SOLVING INTEGER LINEAR PROGRAMS

1. Solving the LP relaxation.
2. How to deal with fractional solutions?

Integer Linear Program: Example

$$
\begin{array}{lll}
\max & -x_1 - 2x_2 - 0.5x_3 - 0.2x_4 - x_5 + 0.6x_6 & \\
\text{s.t.} & x_1 + 2x_2 & \geq \quad 1 \\
& x_1 + x_2 + 3x_6 & \geq \quad 1 \\
& x_1 + x_2 + x_6 & \geq \quad 1 \\
& x_3 - 3x_4 & \geq \quad 1 \\
& x_3 - 2x_4 - 5x_5 & \geq \quad 1 \\
& x_4 + 3x_5 - 4x_6 & \geq \quad 1 \\
& x_2 + x_5 + x_6 & \geq \quad 1 \\
0 \leq & x_1, x_2, \cdots, x_6 & \leq \quad 10 \\
& x_1, \ldots, x_6 & \in \quad \mathbb{Z}
\end{array}
$$

Solving ILPs

```
var x1 integer >= 0, <= 10;
var x2 integer >= 0, <= 10;
var x3 integer >= 0, <= 10;
var x4 integer >= 0, <= 10;
var x5 integer >= 0, <= 10;
var x6 integer >= 0, <= 10;

maximize obj: -x1- 2* x2 -0.5*x3- 0.2* x4 - x5 +0.5* x6;
c1: x1 + 2 * x2 >= 1;
c2: x1 + x2 + 3* x6 >= 1;
c3: x1 + x2 + x6 >= 1;
c4: x3 - 3* x4  >= 1;
c5: x3 - 2* x4 -5* x5 >= 1;
c6: x4 + 3* x5 -4 *x6  >= 1;
c7: x2 + x5 + x6 >= 1;
solve;
display x1, x2,x3, x4, x5, x6;
end;
```

Solution

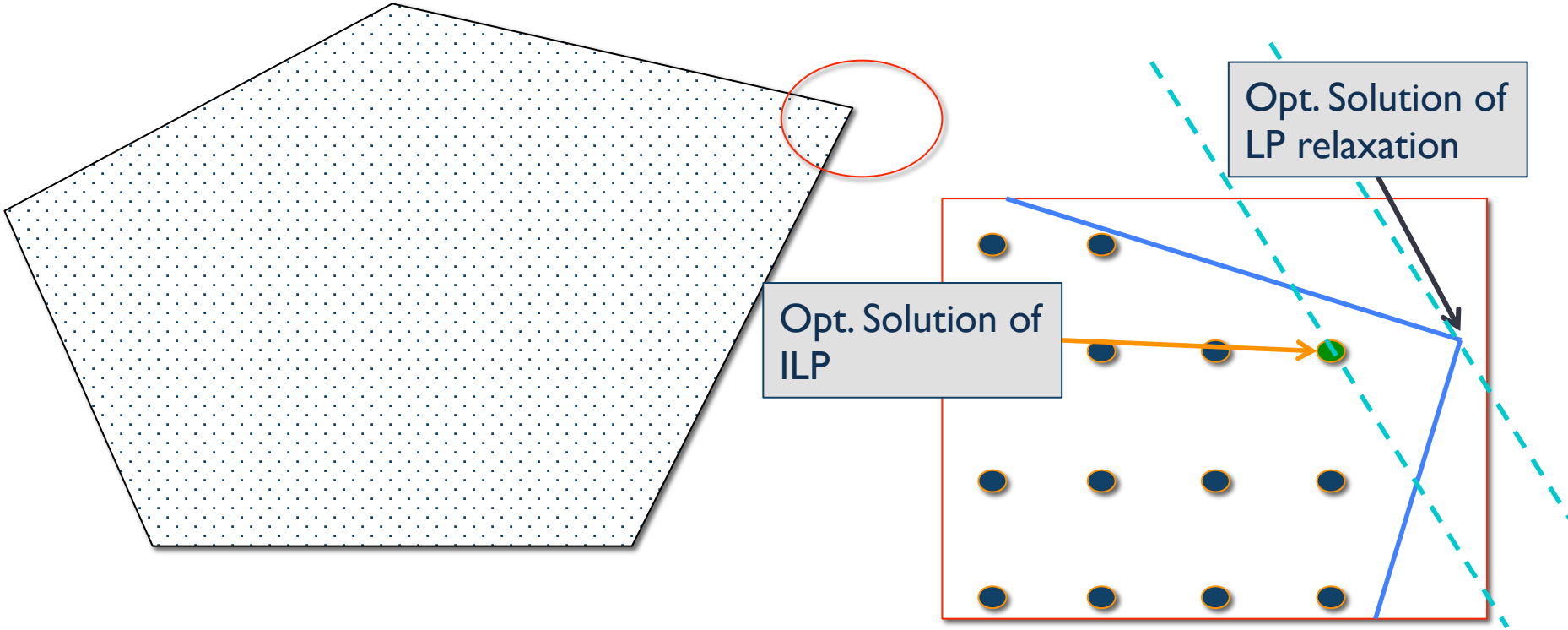
Original ILP

```
x1.val = 0
x2.val = 1
x3.val = 4
x4.val = 1
x5.val = 0
x6.val = 0
Optimal Value: -4.2
```

LP Relaxation

```
x1.val = 0.333333333333333
x2.val = 0.666666666666667
x3.val = 2.66666666666667
x4.val = 0
x5.val = 0.333333333333333
x6.val = 0
Optimal Value: -3.333333
```

Case-1: Both LP and ILP are feasible.



Solving ILPs

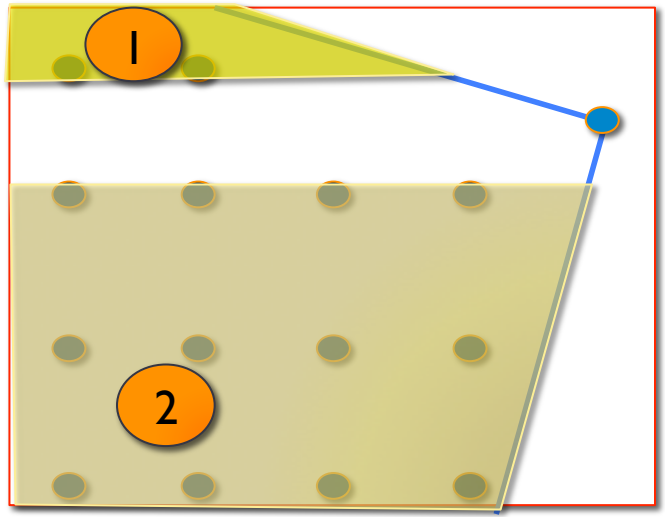
Solve LP relaxation of the ILP.

Case-1: LP relaxation solution satisfies integrality constraint.

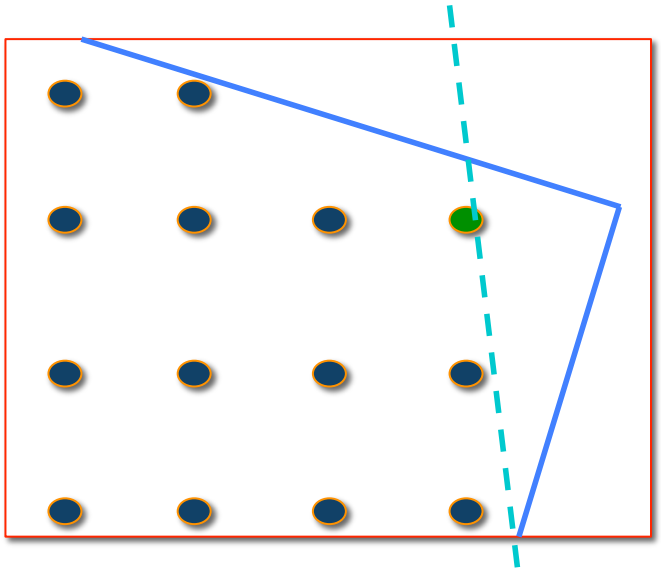
Case-2: LP relaxation solution does not satisfy the integrality constraint.

Dealing with Case-2

Branch and Bound



Cutting Plane Method.



BRANCH-AND-BOUND: BASIC IDEA

Integer Linear Program: Example

$$
\begin{array}{lll}
\max & -x_1 - 2x_2 - 0.5x_3 - 0.2x_4 - x_5 + 0.6x_6 & \\
\text{s.t.} & x_1 + 2x_2 & \geq \quad 1 \\
& x_1 + x_2 + 3x_6 & \geq \quad 1 \\
& x_1 + x_2 + x_6 & \geq \quad 1 \\
& x_3 - 3x_4 & \geq \quad 1 \\
& x_3 - 2x_4 - 5x_5 & \geq \quad 1 \\
& x_4 + 3x_5 - 4x_6 & \geq \quad 1 \\
& x_2 + x_5 + x_6 & \geq \quad 1 \\
0 \leq & x_1, x_2, \cdots, x_6 & \leq \quad 10 \\
& x_1, \ldots, x_6 & \in \quad \mathbb{Z}
\end{array}
$$

Step #1: Solve the LP relaxation

Solving the LP relaxation.

```
x1.val = 0.333333333333333
x2.val = 0.666666666666667
x3.val = 2.66666666666667
x4.val = 0
x5.val = 0.333333333333333
x6.val = 0

Optimal Value (LP relaxation): -3.333333
```

Step #2: Choose a branch variable

```
x1.val = 0.333333333333333
x2.val = 0.666666666666667
x3.val = 2.66666666666667
x4.val = 0
x5.val = 0.333333333333333
x6.val = 0

Optimal Value (LP relaxation): -3.333333
```

Choose a variable that should be integer

Step #3: Branching Constraints

$$
\begin{array}{lll}
\max & -x_1 - 2x_2 - 0.5x_3 - 0.2x_4 - x_5 + 0.6x_6 & \\
\text{s.t.} & x_1 + 2x_2 & \geq 1 \\
& x_1 + x_2 + 3x_6 & \geq 1 \\
& x_1 + x_2 + x_6 & \geq 1 \\
& x_3 - 3x_4 & \geq 1 \\
& x_3 - 2x_4 - 5x_5 & \geq 1 \\
& x_4 + 3x_5 - 4x_6 & \geq 1 \\
& x_2 + x_5 + x_6 & \geq 1 \\
0 \leq & x_1, x_2, \cdots, x_6 & \leq 10
\end{array}
$$

Original Problem

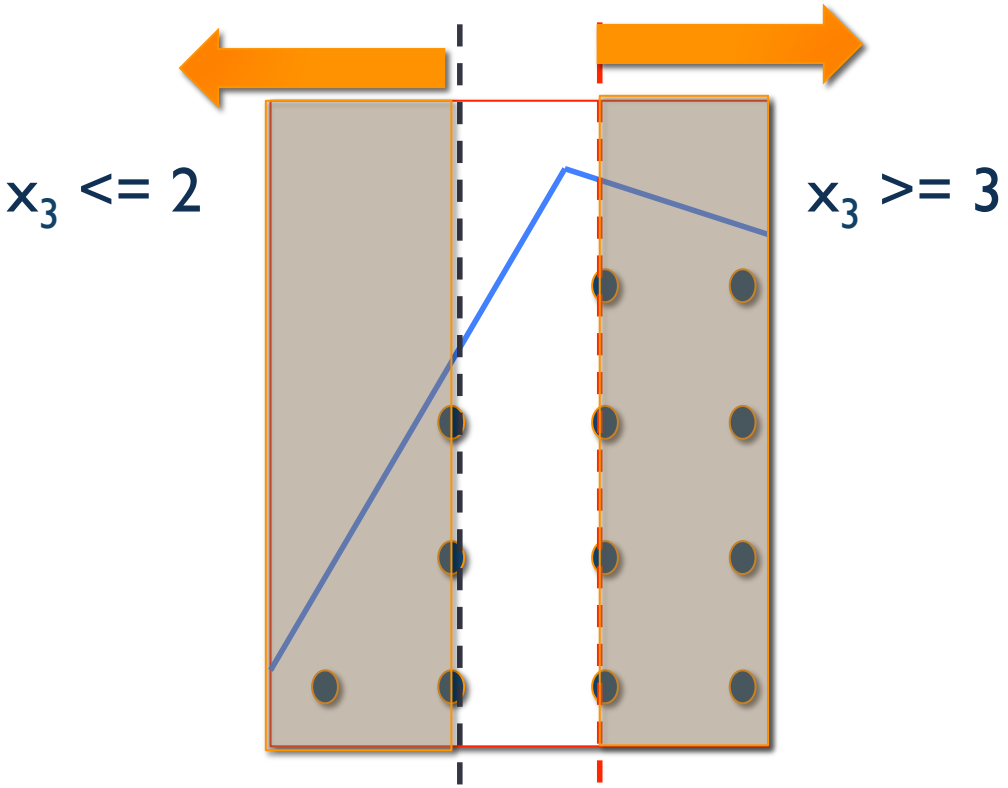
b1

b2

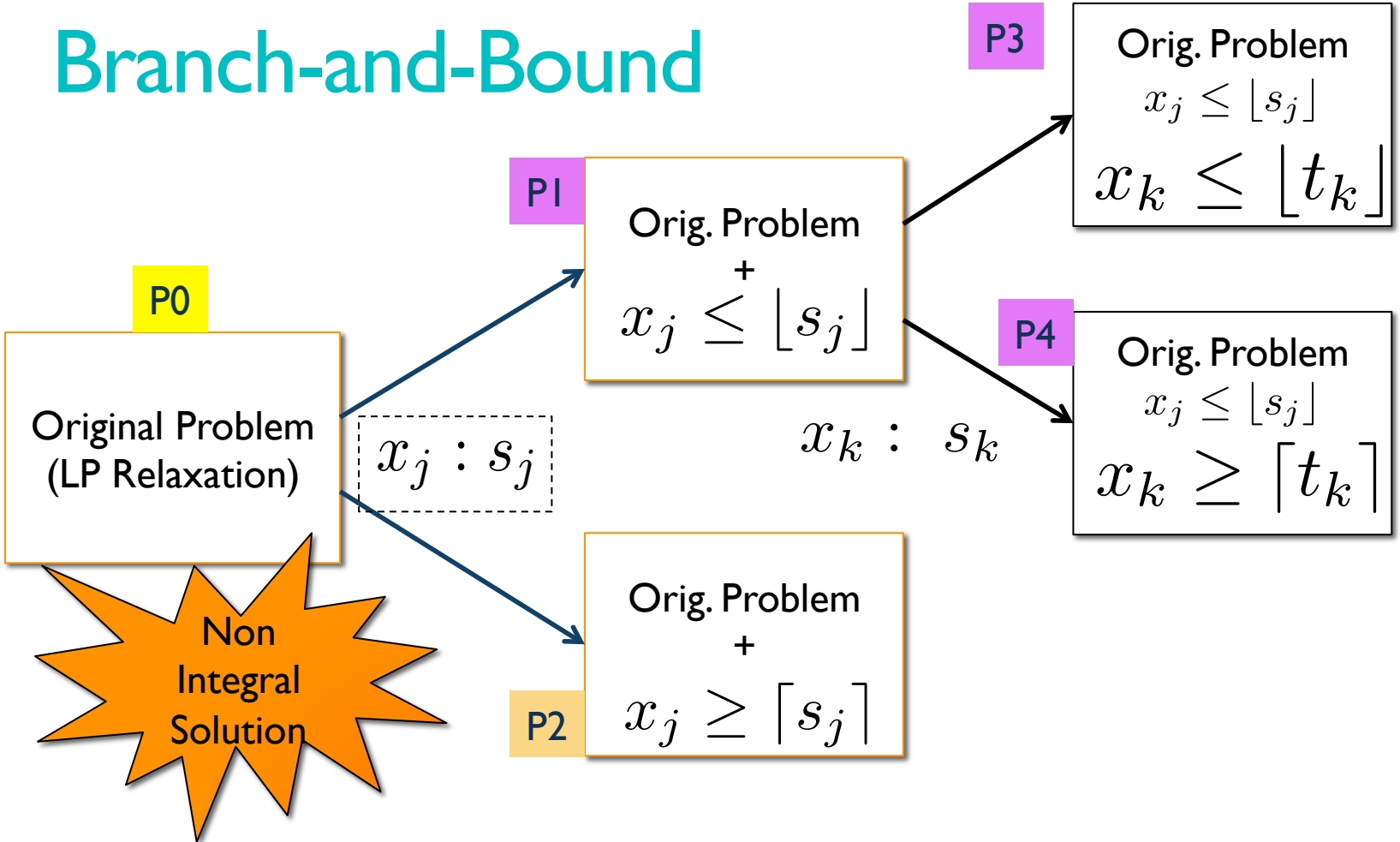
Original Problem Constr.
+
(x_3 <= 2)

Original Problem Constr.
+
(x_3 >= 3)

Visualization

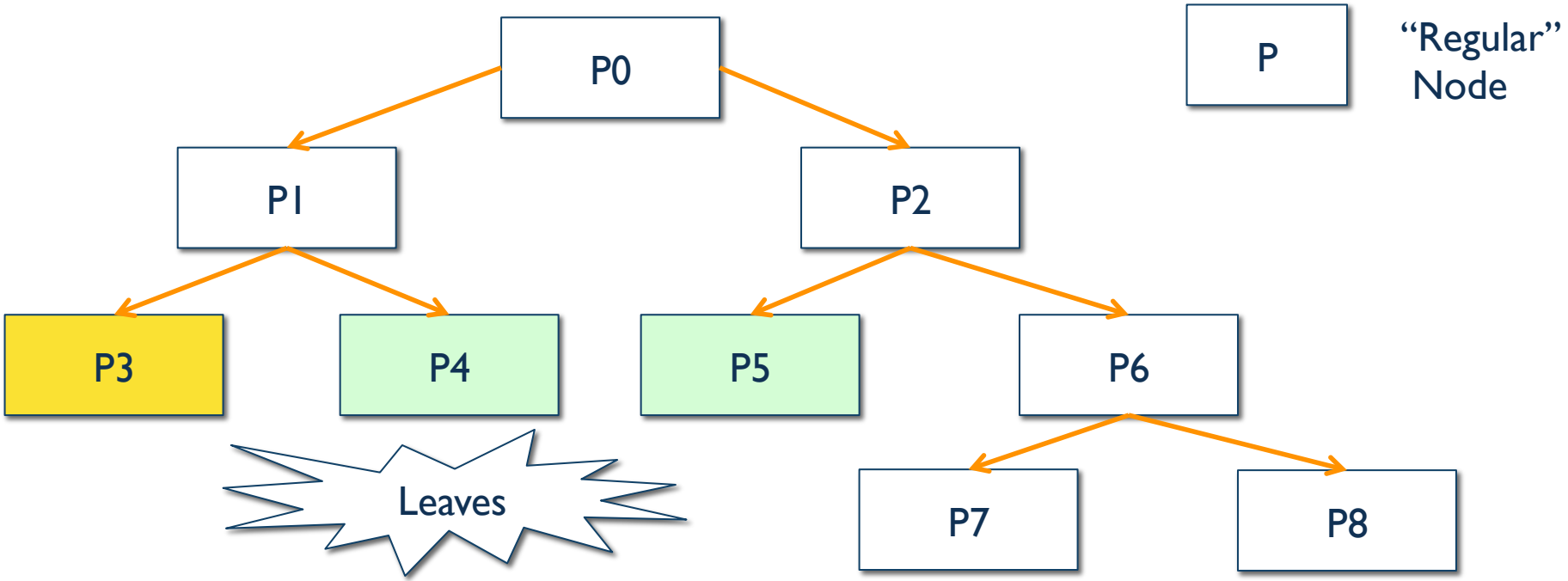


Branch-and-Bound



Branch-And-Bound: Operations

Branch-And-Bound Tree: Nodes are ILP instances.



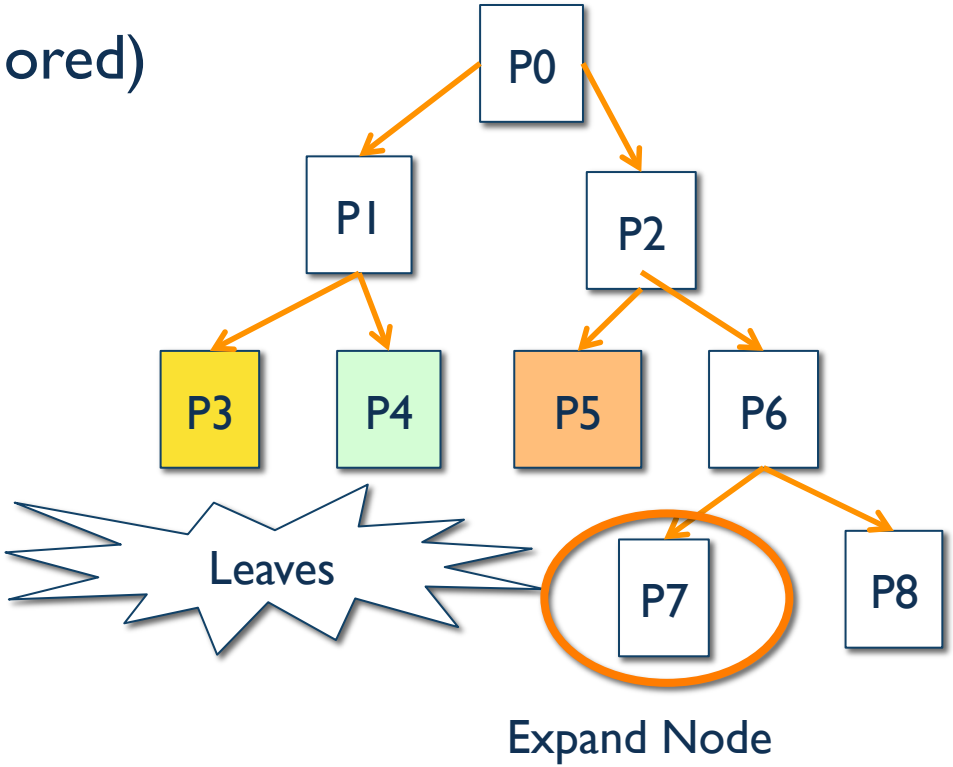
Branch-And-Bound Tree

P0 Regular Node (to be explored)

P3 Leaf node

P4 Leaf node

P5 Leaf node



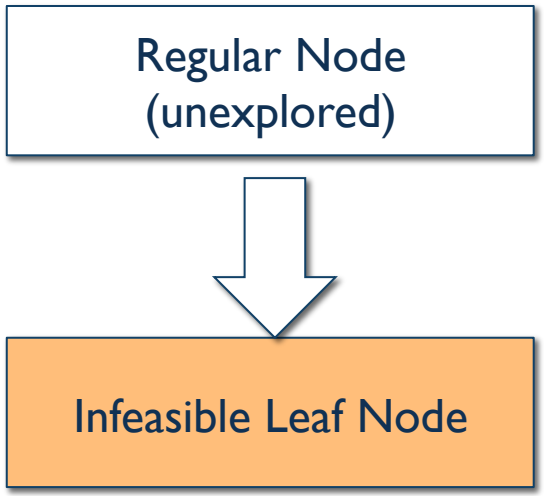
Expanding a Node

1. Solve the LP relaxation for the node ILP.

2. Three cases:
    1. Infeasible.
    2. Integral Solution.
    3. Fractional Solution.

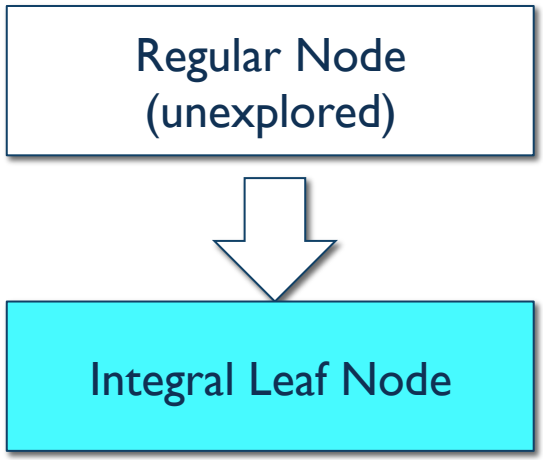
Case-1: Infeasible LP relaxation.

- LP relaxation is infeasible.



Case-2: LP relaxation yields integral solution

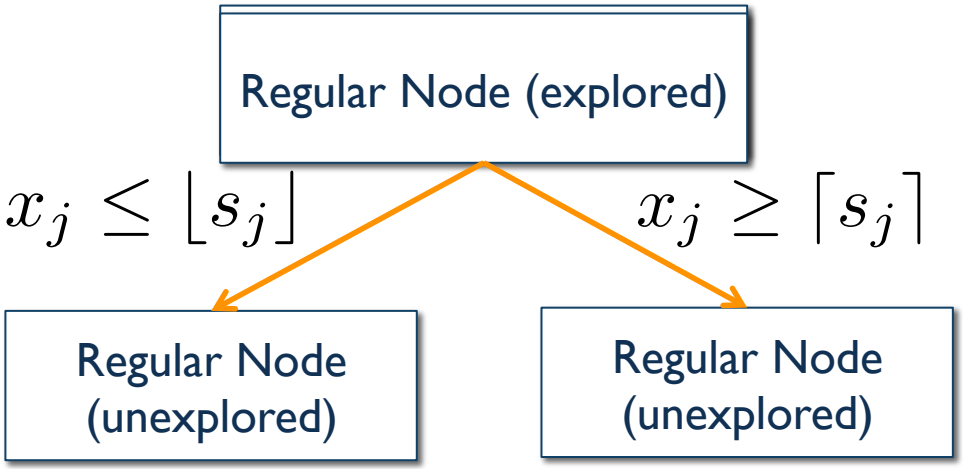
- LP relaxation solution is integral: ILP solution = LP solution.



bestObjective := max (lpOptimum, bestObjective)

Case-3: LP relaxation yields a fractional solution

- LP relaxation yields a fractional solution.



LP relaxation solution:
x1: ...
x2: ...
x3: ...
...
Opt. Solution: optSolution

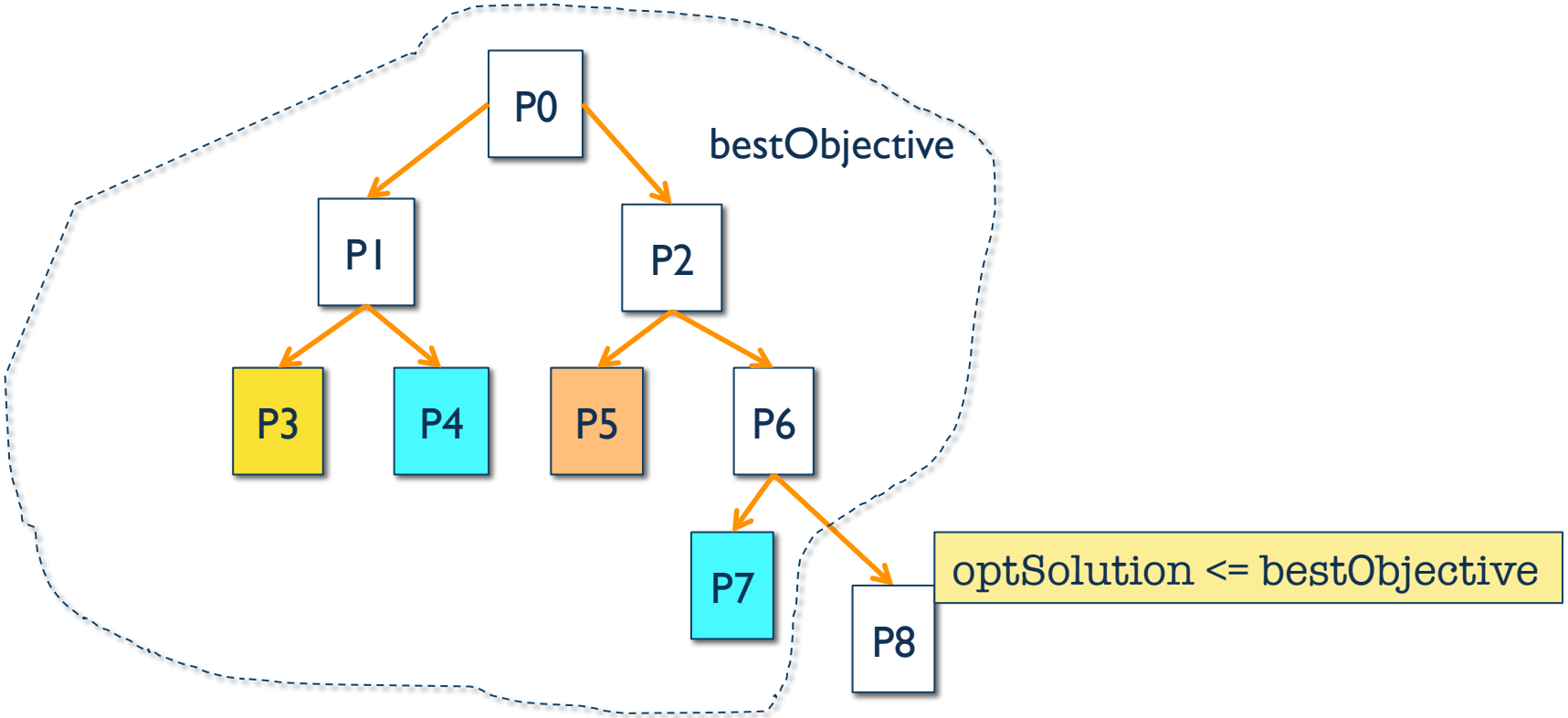
Optimal Pruning

Regular Node
(explored)

optSolution <= bestObjective?

LP relaxation solution:
x1: ...
x2: ...
x3: ...
...
Opt. Solution: optSolution

Optimal Pruning



Branch-And-Bound Initial Node

Original ILP (unexplored node)

bestObjective := -Infinity

BRANCH-AND-BOUND (EXAMPLE)

Original Problem

```
var x1 integer  >= 0, <= 10;
var x2 integer  >= 0, <= 10;
var x3  integer >= 0, <= 10;
var x4 integer  >= 0, <= 10;
var x5  integer >= 0, <= 10;
var x6  integer >= 0, <= 10;
maximize obj: -x1 - 2* x2 -0.5 * x3 - 0.2* x4 - x5 +0.5* x6;
c1: x1 + 2 * x2 >= 1;
c2: x1 + x2 + 3* x6 >= 1;
c3: x1 + x2 + x6 >= 1;
c4: x3 - 3* x4  >= 1;
c5: x3 - 2* x4 -5* x5 >= 1;
c6: x4 + 3* x5 -4 *x6   >= 1;
c7: x2 + x5 + x6 >= 1;
solve;
display x1, x2,x3, x4, x5, x6;
end;
```

Original ILP

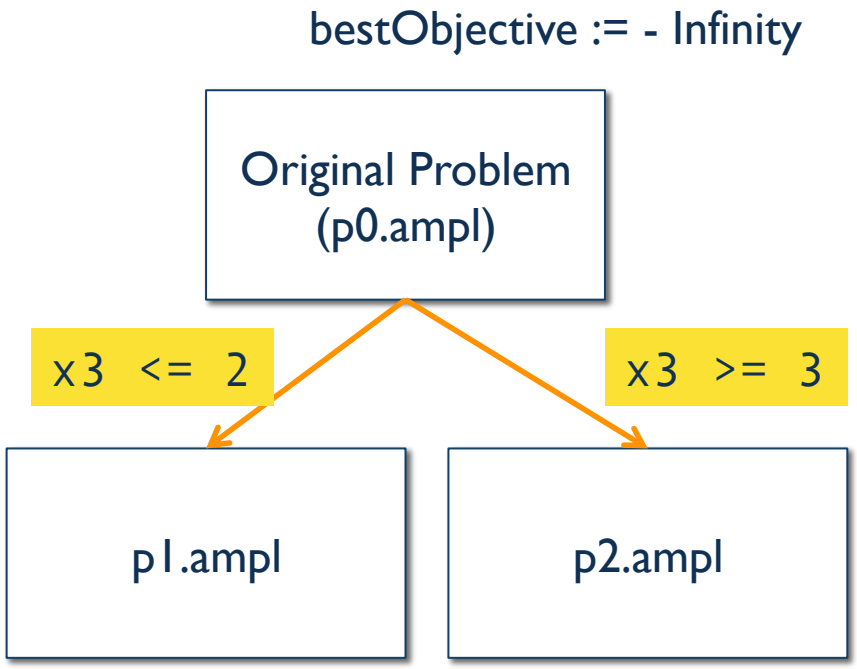
integer solution

x1.val = 0
x2.val = 1
x3.val = 4
x4.val = 1
x5.val = 0
x6.val = 0
Optimal Value: -4.200000

LP relaxation

x1.val = 0.333333333333333
x2.val = 0.666666666666667
x3.val = 2.66666666666667
x4.val = 0
x5.val = 0.333333333333333
x6.val = 0
Optimal Value: -3.333333

Initial Node



x1.val = 0.333333333333333

x2.val = 0.666666666666667

x3.val = 2.66666666666667

x4.val = 0

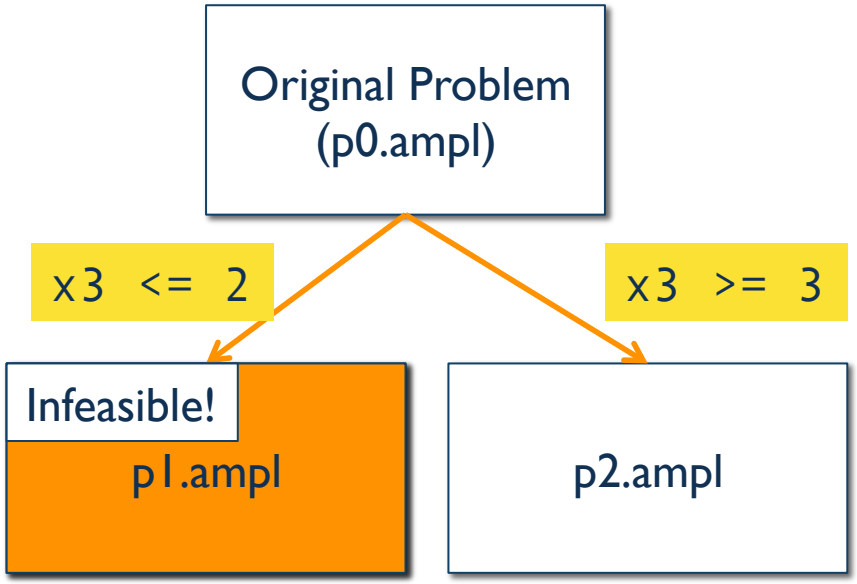
x5.val = 0.333333333333333

x6.val = 0

Optimal Value: -3.333333

Tree #1

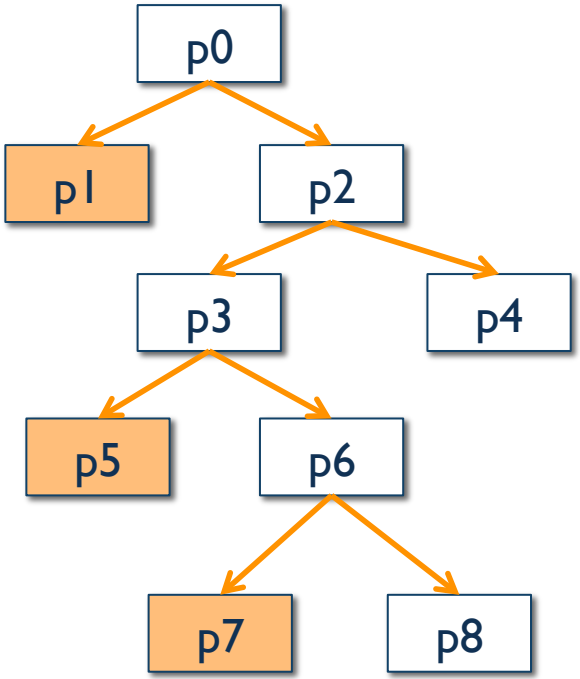
bestObjective := - Infinity



p2.ampl

x1.val = 0.4
x2.val = 0.55
x3.val = 3
x4.val = 0
x5.val = 0.4
x6.val = 0.05

BnB Tree



p3.ampl

x1.val = 1
x2.val = 0
x3.val = 4.571428571428571
x4.val = 0
x5.val = 0.7142857142857143
x6.val = 0.2857142857142857
Optimal Value: -3.857143

p8.ampl

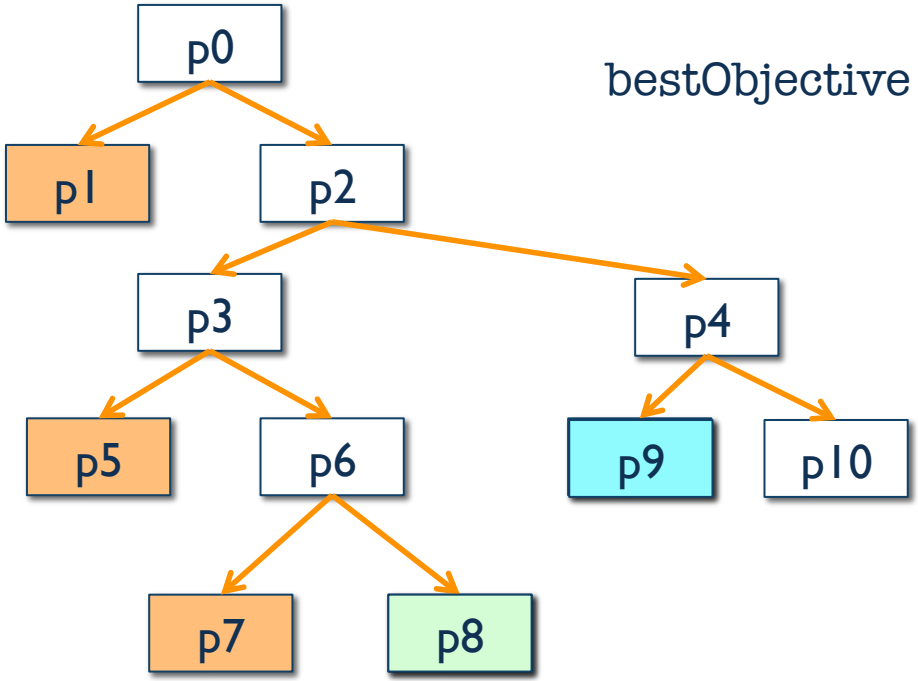
x1.val = 1
x2.val = 0
x3.val = 6
x4.val = 0
x5.val = 1
x6.val = 0.5
Optimal Value: -4.750000

p6.ampl

x1.val = 1
x2.val = 0
x3.val = 5
x4.val = 0.3333333333333333
x5.val = 0.6666666666666667
x6.val = 0.3333333333333333
Optimal Value: -4.066667

BnB Tree

p9.ampl
x1.val = 0
x2.val = 1
x3.val = 4
x4.val = 1
x5.val = 0
x6.val = 0
Optimal Value: -4.2

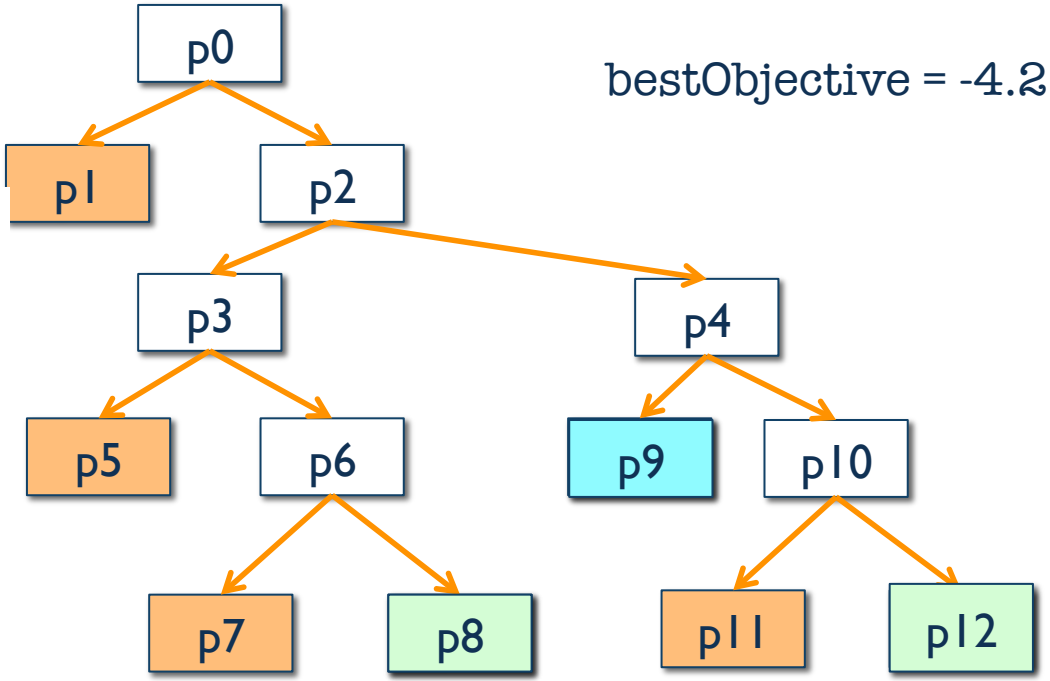


bestObjective = -4.2

p8.ampl
x1.val = 1
x2.val = 0
x3.val = 6
x4.val = 0
x5.val = 1
x6.val = 0.5
Optimal Value: -4.750000

BnB Tree

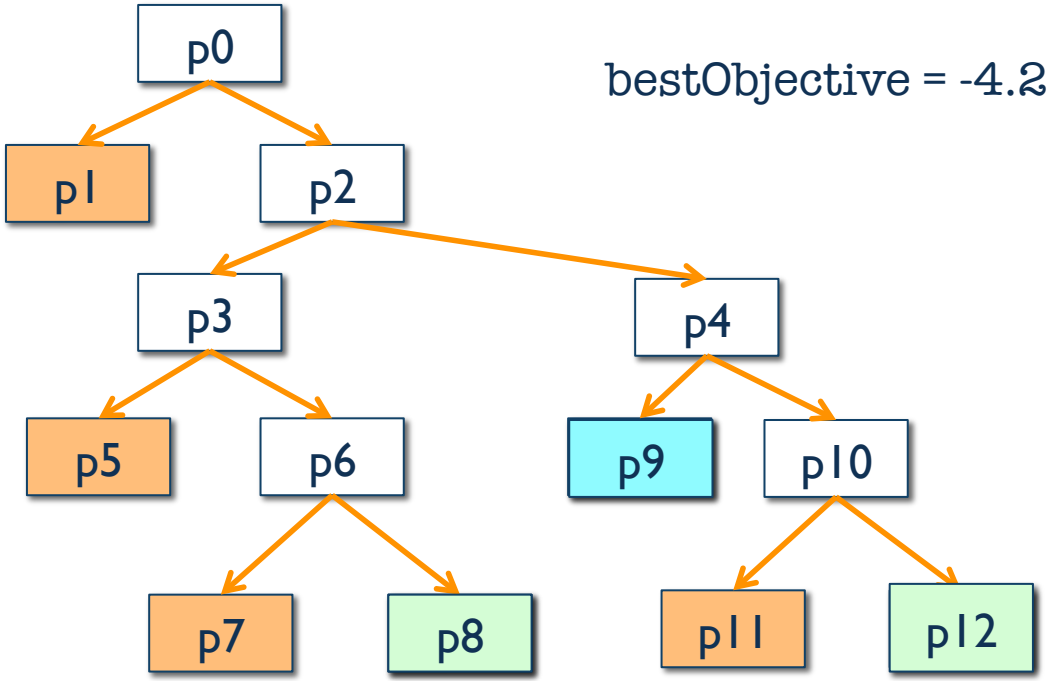
```
p12.ampl
x1.val = 0
x2.val = 1
x3.val = 6
x4.val = 0
x5.val = 1
x6.val = 0.5
Optimal Value: -5.750000
```



bestObjective = -4.2

BnB Final Tree

x1.val = 0
x2.val = 1
x3.val = 4
x4.val = 1
x5.val = 0
x6.val = 0
Optimal Value: -4.200000



bestObjective = -4.2

BRANCH-AND-BOUND METHOD

Heuristics

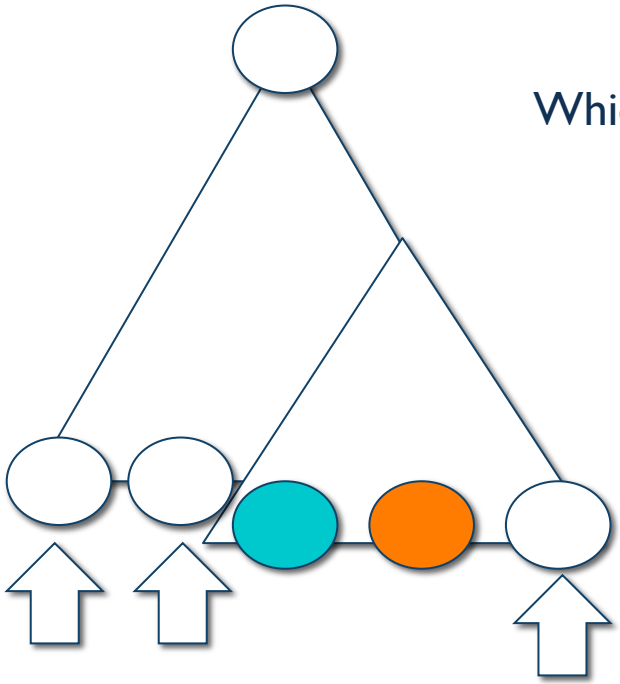
BnB choices to be made

- Which unexplored regular leaf node should I expand?
- How to choose the branching variables?
- Other tricks:
  - Randomized rounding to find an integer solution?
  - Carrying out BnB at the dictionary level.

BRANCH-AND-BOUND

Choice of unexplored node to expand

General Situation

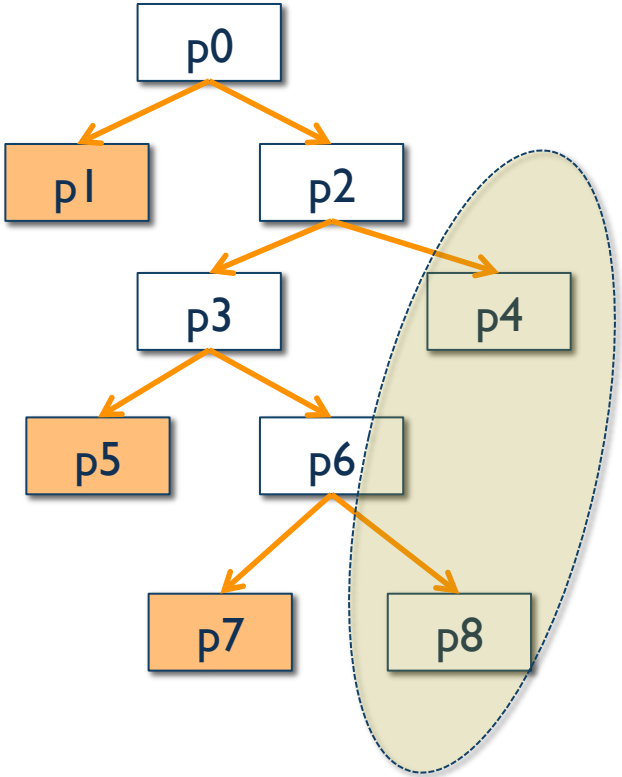


Which node should I examine first?

Unexplored “frontier”

Branch-and-bound

Q: Which one should we examine first?



Unexplored Nodes

Goal

- Minimize the number of nodes explored in the tree.

- Which node should I expand first:
  - Yield integral solutions.
  - Improve the value of bestObjective.

Unexplored Node Selection Heuristic

- Deepest node first.
  - Select the node that has largest depth in the tree to explore first.
- Best LP relaxation solution.
  - Select the node whose LP relaxation has the best answer.
- Breadth First
  - Search breadth-first.

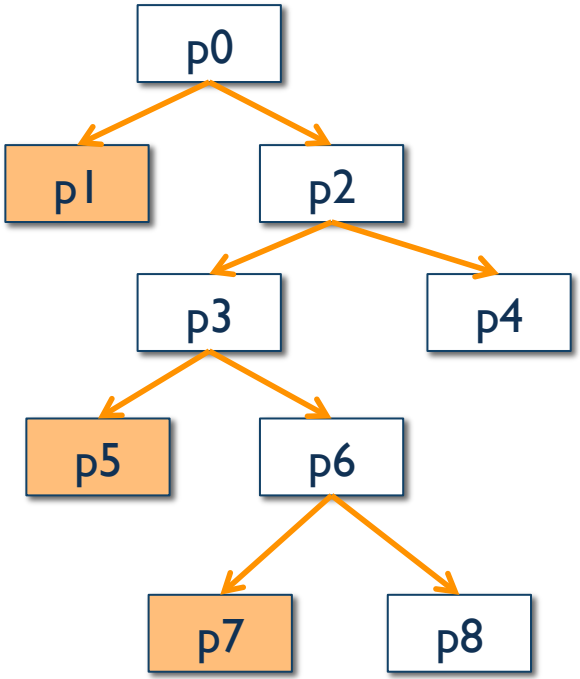
In Practice…

- ILPs come from some problem domain:
  - eg., We are converting a minimum cost vertex cover computation to ILP.
- Selection heuristics are best found by experimenting with a large set of problems from the domain.
- There is no general rule, unfortunately.

BRANCH-AND-BOUND

Choosing the branch variable

BnB Tree



p3.ampl

x1.val = 1
x2.val = 0
x3.val = 4.571428571428571
x4.val = 0
x5.val = 0.7142857142857143
x6.val = 0.2857142857142857
Optimal Value: -3.857143

p8.ampl

x1.val = 1
x2.val = 0
x3.val = 6
x4.val = 0
x5.val = 1
x6.val = 0.5
Optimal Value: -4.750000

p6.ampl

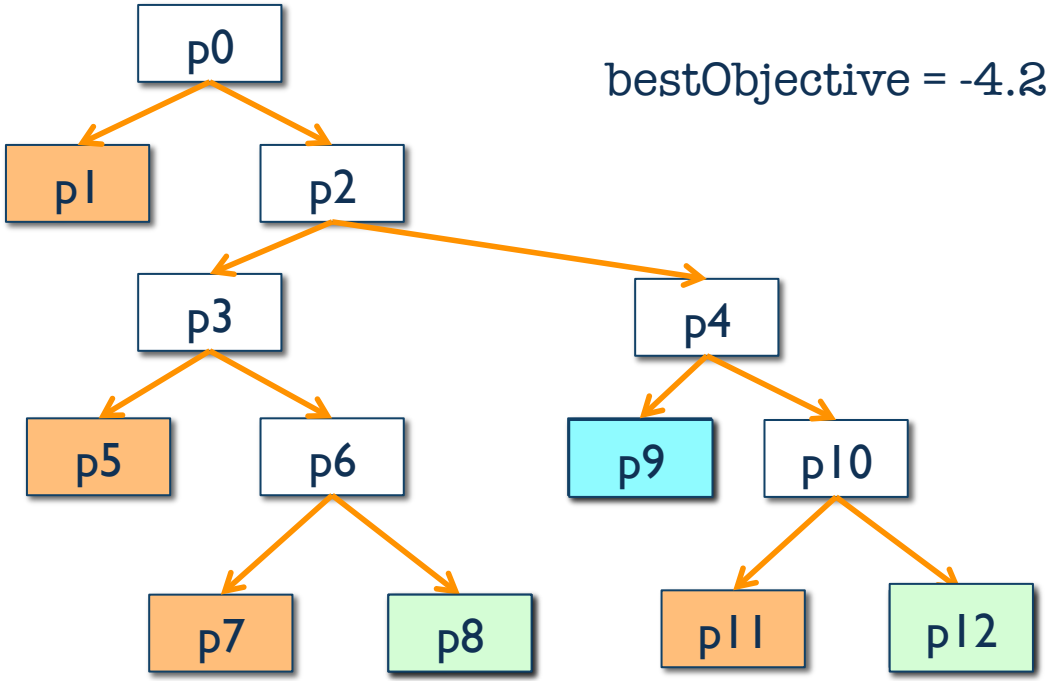
x1.val = 1
x2.val = 0
x3.val = 5
x4.val = 0.3333333333333333
x5.val = 0.6666666666666667
x6.val = 0.3333333333333333
Optimal Value: -4.066667

Original Problem

```
var x1 integer  >= 0, <= 10;
var x2 integer  >= 0, <= 10;
var x3  integer >= 0, <= 10;
var x4 integer  >= 0, <= 10;
var x5  integer >= 0, <= 10;
var x6  integer >= 0, <= 10;
maximize obj: -x1 - 2* x2 -0.5 * x3 - 0.2* x4 - x5 +0.5* x6;
c1: x1 + 2 * x2 >= 1;
c2: x1 + x2 + 3* x6 >= 1;
c3: x1 + x2 + x6 >= 1;
c4: x3 - 3* x4  >= 1;
c5: x3 - 2* x4 -5* x5 >= 1;
c6: x4 + 3* x5 -4 *x6   >= 1;
c7: x2 + x5 + x6 >= 1;
solve;
display x1, x2,x3, x4, x5, x6;
end;
```

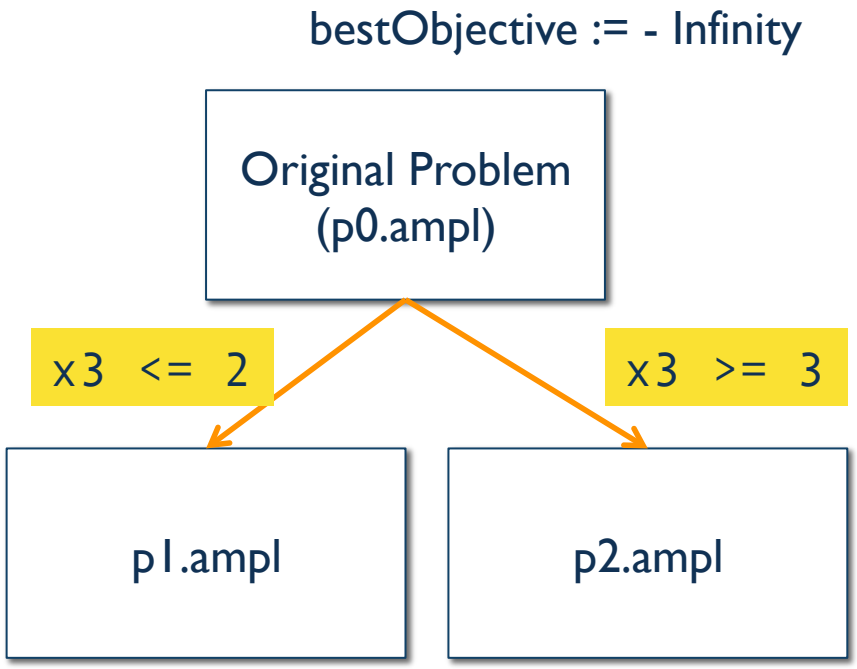
BnB Final Tree

x1.val = 0
x2.val = 1
x3.val = 4
x4.val = 1
x5.val = 0
x6.val = 0
Optimal Value: -4.200000



bestObjective = -4.2

Initial Node



x1.val = 0.333333333333333

x2.val = 0.666666666666667

x3.val = 2.66666666666667

x4.val = 0

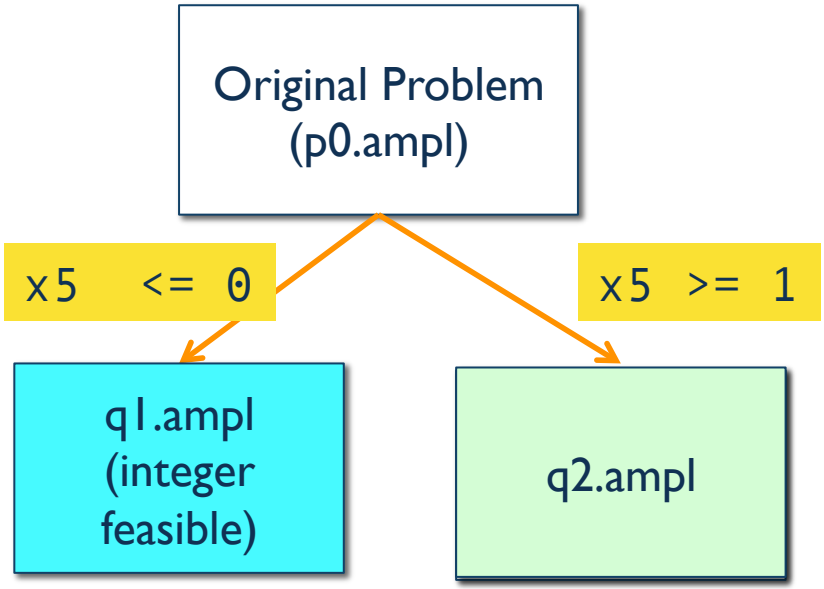
x5.val = 0.333333333333333

x6.val = 0

Optimal Value: -3.333333

What if we chose a different branch variable..?

bestObjective := - Infinity




```
x1.val = 0
x2.val = 0.5
x3.val = 6
x4.val = 0
x5.val = 1
x6.val = 0.5
Optimal Value: -4.750000
```

bestObjective := -4.2

How to select a branch variable?

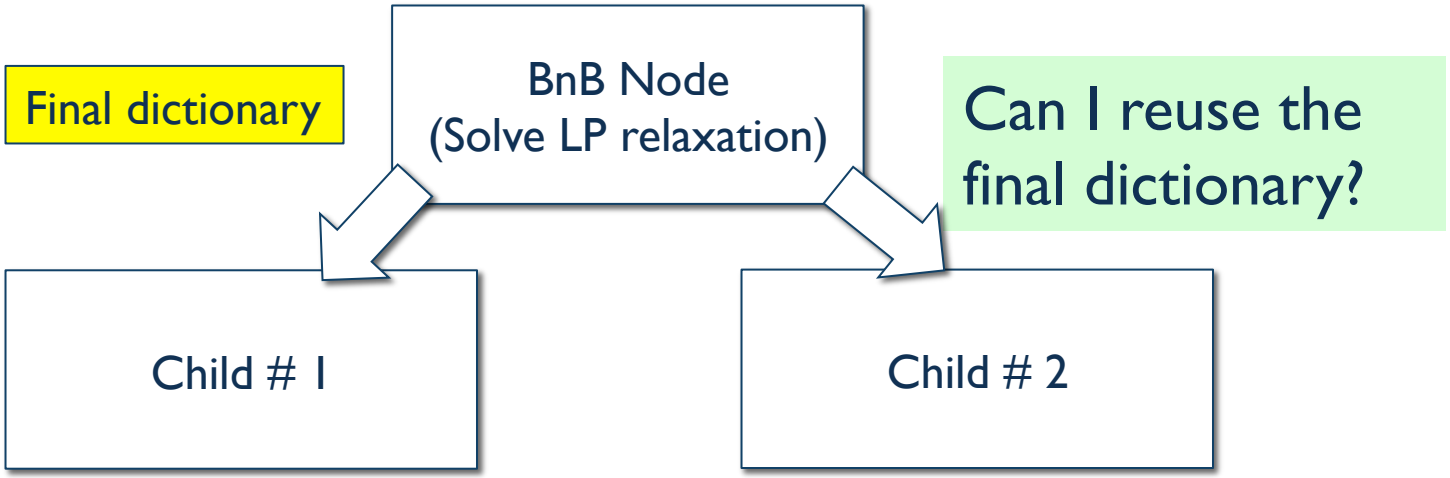
- The choice of a branch variable is very important.
- There is no single "best heuristic".
- Often, some expertise with the problem domain informs the heuristic.

BRANCH-AND-BOUND

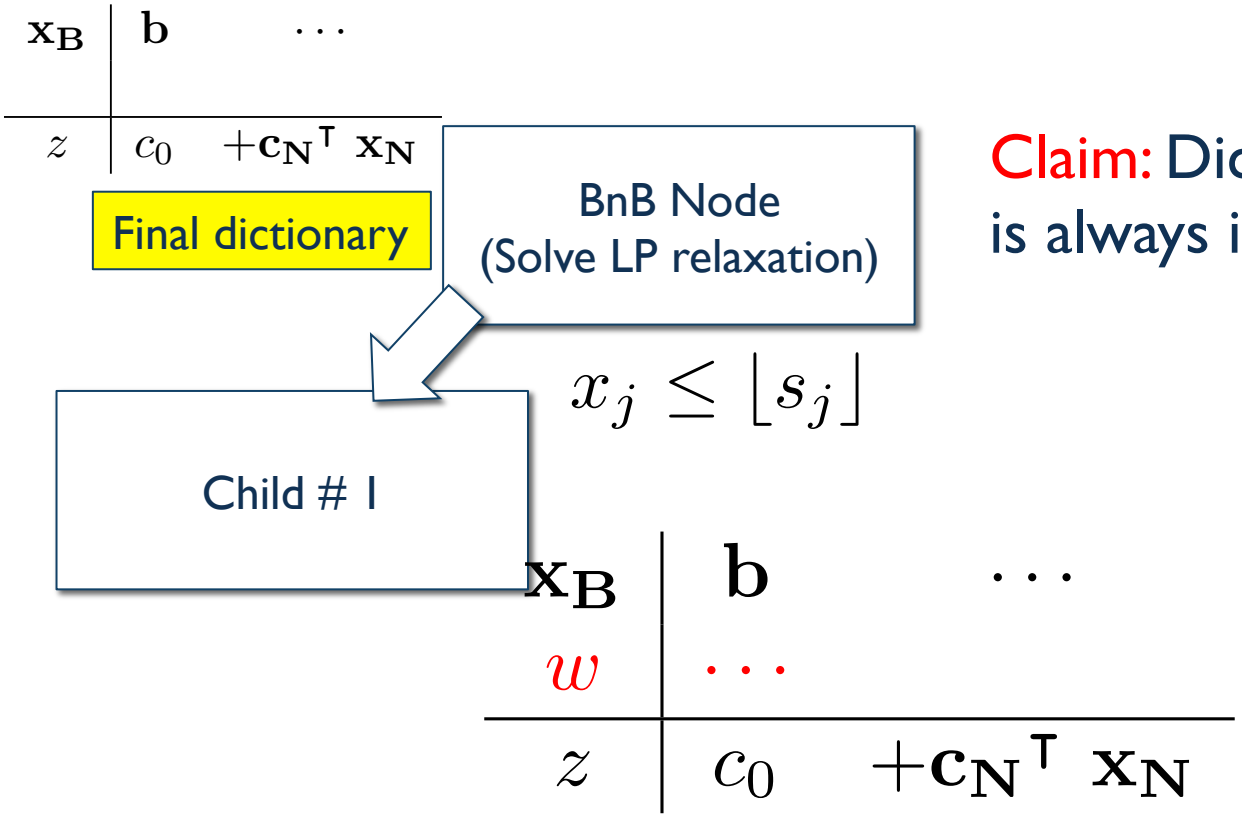
At the dictionary level

Thus far

- Use LP solver as a black box.
- This lecture:
  - Consider how to fold branch-and-bound in a dictionary setup.

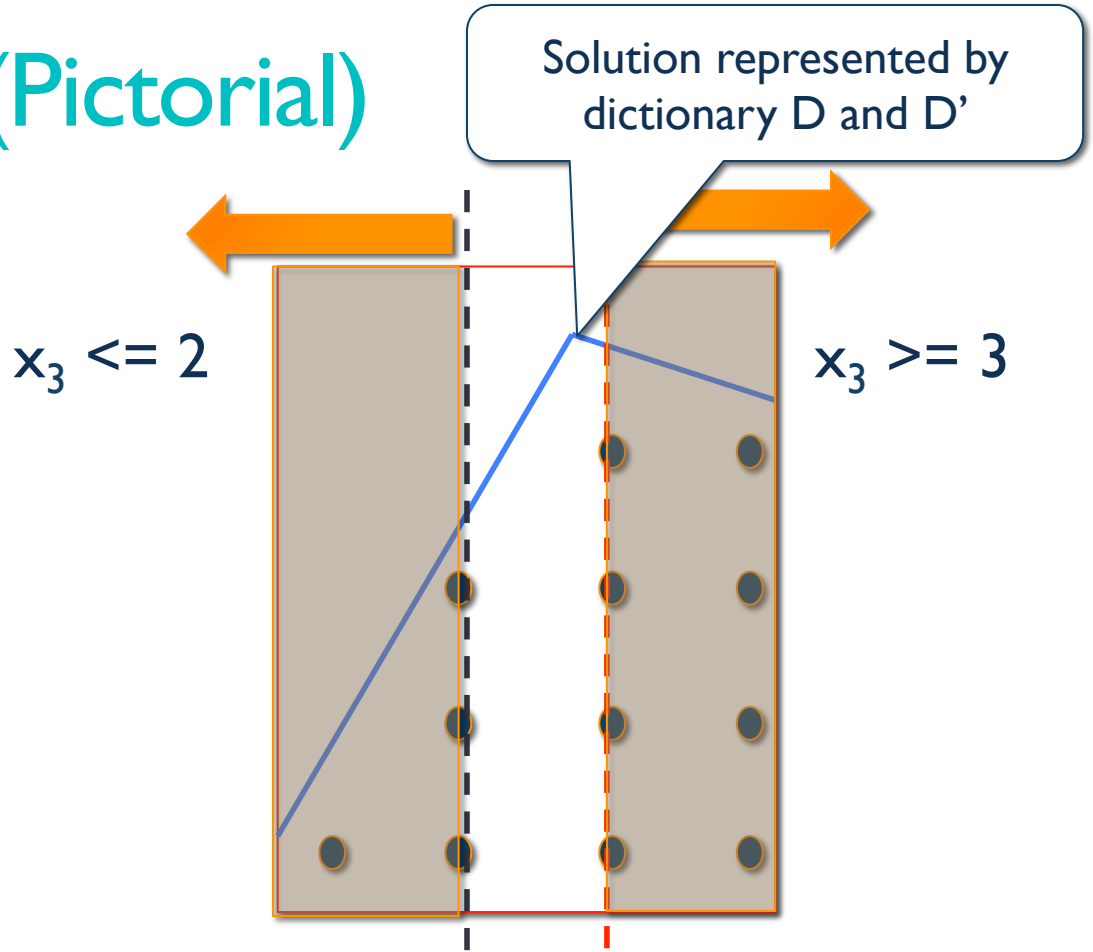


View from the dictionary



Claim: Dictionary D' is always infeasible.

Proof (Pictorial)



Example (ILP)

```
var x1 >= 0, <= 10;
var x2  >= 0, <= 10;
var x3  >= 0, <= 10;

maximize obj: x1 + x2 -5* x3 ;

c1: -2* x1 + 7 *x2 <= 1;
c2: x1 - 2 *x2 + 5* x3 <= 3;
c3: x1 + x2 - 3 * x3 <= 7;

solve;
display x1, x2, x3;
end;
```

$$
\begin{aligned}
x_4 &: 10 - x_1 \\
x_5 &: 10 - x_2 \\
x_6 &: 10 - x_3 \\
x_7 &: 1 + 2x_1 - 7x_2 \\
x_8 &: 3 - x_1 + 2x_2 - 5x_3 \\
x_9 &: 7 - x_1 - x_2 + 3x_3
\end{aligned}
$$

Slack variables are also integers.

Final Dictionary

x_4	4.33333333333	$+0.333333x_8$	$+0.666667x_9$	$-0.333333x_3$
x_5	8.66666666667	$-0.333333x_8$	$+0.333333x_9$	$-2.666667x_3$
x_6	10			$-x_3$
x_7	3	$-3x_8$	$+x_9$	$-18x_3$
x_1	5.66666666667	$-0.333333x_8$	$-0.666667x_9$	$+0.333333x_3$
x_2	1.33333333333	$+0.333333x_8$	$-0.333333x_9$	$+2.666667x_3$
z	7.0	$+0x_8$	$-x_9$	$-2x_3$

$x_1 \geq 6$  $x_{10} : \ -6 \ + \ x_1$

Dictionary After Branch

$$x_{10} : \ -6 \ + \ x_1$$

x_4	4.3333333333	$+0.333333x_8$	$+0.666667x_9$	$-0.333333x_3$
x_5	8.66666666667	$-0.333333x_8$	$+0.333333x_9$	$-2.666667x_3$
x_6	10			$-x_3$
x_7	3	$-3x_8$	$+x_9$	$-18x_3$
x_1	5.66666666667	$-0.333333x_8$	$-0.666667x_9$	$+0.333333x_3$
x_2	1.33333333333	$+0.333333x_8$	$-0.333333x_9$	$+2.666667x_3$
x_{10}	-0.33333	$-0.333333x_8$	$-0.666667x_9$	$+0.333333x_3$
z	7.0	$+0x_8$	$-x_9$	$-2x_3$

Dictionary becomes primal infeasible.

Back to initialization phase Simplex?

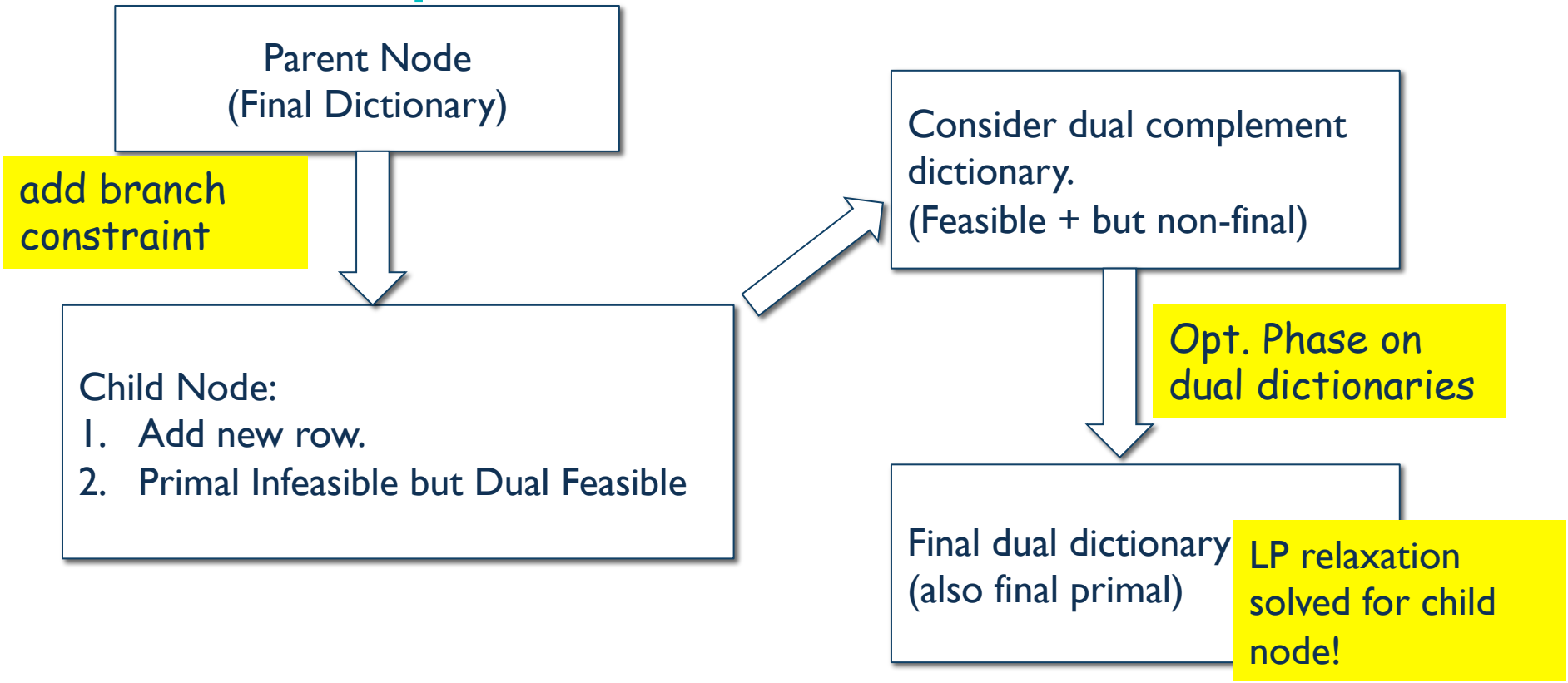
Dictionary After Branch

x_4	4.3333333333	$+0.333333x_8$	$+0.666667x_9$	$-0.333333x_3$
x_5	8.66666666667	$-0.333333x_8$	$+0.333333x_9$	$-2.666667x_3$
x_6	10			$-x_3$
x_7	3	$-3x_8$	$+x_9$	$-18x_3$
x_1	5.66666666667	$-0.333333x_8$	$-0.666667x_9$	$+0.333333x_3$
x_2	1.33333333333	$+0.333333x_8$	$-0.333333x_9$	$+2.666667x_3$
x_{10}	-0.33333	$-0.333333x_8$	$-0.666667x_9$	$+0.333333x_3$
z	7.0	$+0x_8$	$-x_9$	$-2x_3$

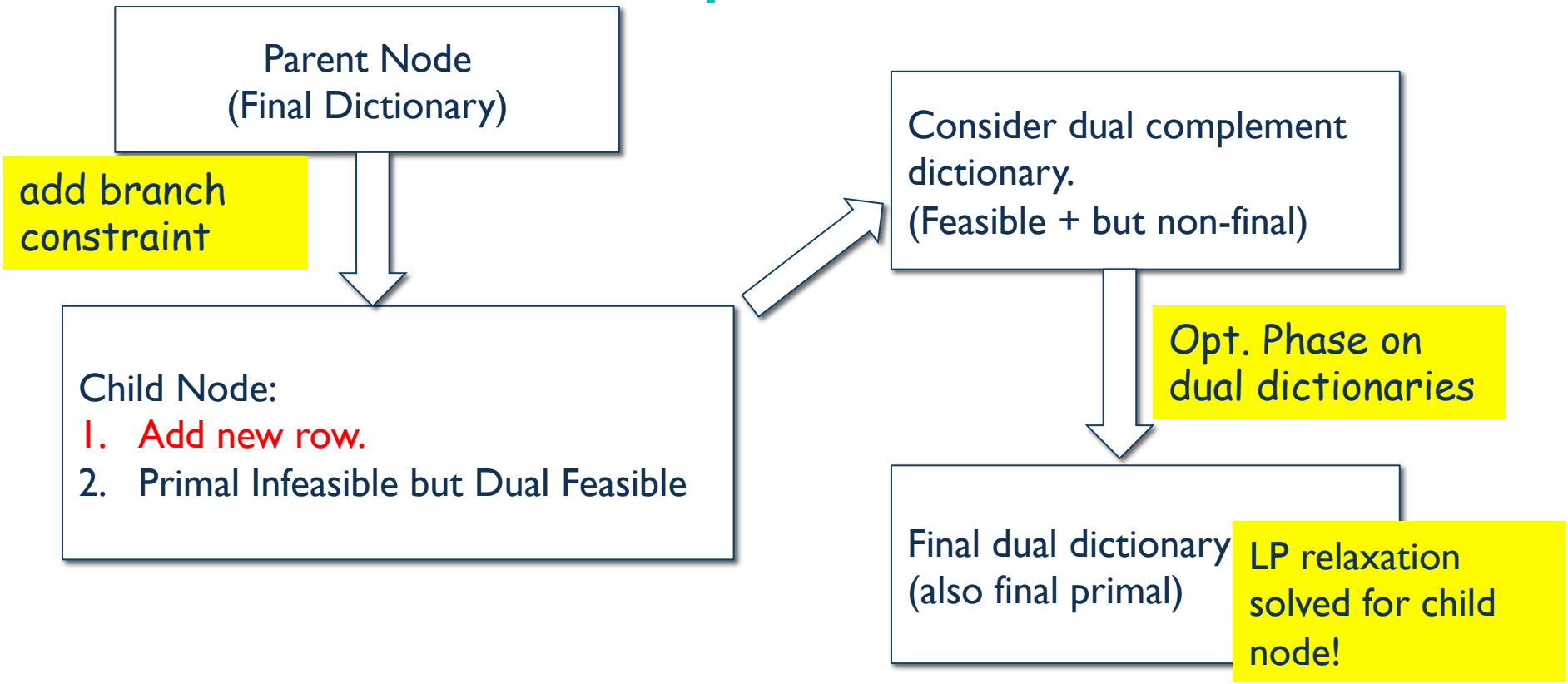
Dictionary becomes primal infeasible.

Dual Feasible Dictionary!!
But not dual final.

How to update after branch?



General Form Simplex



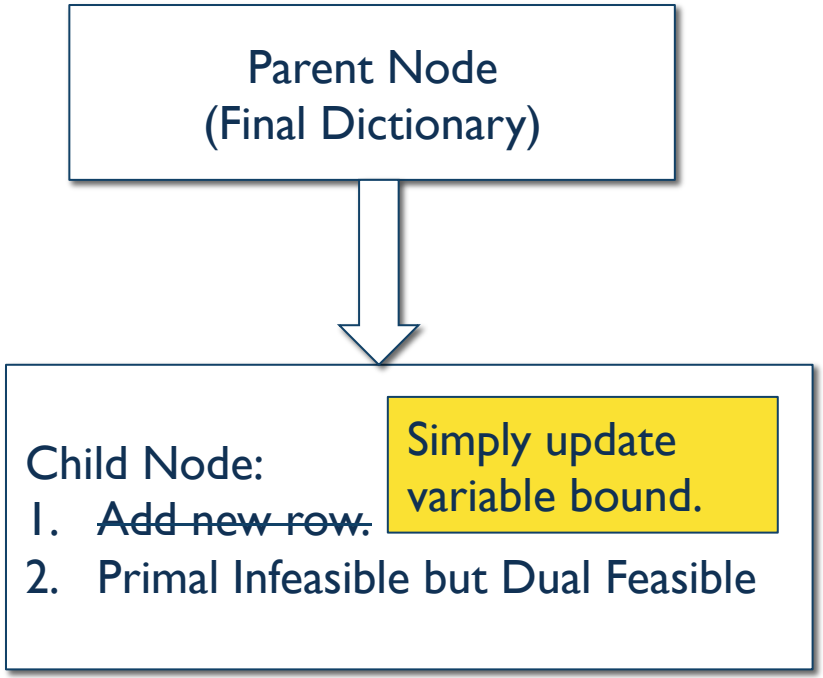
General Form Dictionary

- Give special treatment to bounds on variables.

$$\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$$

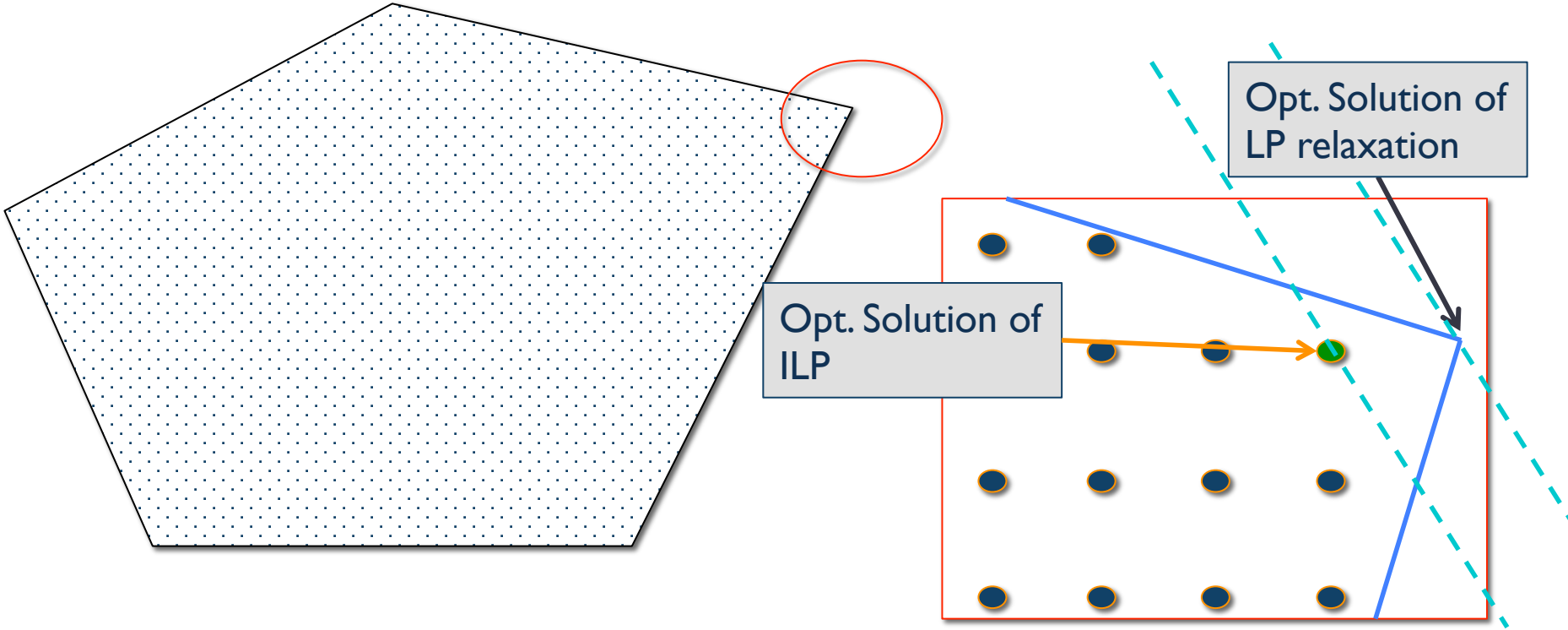
- Modify the Simplex algorithm in two ways:
  - Dictionaries now have special ways to track bounds.
  - Pivoting is modified.

Branch-and-Bound with General Form



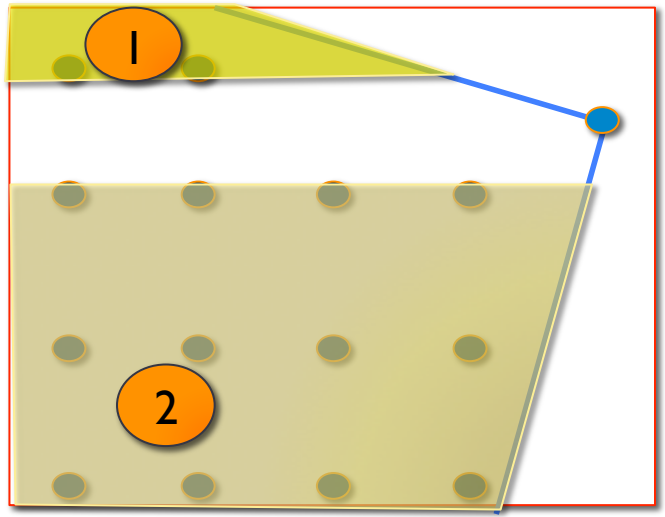
CUTTING PLANE METHOD

LP Relaxation for ILP

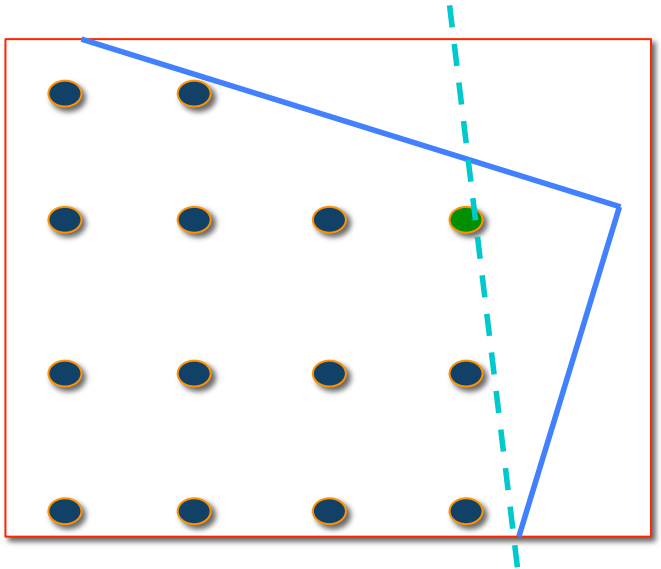


Removing a Fractional Solution

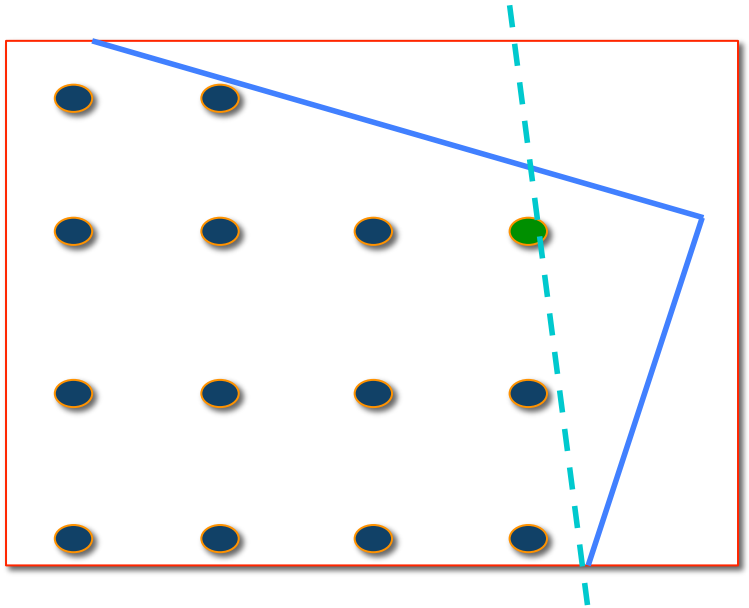
Branch and Bound



Cutting Plane Method.



Cutting Plane



How do we find a cutting plane?

GOMORY-CHVATAL CUTTING PLANE

Part 1: Setting up the problem.

ILP in Standard Form

$$
\begin{aligned}
\max \quad & \mathbf{c}^\intercal \, \mathbf{x} & & \\
\text{s.t.} \quad & A\mathbf{x} & \leq & \mathbf{b} \\
& \mathbf{x} & \geq & \mathbf{0} \\
& \mathbf{x} & \in & \mathbb{Z}
\end{aligned}
$$

A, b, c are all assumed to be integers.

Conversion to standard from

- Recap from LP formulations lectures:
  - Equality constraints into two inequalities.
  - Rewrite in case $x_i \geq 0$ is missing:

$$x_i \mapsto x_i^+ - x_i^-$$

  - Convert ≥ to ≤ by negating both sides.

- How do we make sure A, b, c are integers?

Reasonable assumption: original problem coefficients are rationals.

Integer Coefficients.

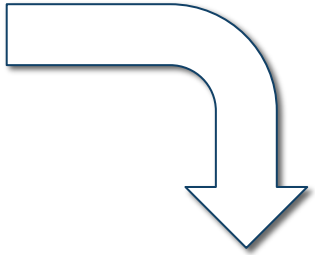
Scale constraints + Objective

$$
\begin{array}{lrrrcll}
\max & 2x_1 & +0.3x_2 & -0.1x_3 & & & \\
\text{s.t.} & 0.1x_1 & -2x_2 & -x_3 & \leq & 0.25 & \\
 & 0.5x_1 & -2.6x_2 & +1.3x_3 & \leq & 0.15 & \\
 & x_1, & x_2, & x_3 & \geq & 0 & \\
 & x_1, & x_2, & \boldsymbol{x_3} & \in & \mathbb{Z} &
\end{array}
$$

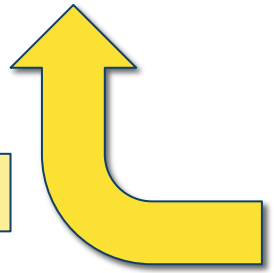
$$
\begin{array}{lrrrcll}
\max & 2x_1 & +0.3x_2 & -0.1x_3 & & & \times 10 \\
\text{s.t.} & 0.1x_1 & -2x_2 & -x_3 & \leq & 0.25 & \times 20 \\
 & 0.5x_1 & -2.6x_2 & +1.3x_3 & \leq & 0.15 & \times 100 \\
 & x_1, & x_2, & x_3 & \geq & 0 & \\
 & x_1, & x_2, & x_4 & \in & \mathbb{Z} &
\end{array}
$$

Conversion to Integer Coefficients.

$$
\begin{array}{lrrrcr}
\max & 2x_1 & +0.3x_2 & -0.1x_3 & & \\
\text{s.t.} & 0.1x_1 & -2x_2 & -x_3 & \leq & 0.25 \\
 & 0.5x_1 & -2.6x_2 & +1.3x_3 & \leq & 0.15 \\
 & x_1, & x_2, & x_3 & \geq & 0 \\
 & x_1, & x_2, & x_4 & \in & \mathbb{Z}
\end{array}
$$



Divide result by 10



$$
\begin{array}{lrrrcr}
\max & 20x_1 & +3x_2 & -x_3 & & \\
\text{s.t.} & 2x_1 & -40x_2 & -20x_3 & \leq & 5 \\
 & 50x_1 & -260x_2 & +130x_3 & \leq & 15 \\
 & x_1, & x_2, & x_3 & \geq & 0 \\
 & x_1, & x_2, & x_4 & \in & \mathbb{Z}
\end{array}
$$

Adding Slack Variables

$$\begin{array}{lrcl} \max & \mathbf{c}^\intercal\ \mathbf{x} & & \\ \text{s.t.} & A\mathbf{x} & \leq & \mathbf{b} \\ & \mathbf{x} & \geq & \mathbf{0} \\ & \mathbf{x} & \in & \mathbb{Z} \end{array}$$

$$\begin{array}{lrcl} \max & \mathbf{c}^\intercal\ \mathbf{x} & & \\ \text{s.t.} & A\mathbf{x} + \mathbf{x_s} & = & \mathbf{b} \\ & \mathbf{x}, \mathbf{x_s} & \geq & \mathbf{0} \\ & \mathbf{x}, \mathbf{x_s} & \in & \mathbb{Z} \end{array}$$

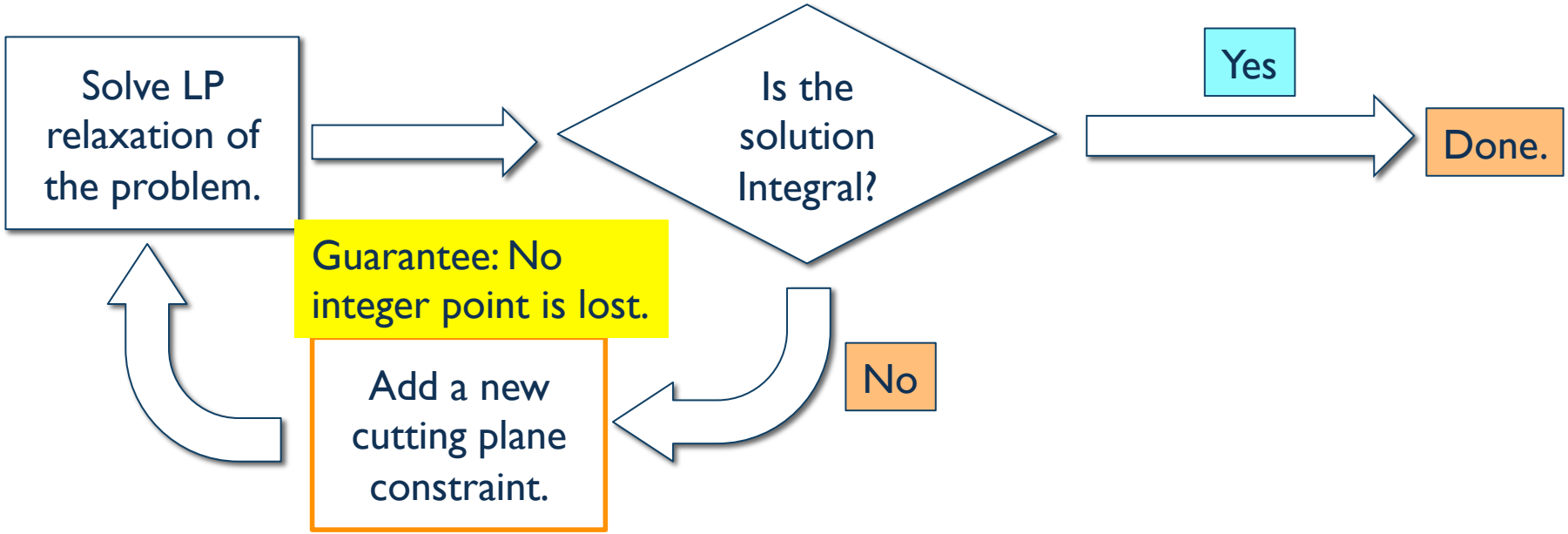
Summary

- We assume problem is in LP standard form.
  - Assume coefficients are rational.
- ILP standard form: coefficients (A,b,c) are integers.
  - Scale the original rational problem.
  - Make sure that result is divided by scale factor for objective.
- Advantage of ILP standard form:
  - Slack variables are naturally integers.
  - No need for separate treatment of decision and slack variables.

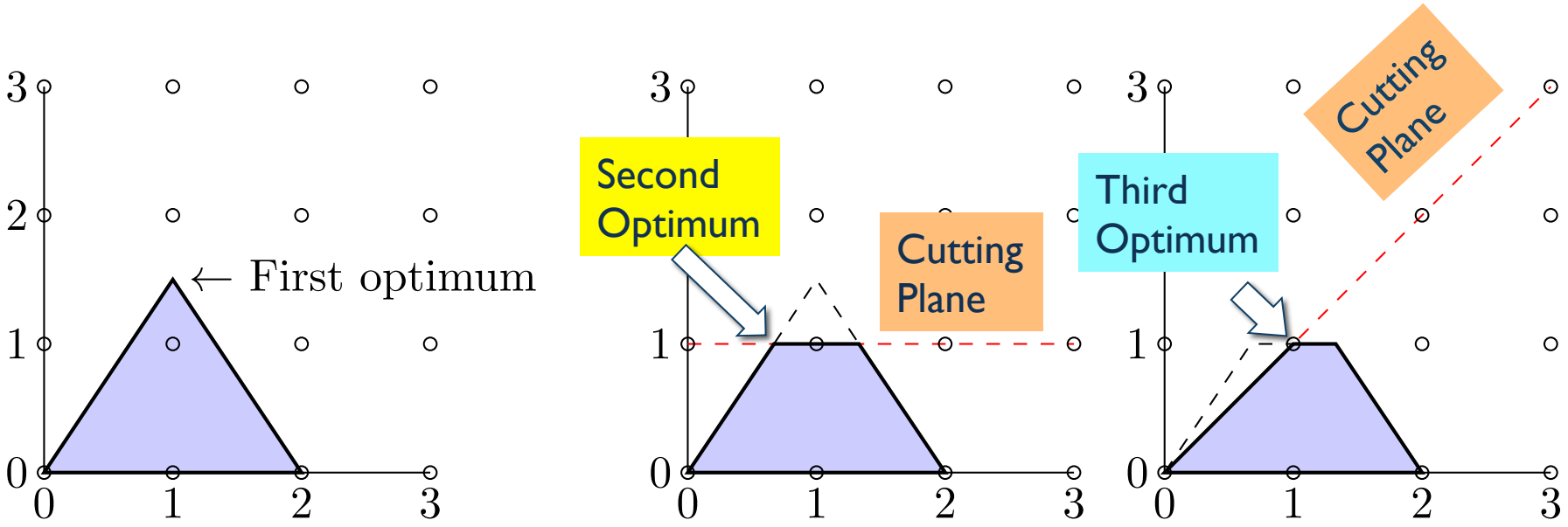
CUTTING PLANE METHOD

Part 2: Gomory-Chvatal Cuts

Overall method



Cutting Plane Visualization



Idea: cutting plane attempts to successively "expose" an integer optimal vertex.

Cutting Plane Method

- Deriving the Cutting Plane
  - Gomory-Chvatal Cuts

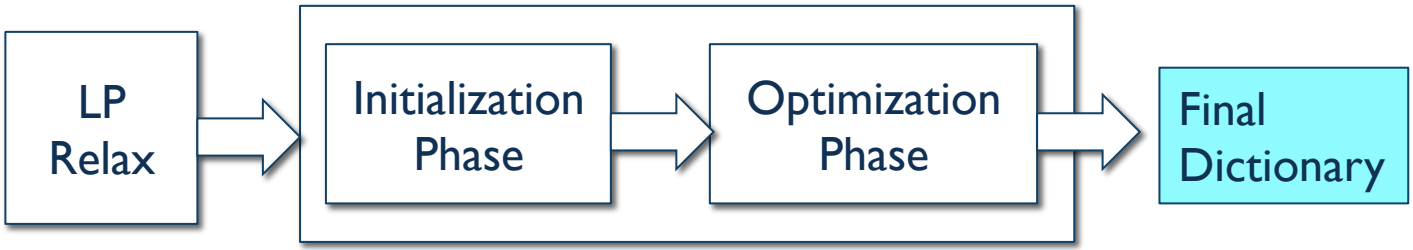
- Integrating the method inside Simplex.
  - Using the dual dictionary to avoid initialization phase Simplex.

GOMORY-CHVATAL CUTS

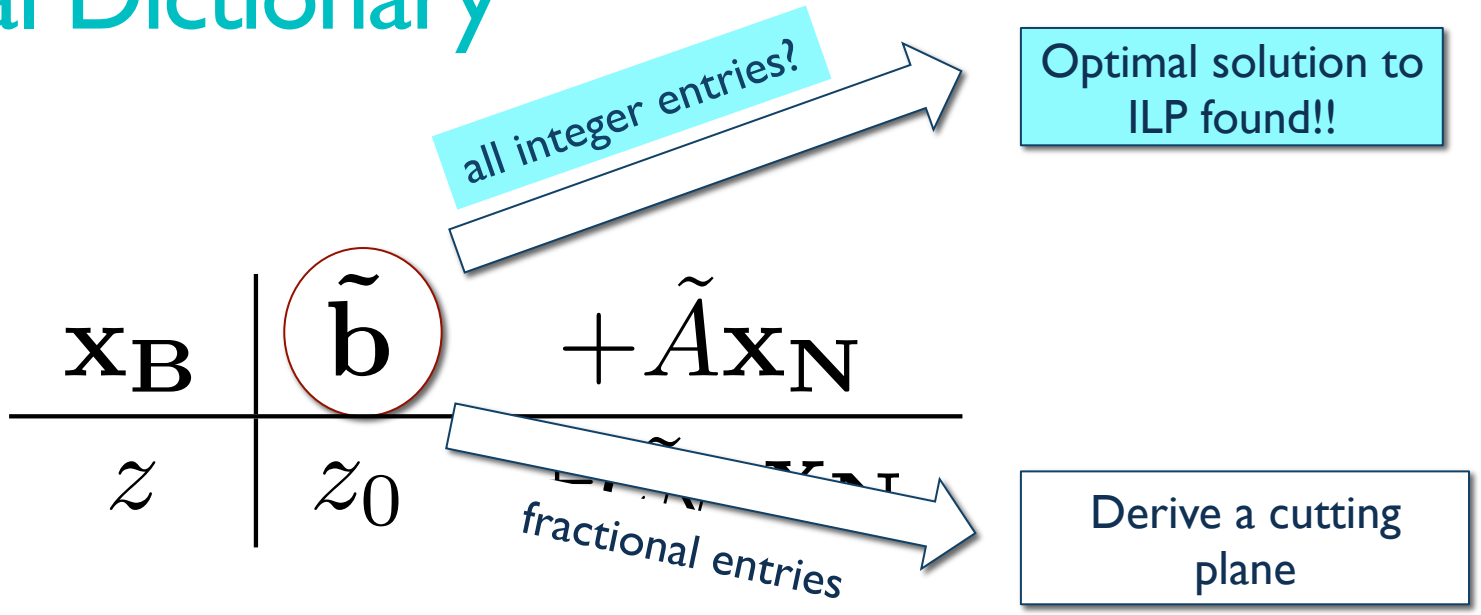
Overall Idea

$$
\begin{array}{lrcl}
\max & \mathbf{c}^\intercal\ \mathbf{x} & & \\
\text{s.t.} & A\mathbf{x} + \mathbf{x_s} & = & \mathbf{b} \\
& \mathbf{x}, \mathbf{x_s} & \geq & \mathbf{0} \\
& \mathbf{x}, \mathbf{x_s} & \in & \mathbb{Z}
\end{array}
$$

1. Solve the LP relaxation using Simplex algorithm.



Final Dictionary



Example: Final Dictionary

x_1	1.2	$-3.1x_2$	$+4.3x_3$	$-0.5x_5$
x_4	1	$-x_2$	$+x_3$	$-x_5$
x_6	2.5	$+1.3x_2$	$-2.1x_3$	$+x_5$
z	1.7	$-1.2x_2$	$-2.3x_3$	$-2.1x_5$

$$x_1 = 1.2,\ x_2 = 0,\ x_3 = 0,\ x_4 = 1,\ x_5 = 0,\ x_6 = 2.5$$

Cutting Plane Derivation: Step 1

Step 1 Identify row with fractional constant.

x_1	1.2	$-3.1x_2$	$+4.3x_3$	$-0.5x_5$
x_4	1	$-x_2$	$+x_3$	$-x_5$
x_6	2.5	$+1.3x_2$	$-2.1x_3$	$+x_5$
z	1.7	$-1.2x_2$	$-2.3x_3$	$-2.1x_5$

$$x_1 + 3.1x_2 - 4.3x_3 + 0.5x_5 = 1.2$$

Cutting Plane Derivation: Step 2

$$x_1 + 3.1x_2 - 4.3x_3 + 0.5x_5 = 1.2$$

$$\underbrace{(x_1 + 3x_2 - 5x_3 + 0x_5)}_{A} + \underbrace{(0.1x_2 + 0.7x_3 + 0.5x_5)}_{B} = 1 + 0.2$$

A: integer

B: integer + fraction

B ≥ 0

A + B = 1.2

Claim

$$\underbrace{(\quad}_{A} \underbrace{\quad 2 + 0.7x_3 + 0.5x_5)}_{B} = 1 + 0.2$$

A

B

B

A

Conclusion:

1. Fractional part of B is 0.2
2. B ≥ 0.2

In other words,

1. B – 0.2 is an integer.
2. B - 0.2 ≥ 0

B	A+B	Possible
0.2	1.2	YES
-0.8	1.2	No
0.1	1.2	No
-2.8	1.2	No
4.2	1.2	Yes
101.2	1.2	Yes

Cutting Plane

x_1	1.2	$-3.1x_2$	$+4.3x_3$	$-0.5x_5$
x_4	1	$-x_2$	$+x_3$	$-x_5$
x_6	2.5	$+1.3x_2$	$-2.1x_3$	$+x_5$
z	1.7	$-1.2x_2$	$-2.3x_3$	$-2.1x_5$

Cutting Plane:

$$0.1x_2 + 0.7x_3 + 0.5x_5 \geq 0.2$$

Cutting Plane: Definition.

Final Dictionary (LP Relax)

$$
\begin{array}{rcllllll}
x_{B1} & = & b_1 & +a_{11}x_{I1} & +\cdots & +a_{1j}x_{Ij} & +\cdots & +a_{1n}x_{In} \\
 & \vdots & & & & & & \\
x_{Bk} & = & b_k & +a_{k1}x_{I1} & +\cdots & +a_{kj}x_{Ij} & +\cdots & +a_{kn}x_{In} \\
 & \vdots & & & & & & \\
x_{Bm} & = & b_m & +a_{m1}x_{I1} & +\cdots & +a_{mj}x_{Ij} & +\cdots & +a_{mn}x_{In} \\
\hline
z & = & c_0 & +c_1x_{I1} & +\cdots & +c_jx_{Ij} & +\cdots & +c_nx_{In}
\end{array}
$$

$b_k \notin \mathbb{Z}$

Cutting Plane

$$\text{frac}(-a_{k1})x_{I1} + \cdots + \text{frac}(-a_{kn})x_{In} \geq \text{frac}(b_k)$$

$$\text{frac}(x) = x - \lfloor x \rfloor$$

$$
\begin{array}{lcllllll}
x_{B1} & = & b_1 & +a_{11}x_{I1} & +\cdots & +a_{1j}x_{Ij} & +\cdots & +a_{1n}x_{In} \\
 & \vdots & & & & & & \\
x_{Bk} & = & b_k & +a_{k1}x_{I1} & +\cdots & +a_{kj}x_{Ij} & +\cdots & +a_{kn}x_{In} \\
 & \vdots & & & & & & \\
x_{Bm} & = & b_m & +a_{m1}x_{I1} & +\cdots & +a_{mj}x_{Ij} & +\cdots & +a_{mn}x_{In} \\
\hline
z & = & c_0 & +c_1x_{I1} & +\cdots & +c_jx_{Ij} & +\cdots & +c_nx_{In}
\end{array}
$$

$b_k \notin \mathbb{Z}$

Cutting Plane

$$\text{frac}(-a_{k1})x_{I1} + \cdots + \text{frac}(-a_{kn})x_{In} \geq \text{frac}(b_k)$$

Claim: Every (integer) feasible point of the ILP satisfies the cutting plane constraint.

We have

$$x_{Bk} + \sum_{j=1}^{n} (-a_{kj}) x_{Ij} = b_k \tag{1}$$

Since $-a_{kj} \geq \lfloor -a_{kj} \rfloor$ and $x_{Ij} \geq 0$, we obtain

$$\sum_{j=1}^{n} (-a_{kj}) x_{Ij} \geq \sum_{j=1}^{n} \lfloor -a_{kj} \rfloor x_{Ij}$$

Applying this to Eq.(1) we obtain:

$$x_{Bk} + \sum_{j=1}^{n} (-a_{kj}) x_{Ij} \;\geq\; x_{Bk} + \sum_{j=1}^{n} \lfloor -a_{kj} \rfloor x_{Ij} \tag{2}$$

Therefore, we get

$$b_k \geq x_{Bk} + \sum_{j=1}^{n} \lfloor -a_{kj} \rfloor x_{Ij} \tag{3}$$

The RHS of (3) is an integer, therefore, we conclude:

$$\lfloor b_k \rfloor \geq x_{Bk} + \sum_{j=1}^{n} \lfloor -a_{kj} \rfloor x_{Ij} \tag{4}$$

Combining (4) with (1), we have,

$$b_k - \lfloor b_k \rfloor \leq \textstyle\sum_{j=1}^{n} (-a_{kj} - \lfloor -a_{kj} \rfloor) x_{Ij}$$

In other words, $\sum_{j=1}^{n} \mathsf{frac}(-a_{kj}) x_{Ij} \geq \mathsf{frac}(b_k)$

$$\begin{array}{lcllllll}
x_{B1} & = & b_1 & +a_{11}x_{I1} & +\cdots & +a_{1j}x_{Ij} & +\cdots & +a_{1n}x_{In} \\
 & \vdots & & & & & & \\
x_{Bk} & = & b_k & +a_{k1}x_{I1} & +\cdots & +a_{kj}x_{Ij} & +\cdots & +a_{kn}x_{In} \\
 & \vdots & & & & & & \\
x_{Bm} & = & b_m & +a_{m1}x_{I1} & +\cdots & +a_{mj}x_{Ij} & +\cdots & +a_{mn}x_{In} \\
\hline
z & = & c_0 & +c_1x_{I1} & +\cdots & +c_jx_{Ij} & +\cdots & +c_nx_{In}
\end{array}$$

$b_k \notin \mathbb{Z}$

Example

x_1	1.2	$-3.1x_2$	$+4.3x_3$	$-0.5x_5$
x_4	1	$-x_2$	$+x_3$	$-x_5$
x_6	2.5	$+1.3x_2$	$-2.1x_3$	$+x_5$
z	1.7	$-1.2x_2$	$-2.3x_3$	$-2.1x_5$

$$0.7x_2 + 0.1x_3 + 0x_5 \geq 0.5$$

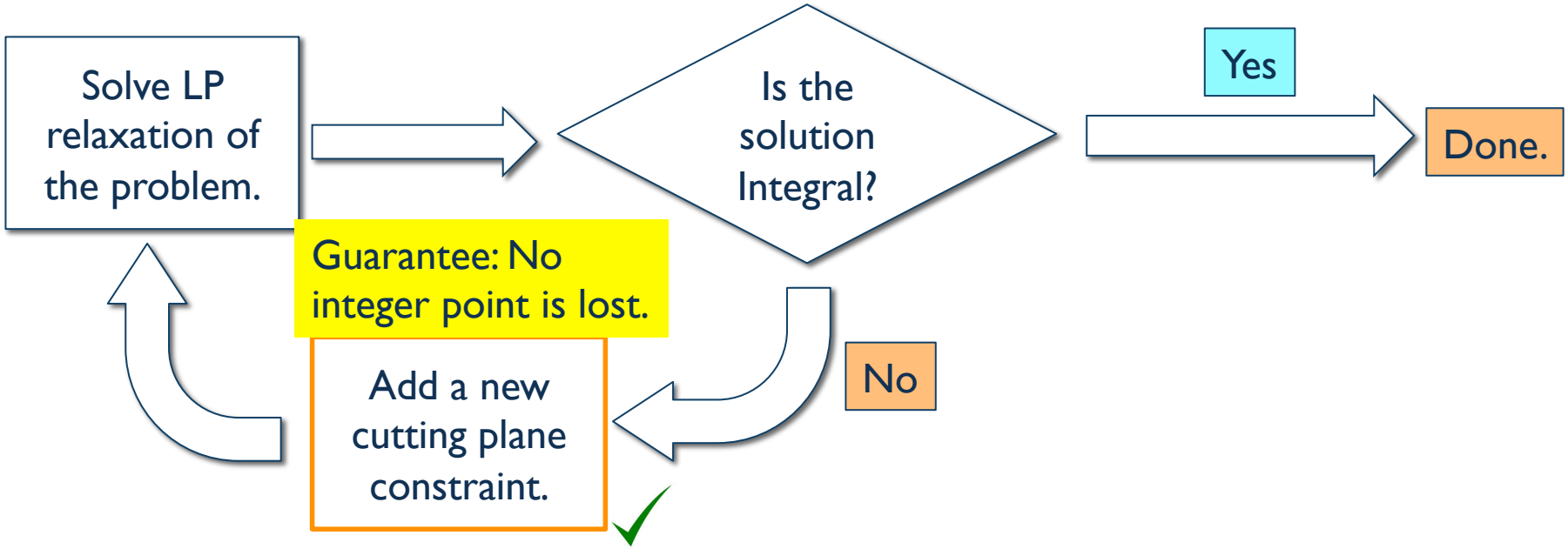
$$0.2x_2 + 0.3x_3 + 0.1x_5 \geq 0.7$$

CUTTING PLANE METHOD

Updating the dictionary

Setup for subsequent iterations.

Overall method



Cutting Plane Example

x_1	1.2	$-3.1x_2$	$+4.3x_3$	$-0.5x_5$
x_4	1	$-x_2$	$+x_3$	$-x_5$
x_6	2.5	$+1.3x_2$	$-2.1x_3$	$+x_5$
z	1.7	$-1.2x_2$	$-2.3x_3$	$-2.1x_5$

Cutting Plane:

$$0.1x_2 + 0.7x_3 + 0.5x_5 \geq 0.2$$

Adding cutting plane to dictionary

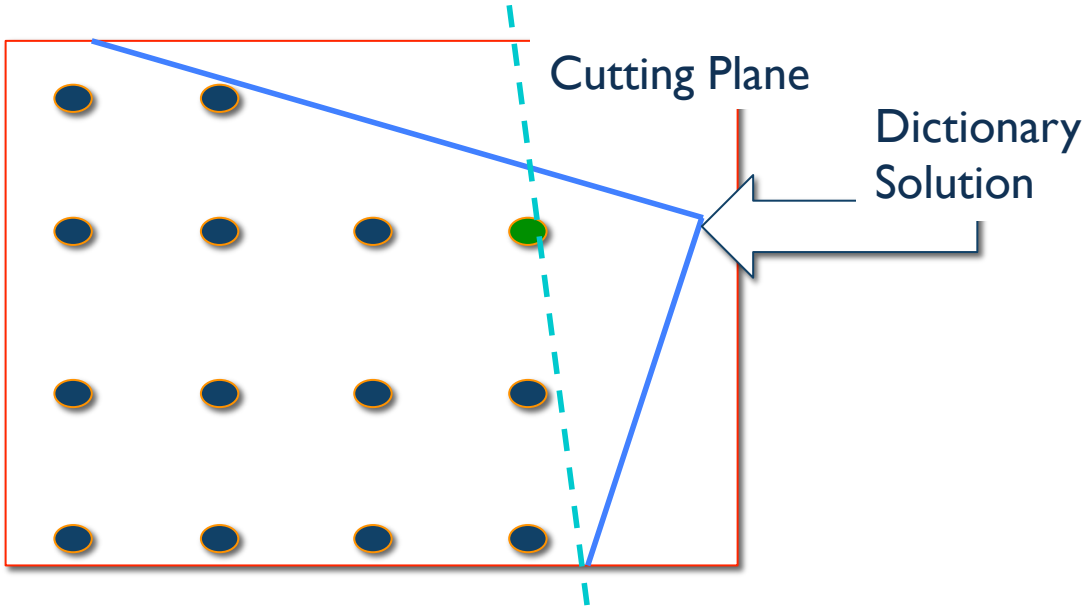
x_1	1.2	$-3.1x_2$	$+4.3x_3$	$-0.5x_5$
x_4	1	$-x_2$	$+x_3$	$-x_5$
x_6	2.5	$+1.3x_2$	$-2.1x_3$	$+x_5$
z	1.7	$-1.2x_2$	$-2.3x_3$	$-2.1x_5$

$$0.1x_2 + 0.7x_3 + 0.5x_5 \geq 0.2$$

x_1	1.2	$-3.1x_2$	$+4.3x_3$	$-0.5x_5$
x_4	1	$-x_2$	$+x_3$	$-x_5$
x_6	2.5	$+1.3x_2$	$-2.1x_3$	$+x_5$
w_1	-0.2	$+0.1x_2$	$+0.7x_3$	$+0.5x_5$
z	1.7	$-1.2x_2$	$-2.3x_3$	$-2.1x_5$

Cutting Plane Method

- Claim: Resulting dictionary after adding cutting plane is primal infeasible.



Cutting Plane: Example

$$
\begin{array}{c|cccc}
x_1 & 1.2 & -3.1x_2 & +4.3x_3 & -0.5x_5 \\
x_4 & 1 & -x_2 & +x_3 & -x_5 \\
x_6 & 2.5 & +1.3x_2 & -2.1x_3 & +x_5 \\
w_1 & -0.2 & +0.1x_2 & +0.7x_3 & +0.5x_5 \\
\hline
z & 1.7 & -1.2x_2 & -2.3x_3 & -2.1x_5
\end{array}
$$

Dual dictionary is feasible but non-final.

Naïve Approach: Re-solve initialization phase Simplex.

Cutting Plane: Solving again after cut.