utilities and natural resources
  - Electric power distribution
  - Power generator scheduling
  - Power tariff rate determination
  - Natural gas distribution planning
  - Natural gas pipeline transportation
  - Water resource management
  - Alternative water supply evaluation
  - Water reservoir management
  - Public water transportation models
  - Mining excavation models
- Oil and gas exploration and production
  - Oil and gas production scheduling
  - Natural gas transportation scheduling
- Communications and computing
  - Circuit board (VLSI) layout
  - Logical circuit design
  - Magnetic field design
  - Complex computer graphics
  - Curve fitting
  - Virtual reality systems
  - Computer system capacity planning
  - Office automation
  - Multiprocessor scheduling
  - Telecommunications scheduling
  - Telephone operator scheduling
  - Telemarketing site selection

Application of LP & MIP - IV

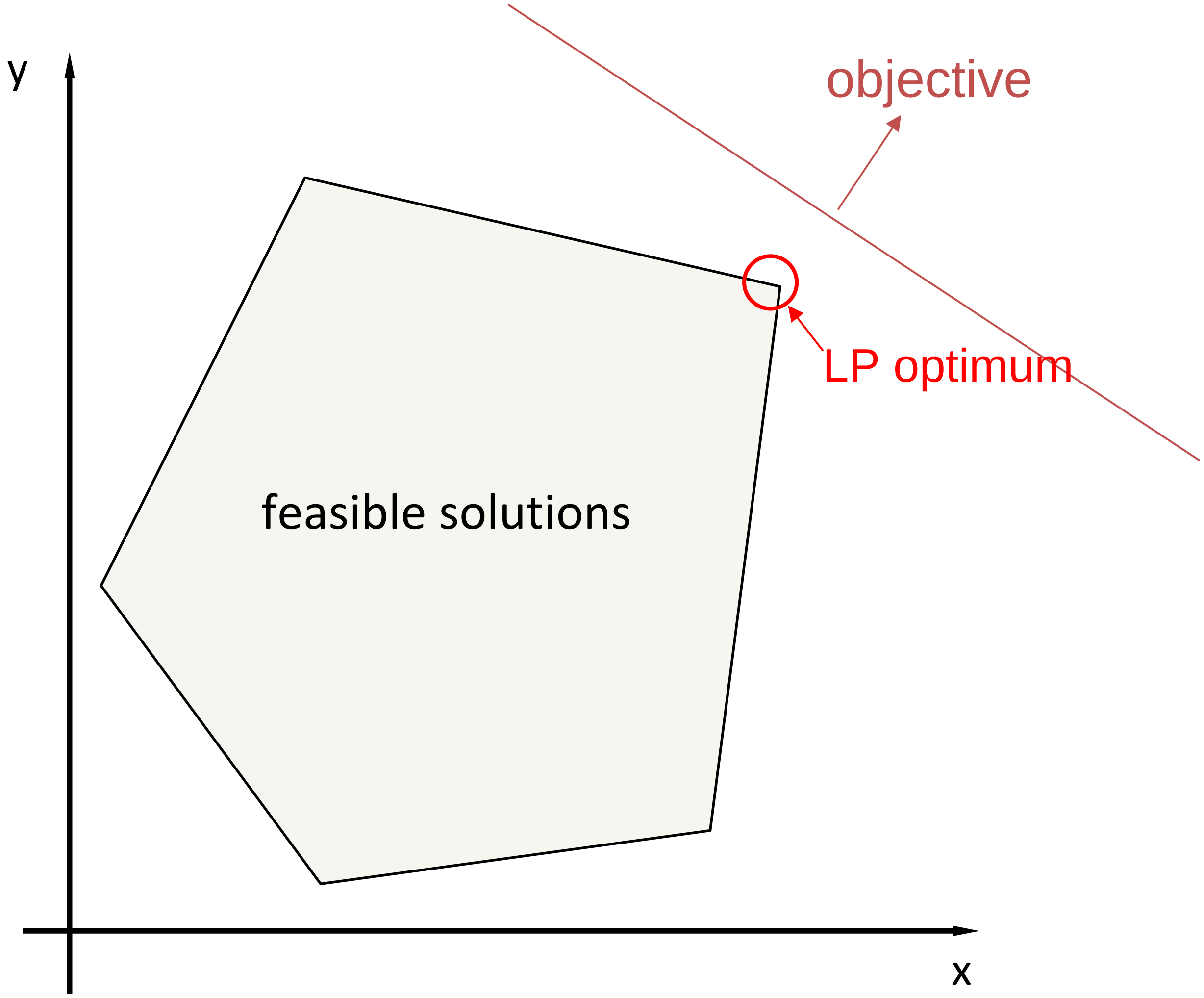
- Food processing
  - Food blending
  - Recipe optimization
  - Food transportation logistics
  - Food manufacturing logistics and scheduling
- Health care
  - Hospital staff scheduling
  - Hospital layout
  - Health cost reimbursement
  - Ambulance scheduling
  - Radiation exposure models
- Pulp and paper industry
  - Inventory planning
  - Trim loss minimization
  - Waste water recycling
  - Transportation planning
- Textile industry
  - Pattern layout and cutting optimization
  - Production scheduling
- Government and military
  - Post office scheduling and planning
  - Military logistics
  - Target assignment
  - Missile detection
  - Manpower deployment
- Miscellaneous applications
  - Advertising mix/media scheduling
  - Pollution control models
  - Sales region definition
  - Sales force deployment

CPLEX Across the World



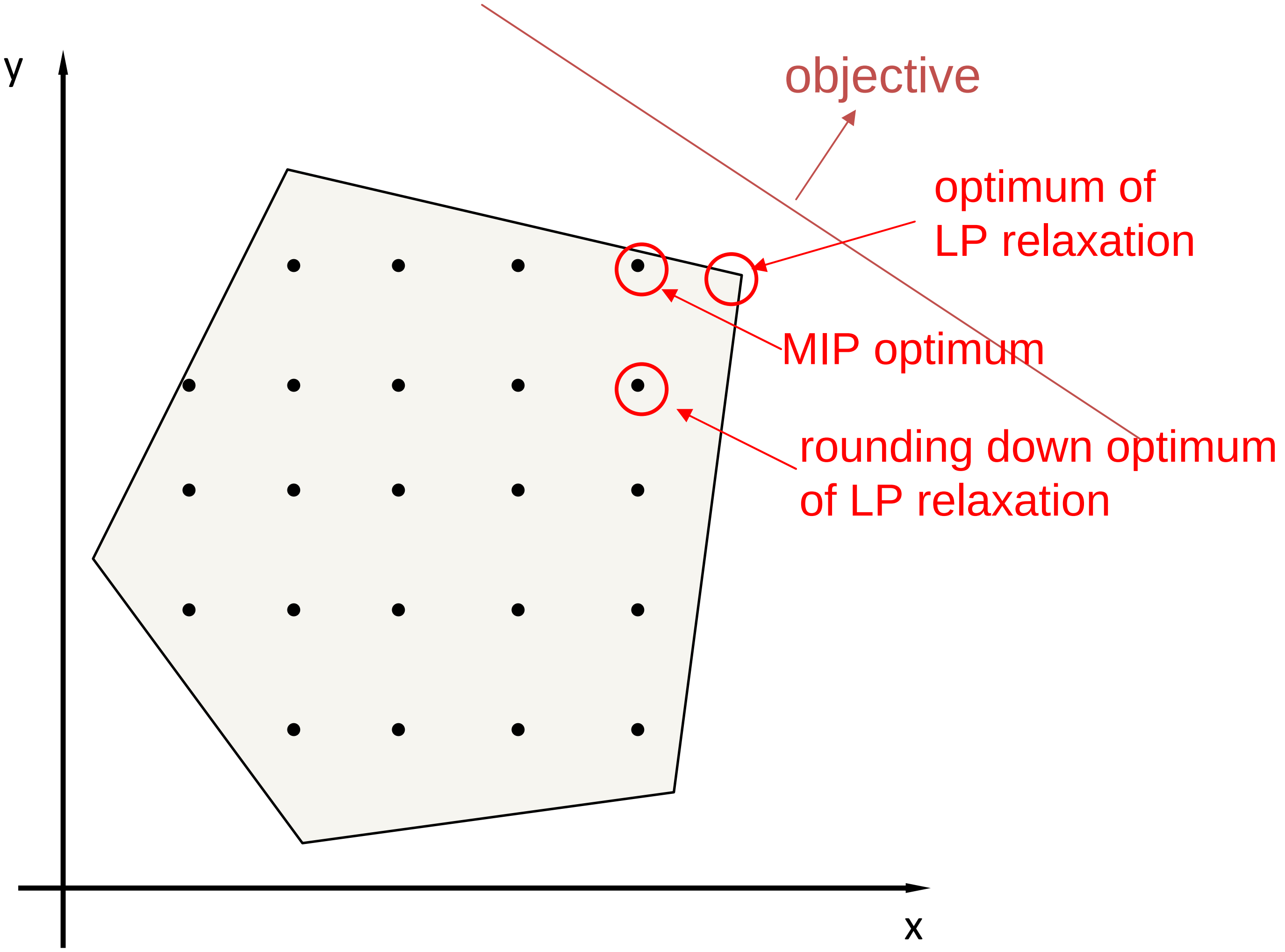
1194 Cities – Excluding ISV Deployments

How Can We Solve a MIP?

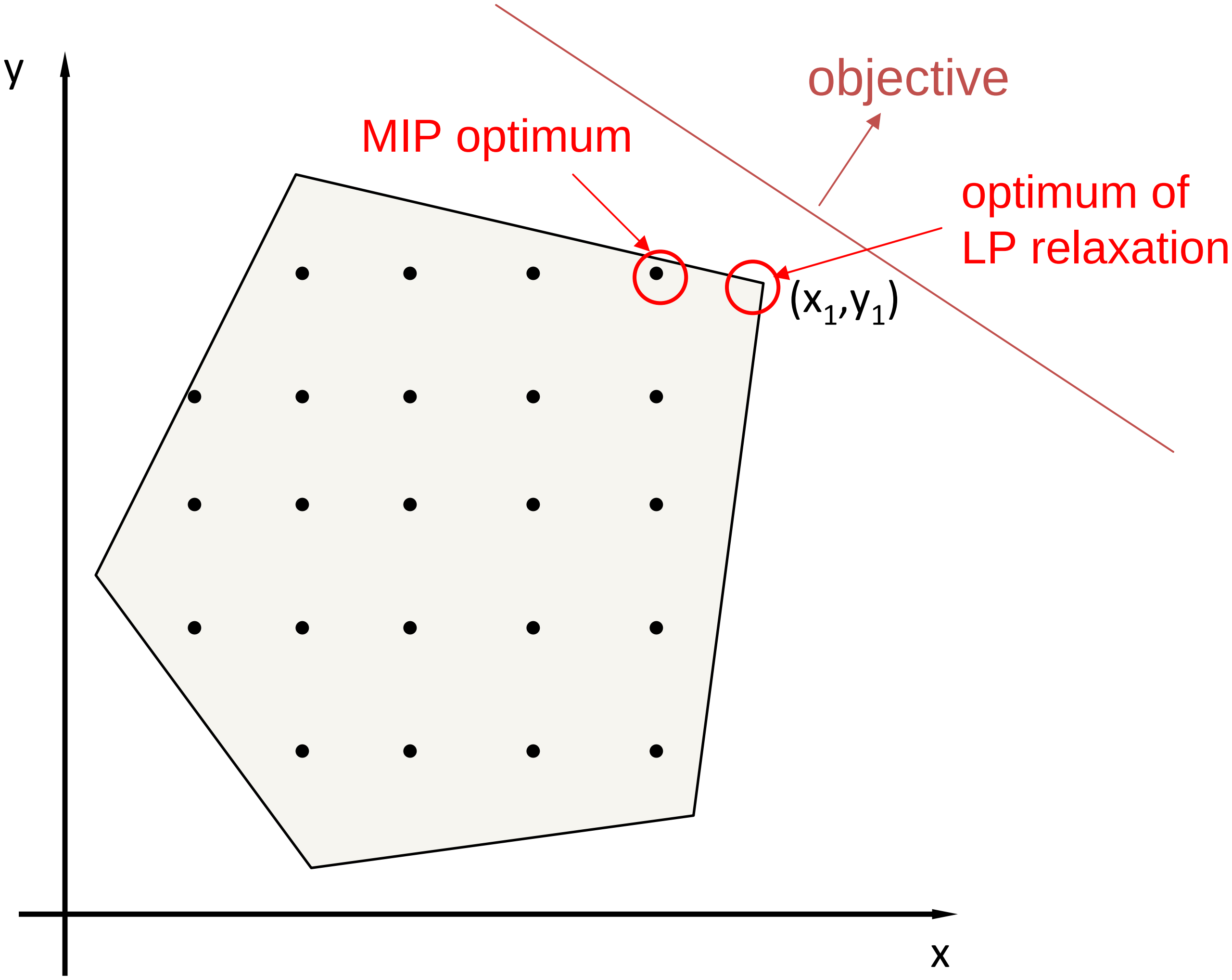


Linear Program

feasible solutions = •

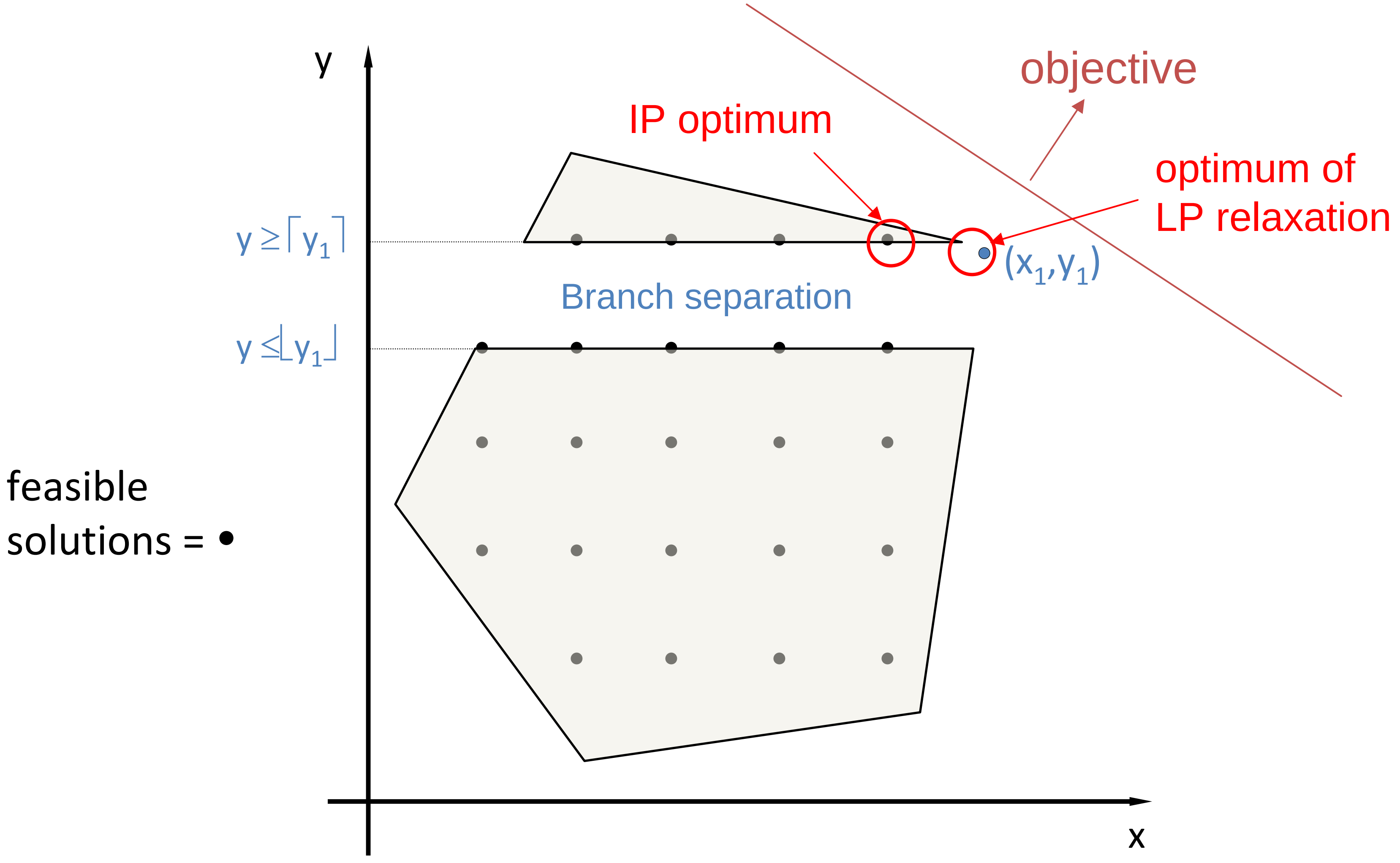


Integer Program

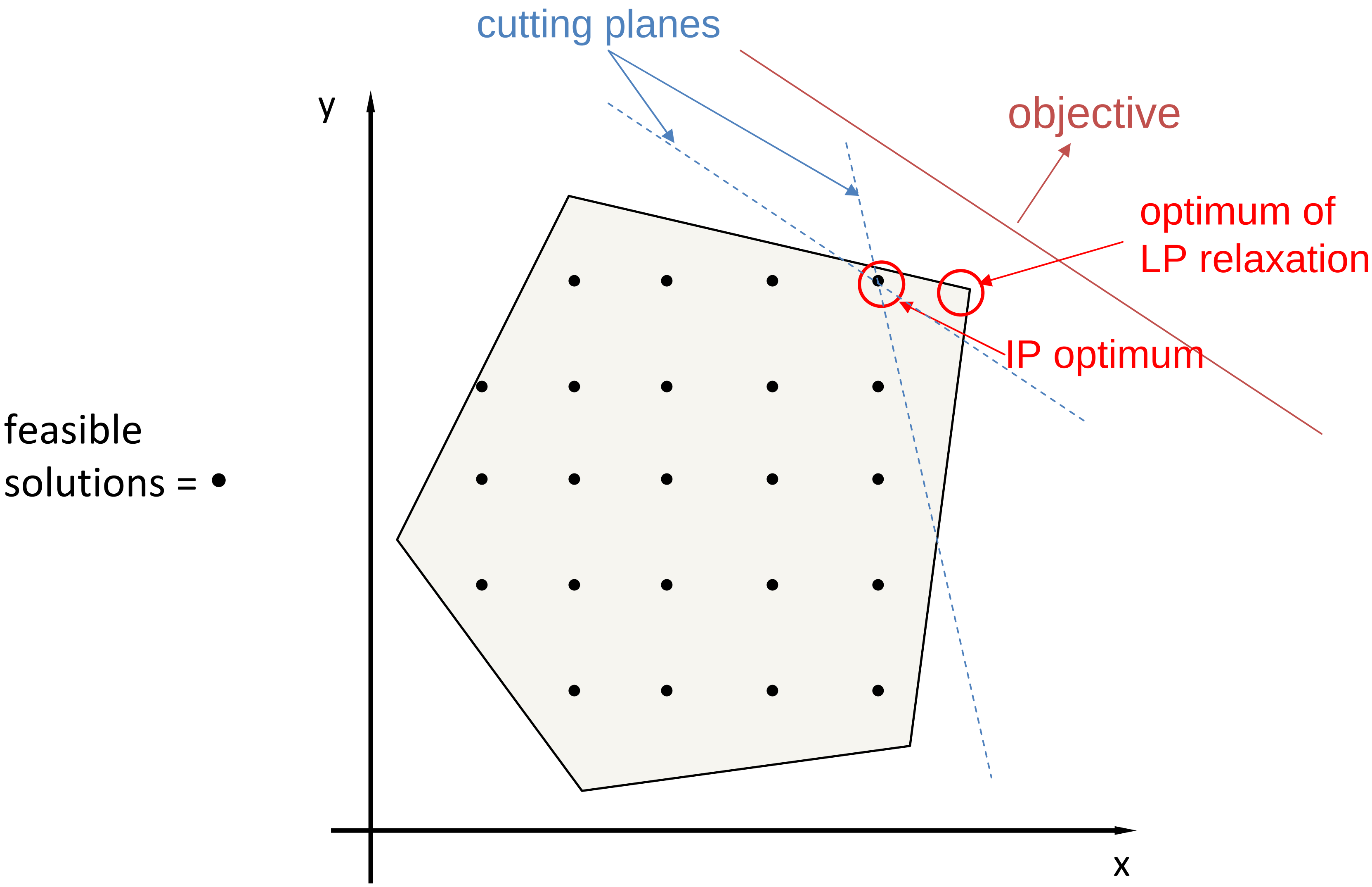


feasible solutions = •

Integer Program



Integer Program: Branch and Bound



Integer Program: Cutting Planes

Solving MIPs: Branch and Bound

Consider the following integer program:

Maximize $x + y + 2z$

Subject to $7x + 2y + 3z \leq 36$

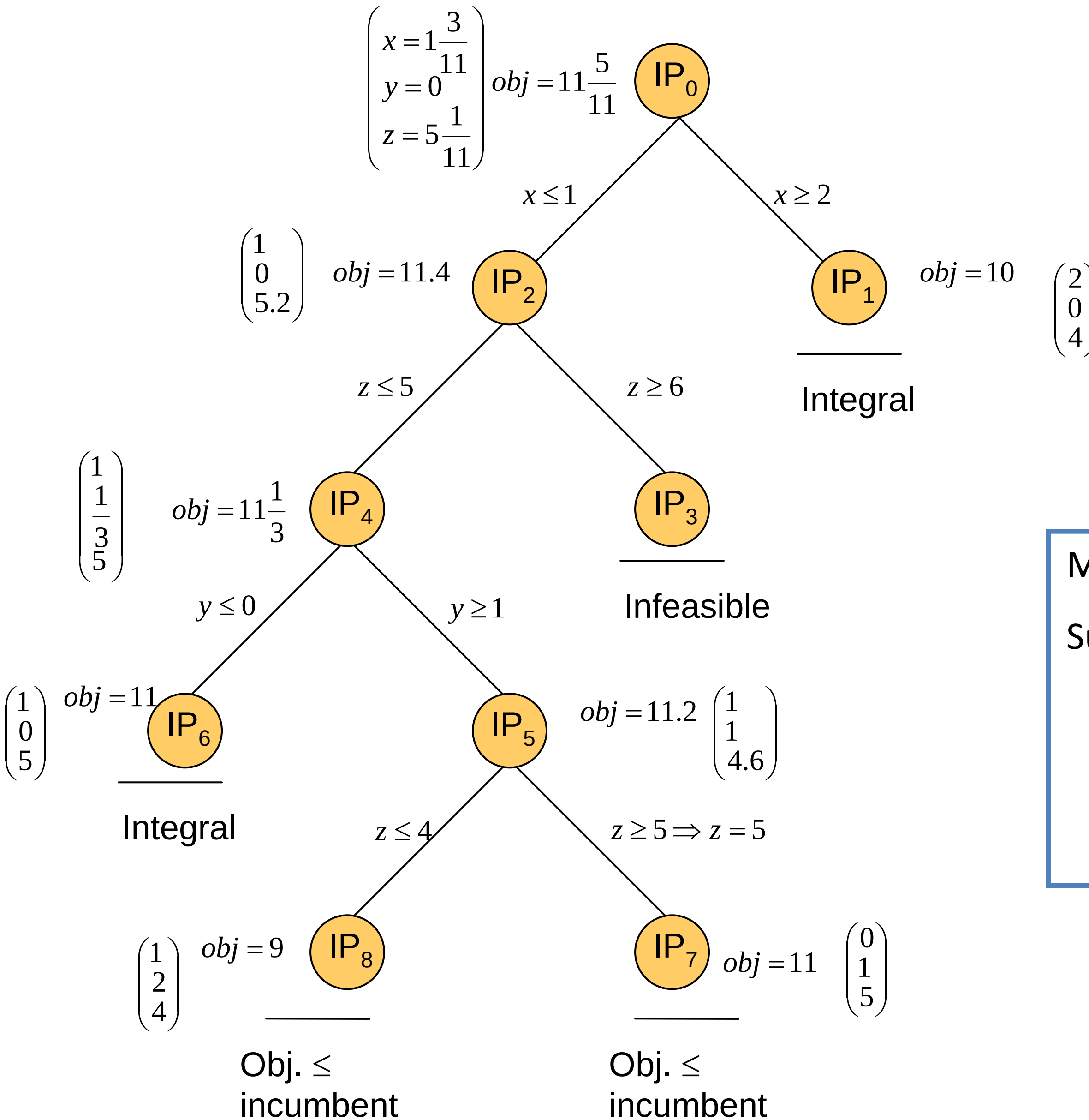
$5x + 4y + 7z \leq 42$

$2x + 3y + 5z \leq 28$

$x, y, z \geq 0$, integer

(IP_0)

Branch & Bound: Example



Best IP value: ~~$-\infty$~~ ~~10~~ 11

Best IP solution: ~~$\begin{pmatrix}2\\0\\4\end{pmatrix}$~~ $\begin{pmatrix}1\\0\\5\end{pmatrix}$

$$\begin{aligned}
\text{Maximize} \quad & x + y + 2z \\
\text{Subject to} \quad & 7x + 2y + 3z \le 36 \\
& 5x + 4y + 7z \le 42 \\
& 2x + 3y + 5z \le 28 \\
& x, y, z \ge 0, \text{ integer}
\end{aligned}$$

Unit Commitment:
Using Branch-and-Cut

```
Read MPS format model from file unitcal_1.lp
Optimize a model with 7035 Rows, 3723 Columns and 18155 NonZeroes
Presolved: 5111 Rows, 2717 Columns, 14193 Nonzeros
Root relaxation: 2287 iterations, 0.11 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 2653304.4    0   83          - 2653304.4      -      -    0s
H    0     0                    2690605.2 2653304.4  1.39%     -    0s
H    0     0                    2687988.7 2676216.2  0.44%     -    0s
     0     0 2676687.8    0   71 2687988.7 2676687.8  0.42%     -    0s
H    0     0                    2686373.6 2676982.3  0.35%     -    0s
     0     0 2678857.7    0   54 2686373.6 2678857.7  0.28%     -    1s
H    0     0                    2685687.5 2678881.2  0.25%     -    1s
     0     0 2678901.7    0   65 2685687.5 2678901.7  0.25%     -    1s
H   60    35                    2684072.4 2679836.3  0.16%  30.8    1s
*   93    35              14    2683479.8 2680016.5  0.13%  23.6    1s

Cutting planes:
  Gomory: 5
  Implied bound: 167
  MIR: 1
  Flow cover: 71

Explored 568 nodes (14223 simplex iterations) in 2.07 seconds

Optimal solution found (tolerance 1.00e-04)
Best objective 2.6834797511e+06, best bound 2.6834797511e+06, gap 0.0000%
```

Applied ‘tricks’ or heuristics

Total nodes processed

Current UB

Current LB

Constraints auto-added to strengthen formulation

The Basic Algorithm:
Branch and Bound

Solution Strategy: Branch & Bound

- Split the solution space into disjoint subsets
- Bound the objective value for all solutions in a subset

Branching

- Choose a *branching variable* x_j
  - Must be an integer variable
- Split the model into two sub-models
  - $x_j \leq i$ or $x_j \geq i+1$
- Binary variable special case:
  - $x_j=0$ or $x_j=1$

Bounding - Continuous Relaxation

Minimize $c^T x$ $(= z_{lp})$

Subject to $Ax = b$

$l \leq x \leq u$

Some x are integer

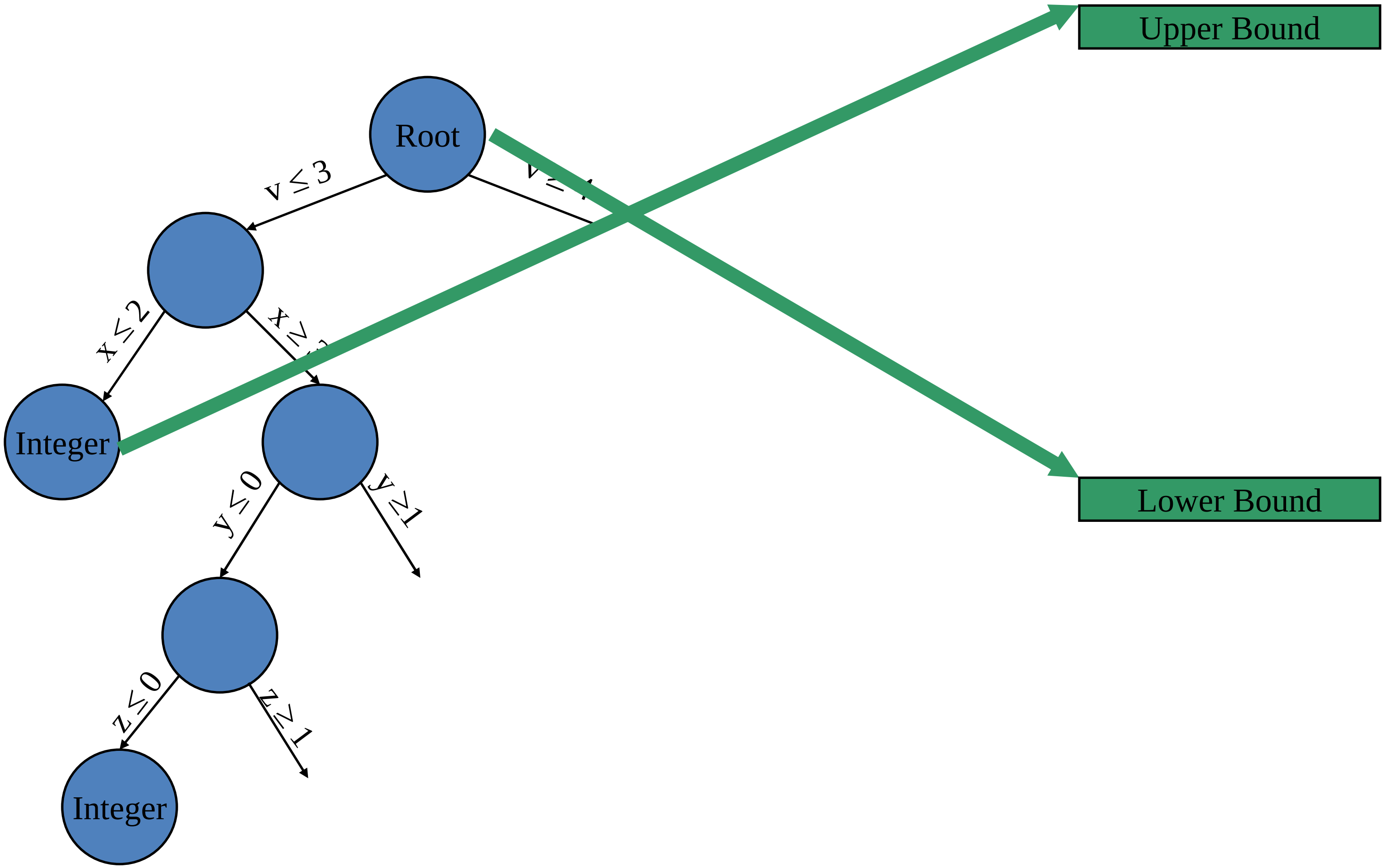
Lower bound on MIP objective

Relax Integrality Restriction

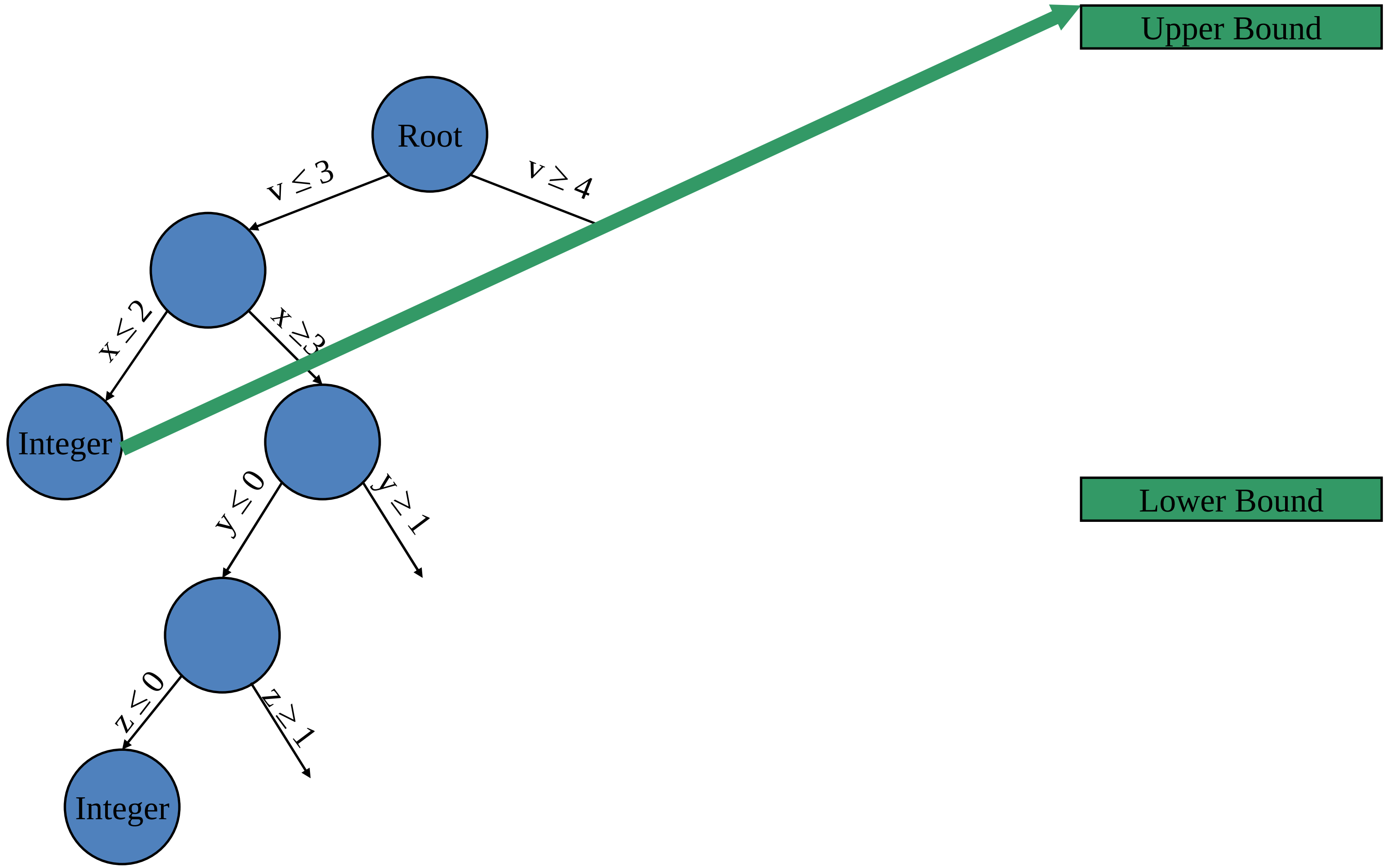
Nice Properties of Continuous Rel.

- If relaxation solution satisfies integrality restrictions:
  - No need to further explore subspace
- Natural branching candidates:
  - Integer variables that are fractional in relaxation

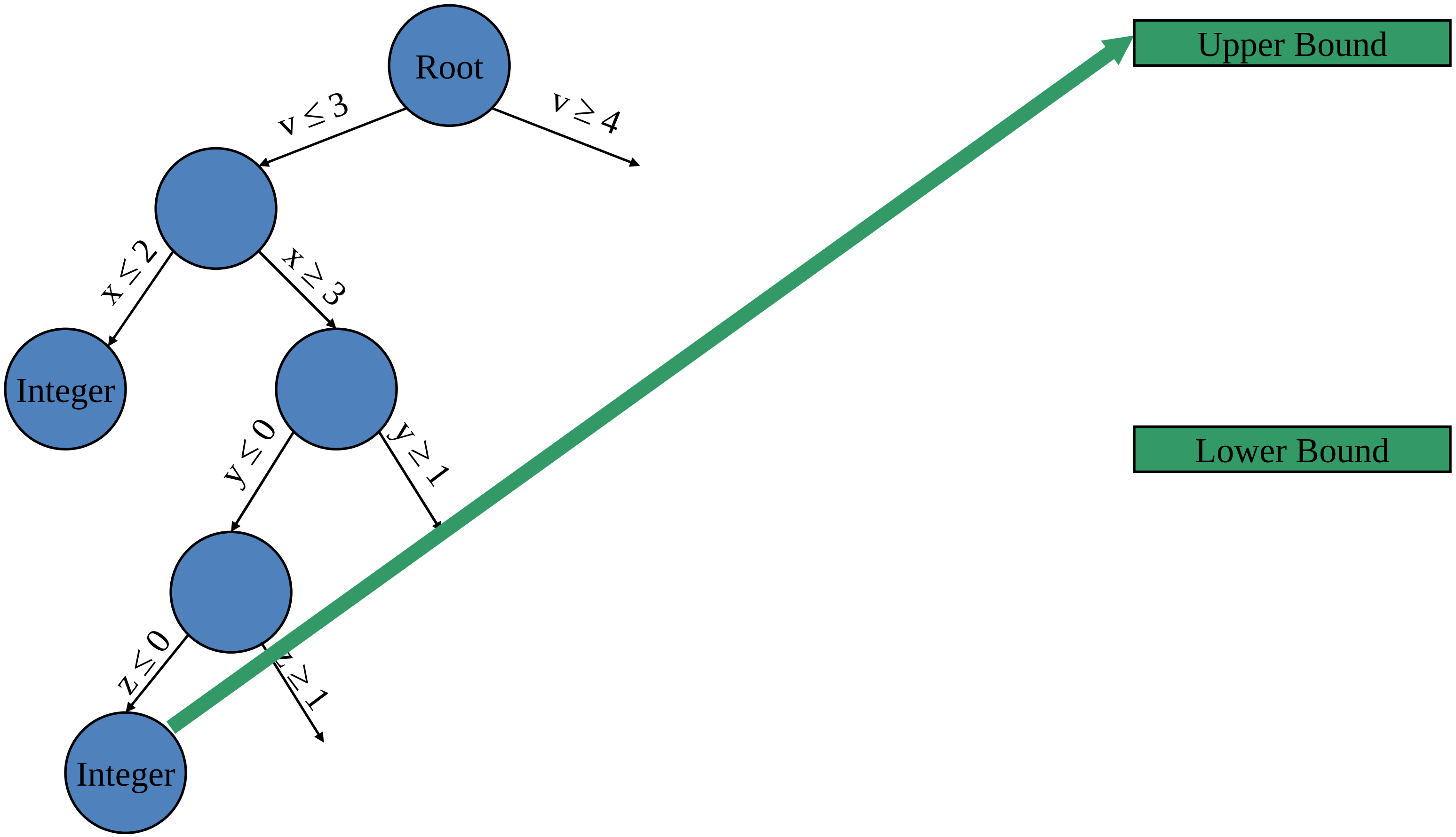
Branch and Bound for MIP



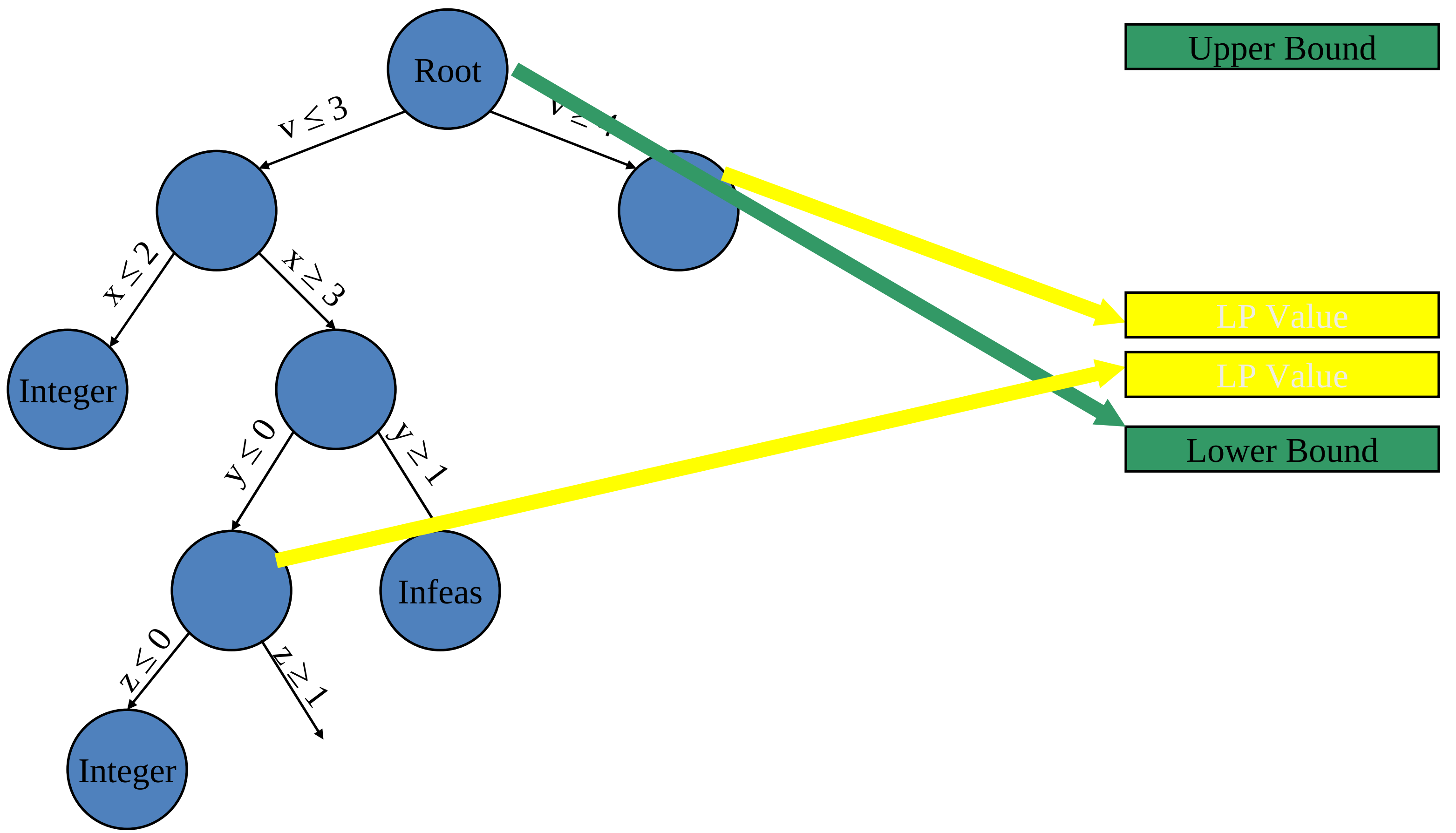
Branch and Bound for MIP



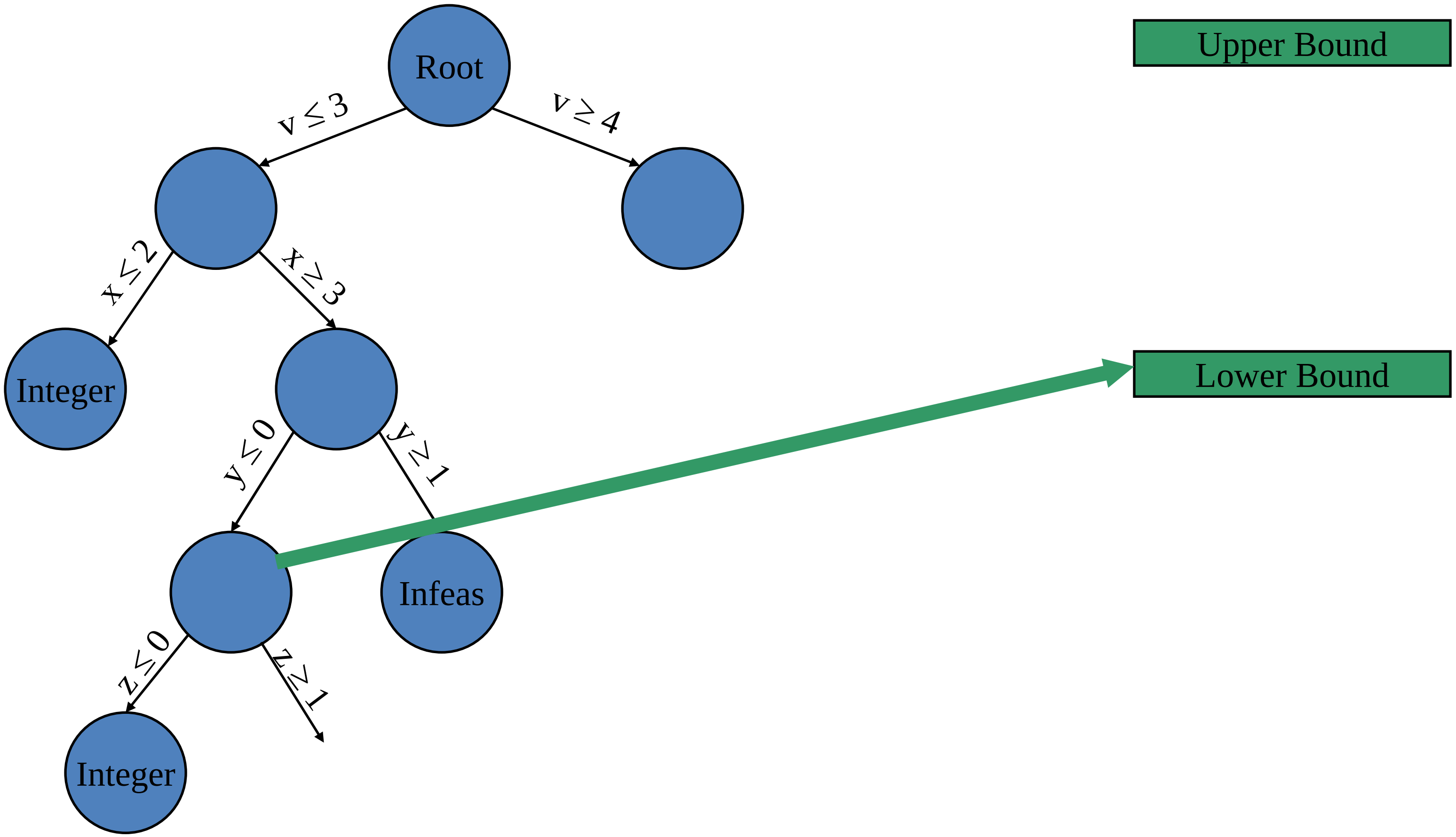
Branch and Bound for MIP



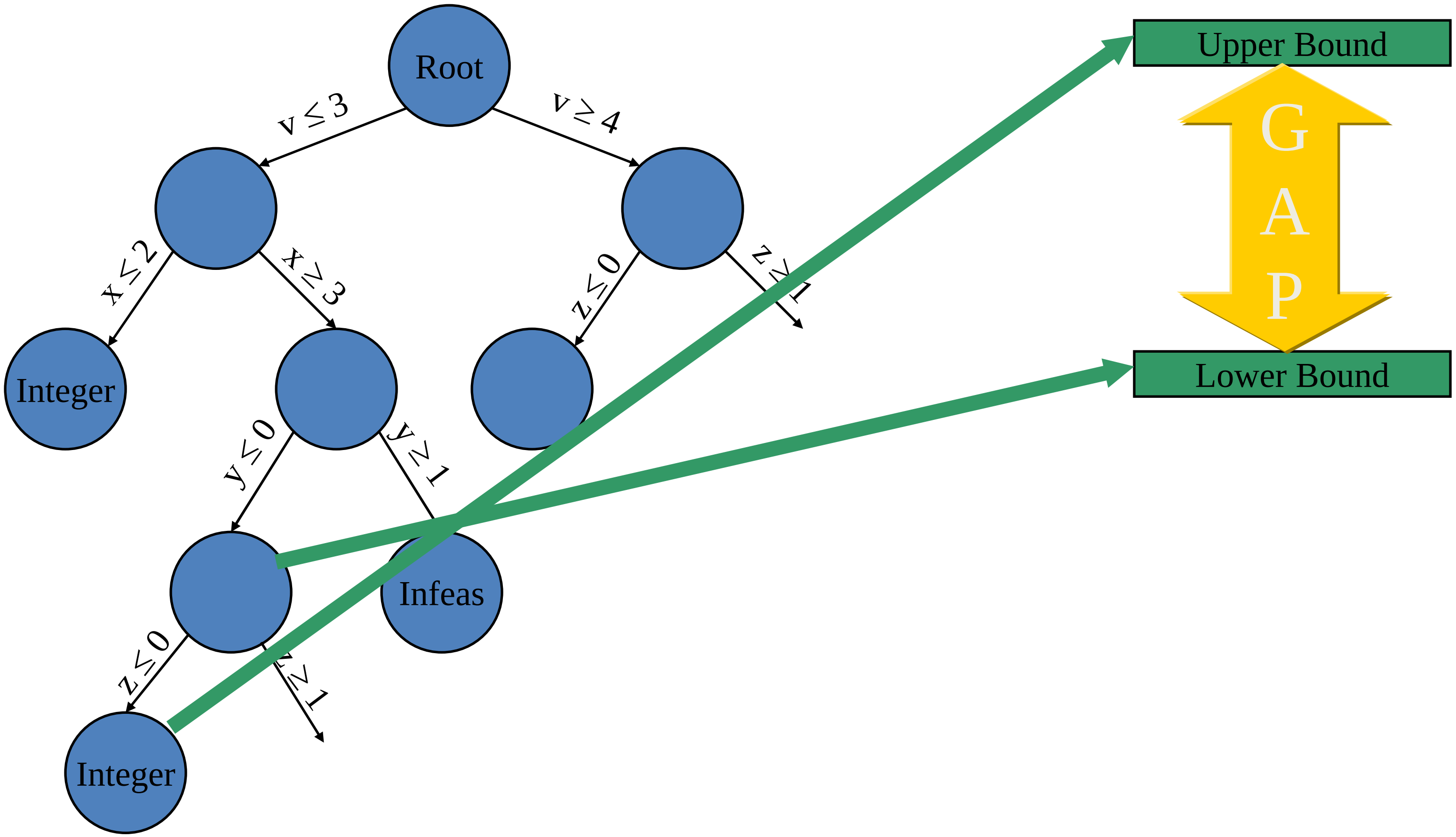
Branch and Bound for MIP



Branch and Bound for MIP



Branch and Bound for MIP



Important Steps

Important Steps

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

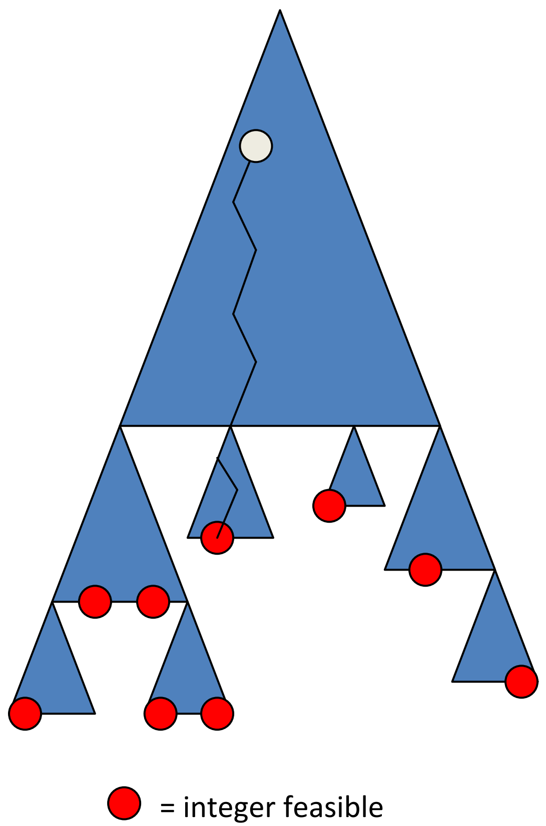
Important Steps

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

- 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

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

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, Chvatal, Cook]
  - Use historical information
    - Pseudo-costs

Important Steps
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

- 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

- Brief History of CPLEX MIP
- Heuristic details
- Cutting plane details