

SOLVING INTEGER LINEAR PROGRAMS

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

Integer Linear Program: Example

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

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

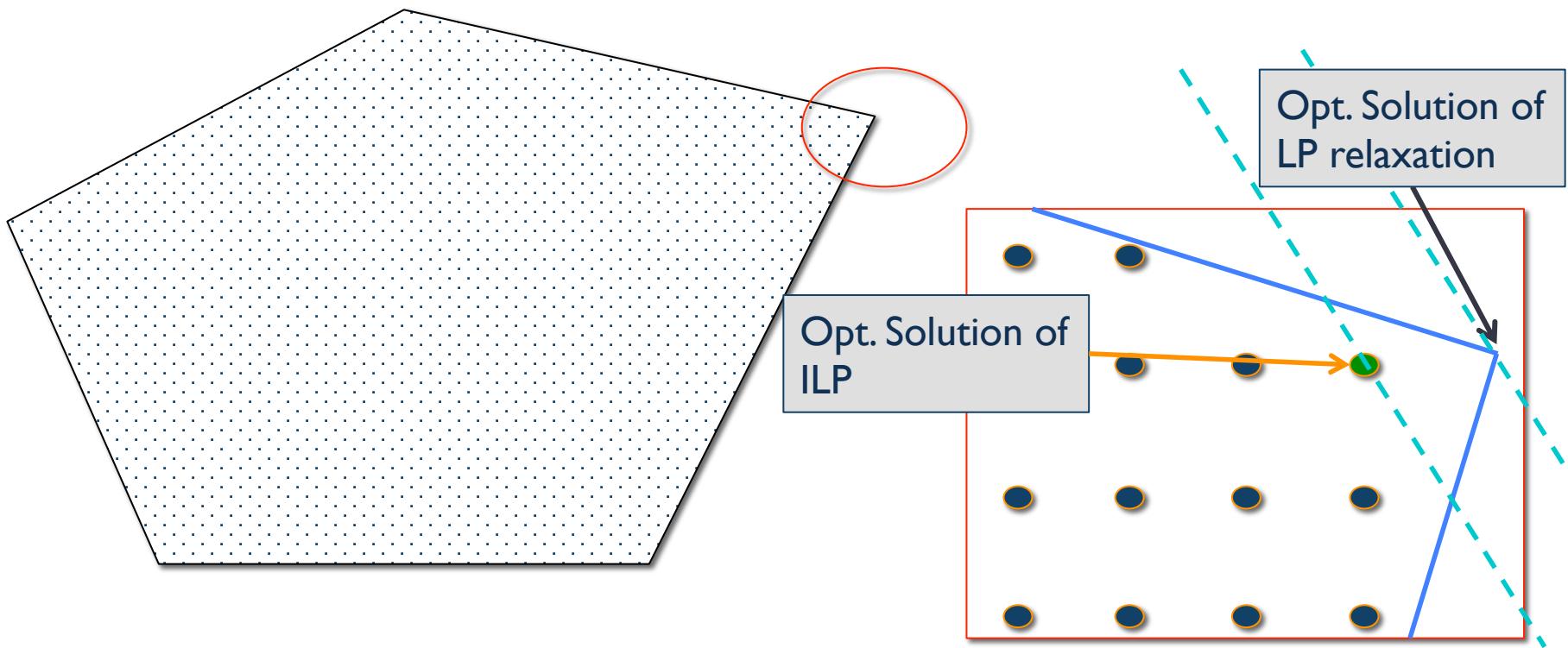
Original ILP

```
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 = 0.3333333333333333  
x2.val = 0.6666666666666667  
x3.val = 2.6666666666666667  
x4.val = 0  
x5.val = 0.3333333333333333  
x6.val = 0  
Optimal Value: -3.333333
```

Case-I: Both LP and ILP are feasible.



Solving ILPs

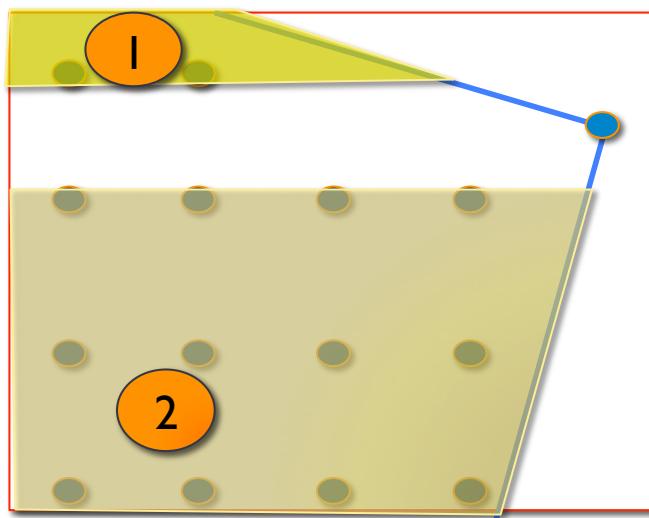
Solve LP relaxation of the ILP.

Case-1: LP relaxation solution satisfies integrality constraint.

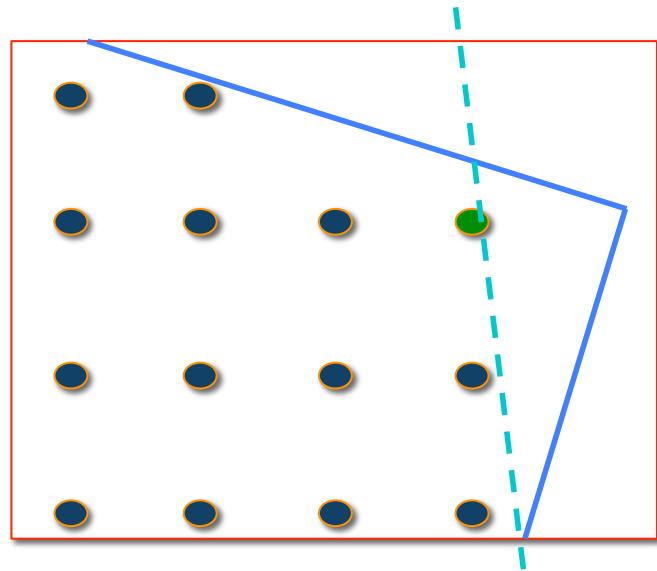
Case-2: LP relaxation solution does not satisfy the integrality constraint.

Dealing with Case-2

Branch and Bound



Cutting Plane Method.



BRANCH-AND-BOUND: BASIC IDEA

Integer Linear Program: Example

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

Step #1: Solve the LP relaxation

Solving the LP relaxation.

```
x1.val = 0.33333333333333  
x2.val = 0.66666666666667  
x3.val = 2.66666666666667  
x4.val = 0  
x5.val = 0.33333333333333  
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 = 2.666666666666667  
x4.val = 0  
x5.val = 0.333333333333333  
x6.val = 0
```

Choose a variable that should be integer

Optimal Value (LP relaxation): -3.33333

Step #3: Branching Constraints

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

Original Problem

b1

b2

Original Problem Constr.

+

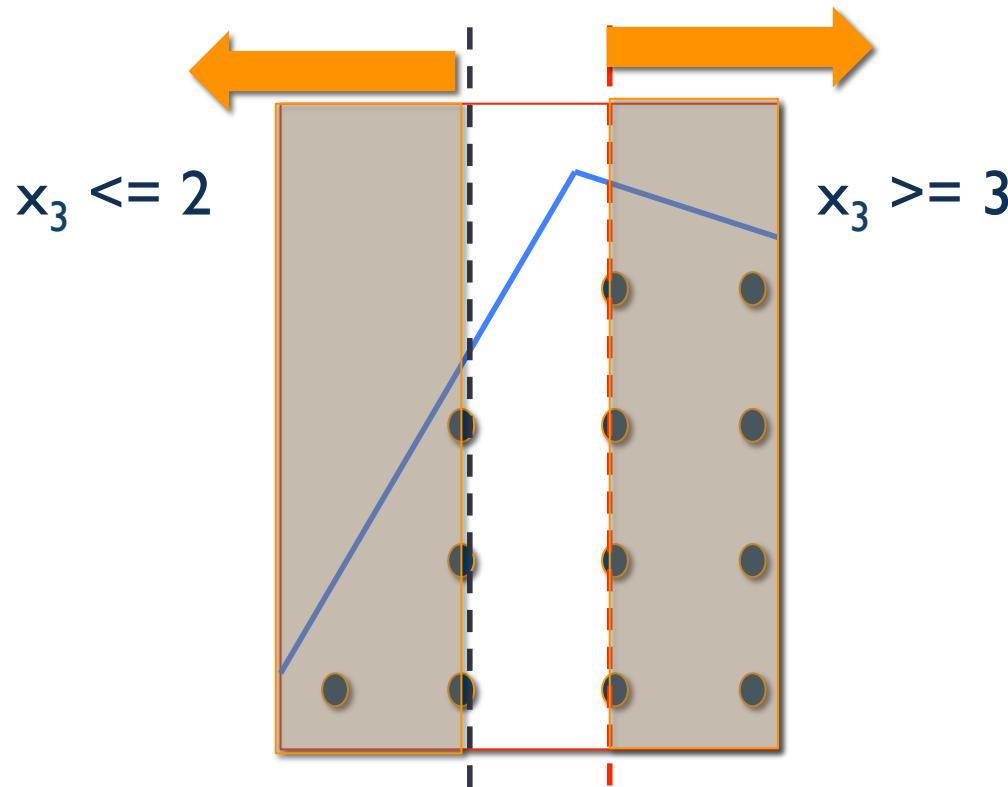
$$(x_3 \leq 2)$$

Original Problem Constr.

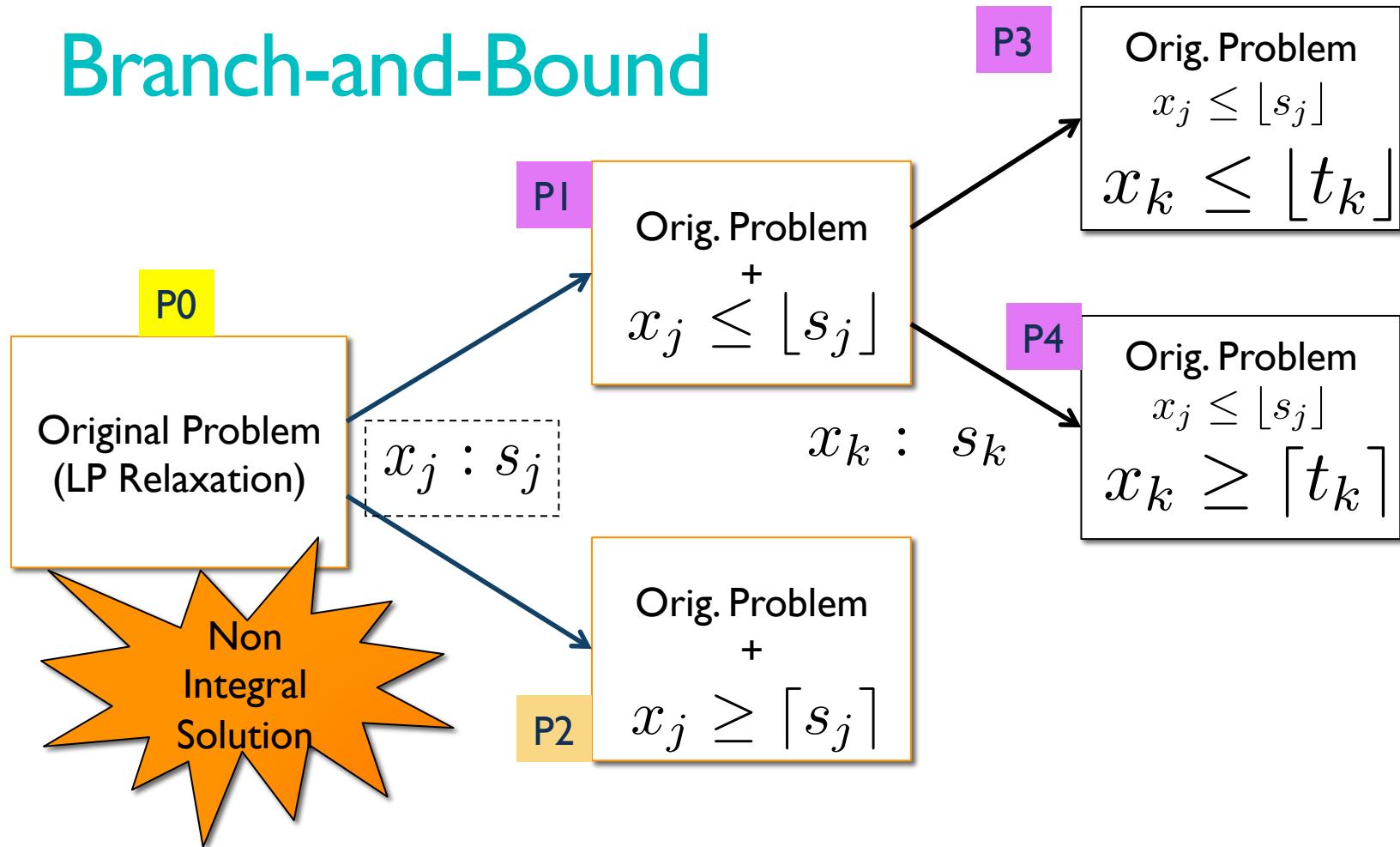
+

$$(x_3 \geq 3)$$

Visualization

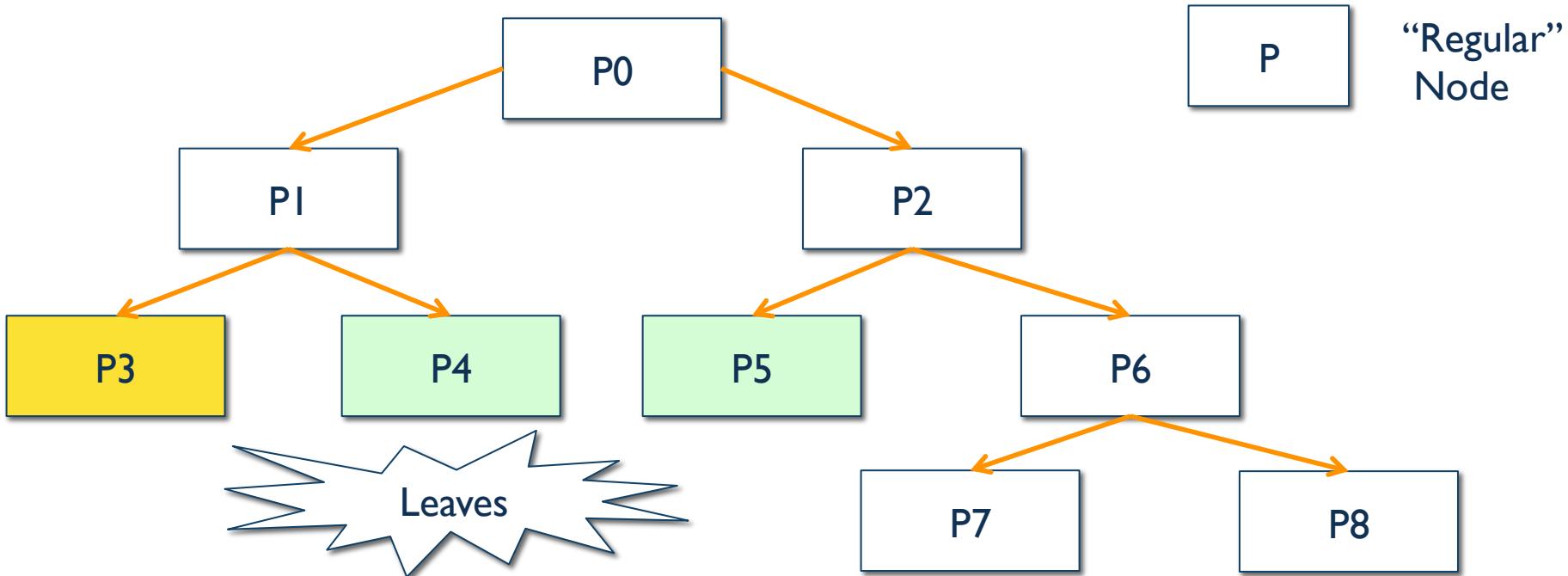


Branch-and-Bound



Branch-And-Bound: Operations

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



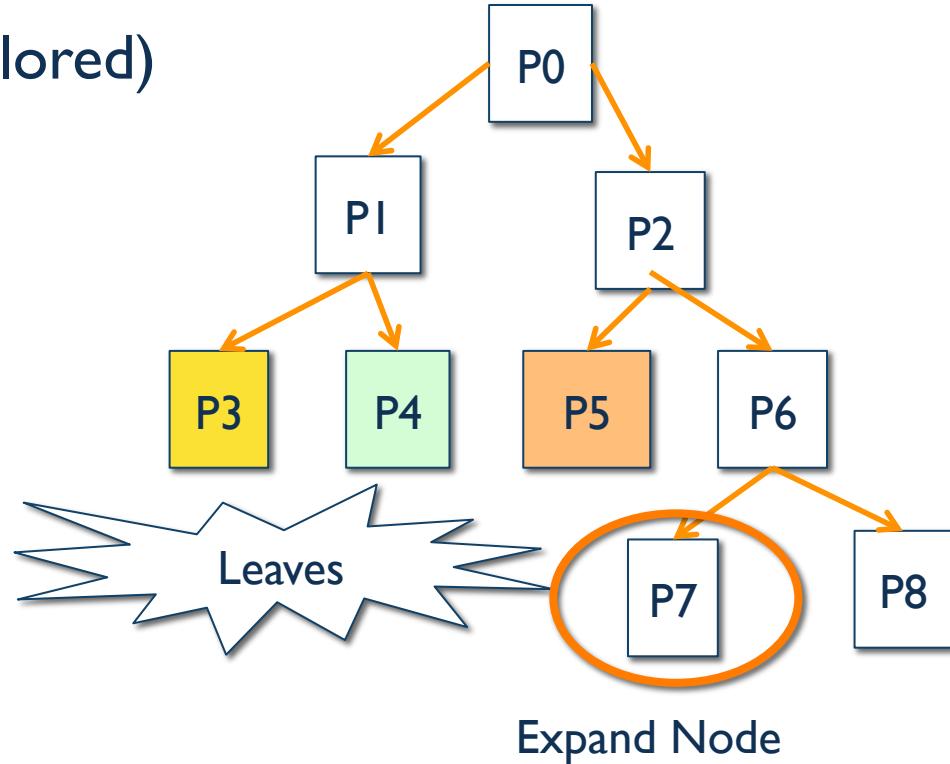
Branch-And-Bound Tree

P0 Regular Node (to be explored)

P3 Leaf node

P4 Leaf node

P5 Leaf node



Expanding a Node

- I. Solve the LP relaxation for the node ILP.

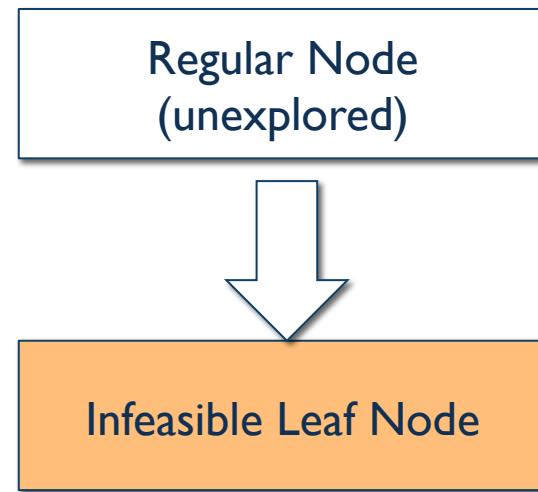
Regular Node
(unexplored)

LP relaxation solution

2. Three cases:
 - I. Infeasible.
 2. Integral Solution.
 3. Fractional Solution.

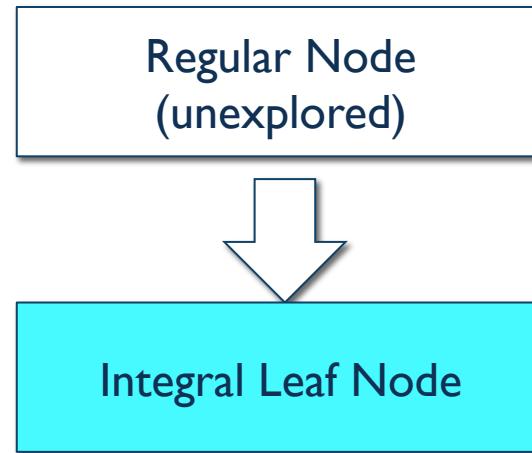
Case-I: Infeasible LP relaxation.

- LP relaxation is infeasible.



Case-2: LP relaxation yields integral solution

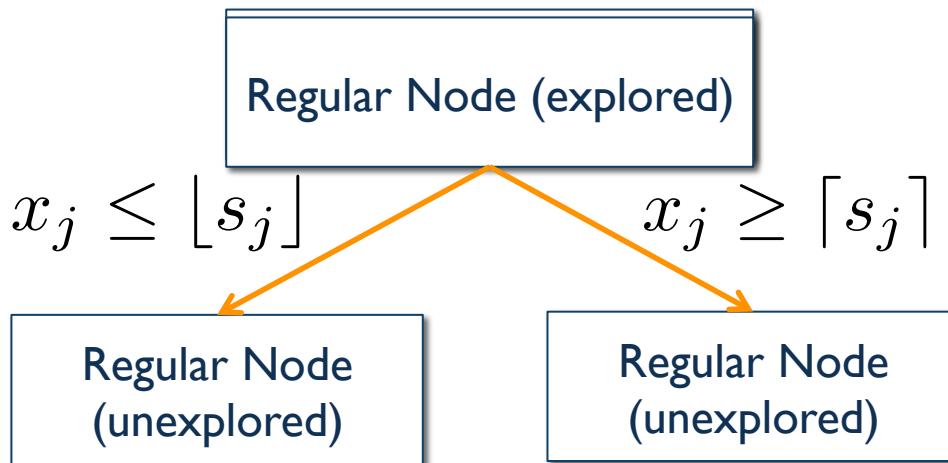
- LP relaxation solution is integral: ILP solution = LP solution.



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)

$\text{optSolution} \leq \text{bestObjective?}$

LP relaxation solution:

x1: ...

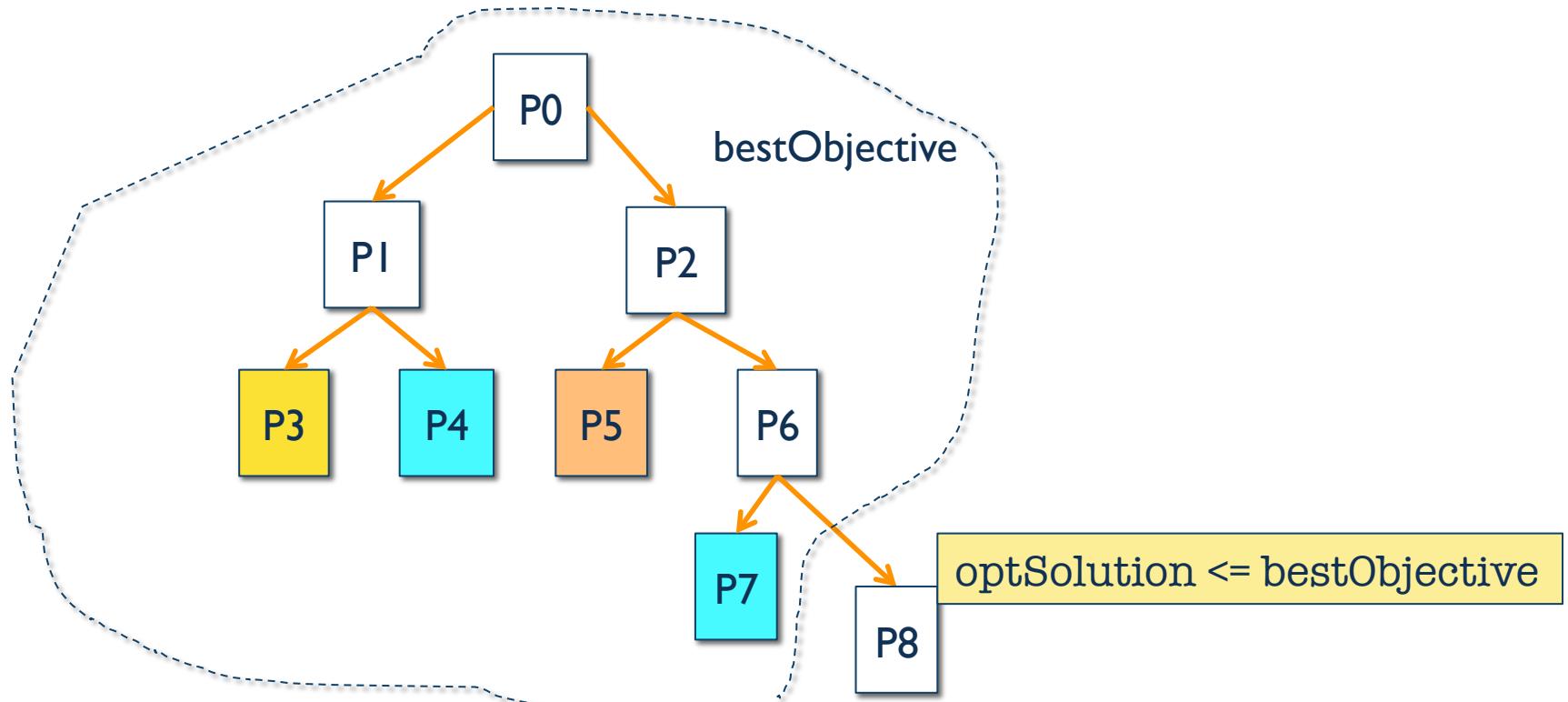
x2: ...

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

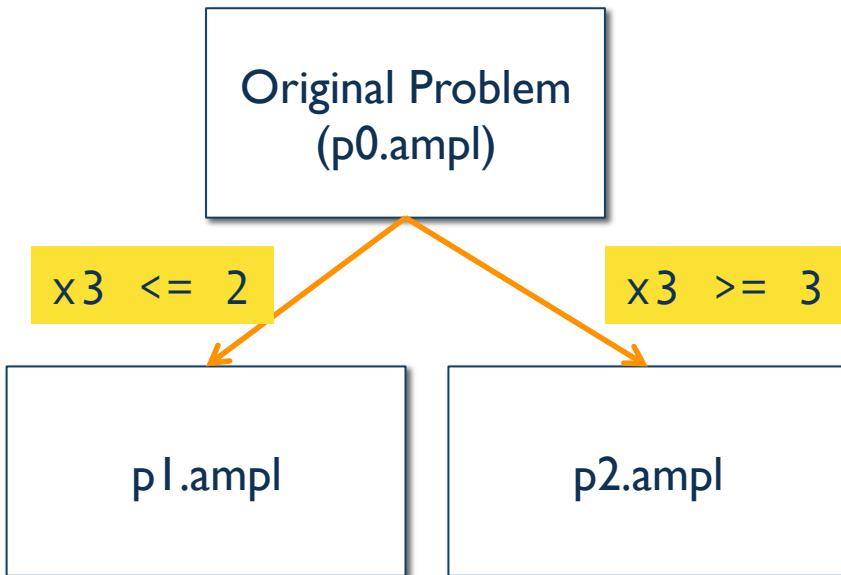
x1.val = 0
x2.val = 1
x3.val = 4
x4.val = 1
x5.val = 0
x6.val = 0
Optimal Value: -4.200000

LP relaxation

x1.val = 0.3333333333333333
x2.val = 0.6666666666666667
x3.val = 2.666666666666667
x4.val = 0
x5.val = 0.3333333333333333
x6.val = 0
Optimal Value: -3.333333

Initial Node

```
bestObjective := -Infinity
```



x1.val = 0.3333333333333333

x2.val = 0.6666666666666667

x3.val = 2.666666666666667

x4.val = 0

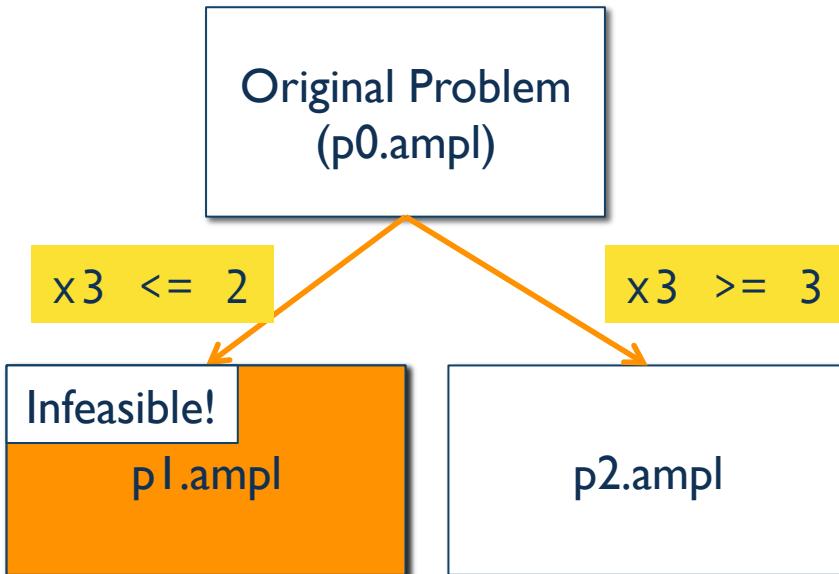
x5.val = 0.3333333333333333

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

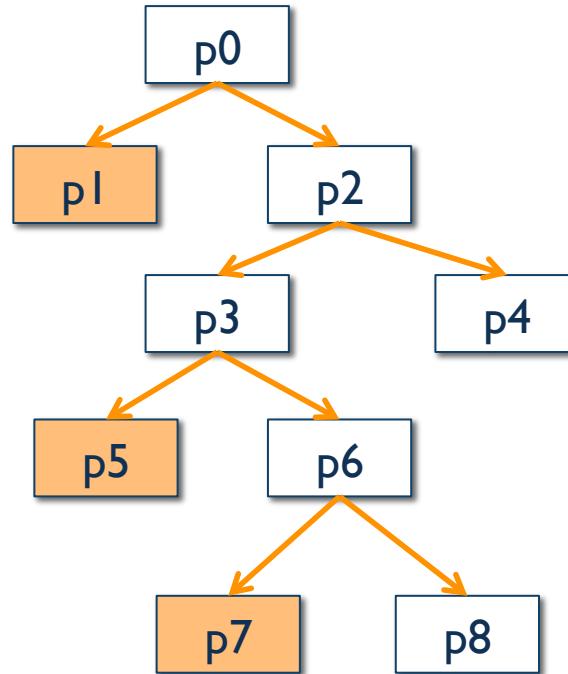
x3.val = 4.57142857142857

x4.val = 0

x5.val = 0.714285714285714

x6.val = 0.285714285714286

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 = 0.333333333333333
x5.val = 0.666666666666667
x6.val = 0.333333333333333
Optimal Value: -4.066667

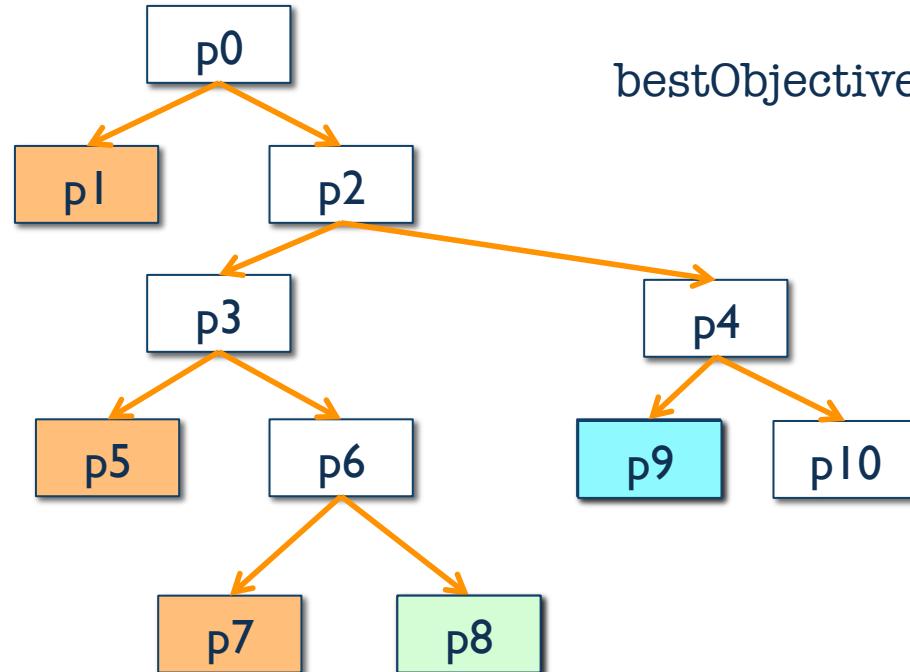
BnB Tree

p9.ampl
x1.val = 0
x2.val = 1
x3.val = 4
x4.val = 1
x5.val = 0
x6.val = 0

Optimal Value: -4.2

p8.ampl
x1.val = 1
x2.val = 0
x3.val = 6
x4.val = 0
x5.val = 1
x6.val = 0.5

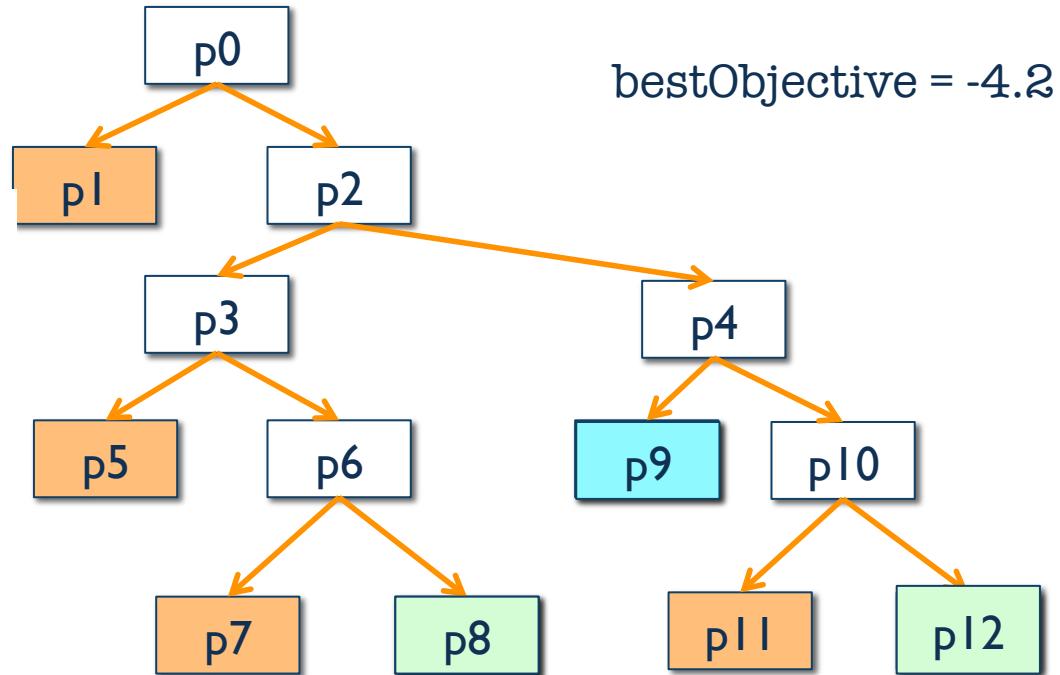
Optimal Value: -4.750000



bestObjective = -4.2

BnB Tree

```
p12AMPL  
x1.val = 0  
x2.val = 1  
x3.val = 6  
x4.val = 0  
x5.val = 1  
x6.val = 0.5  
Optimal Value: -5.750000
```



BnB Final Tree

x1.val = 0

x2.val = 1

x3.val = 4

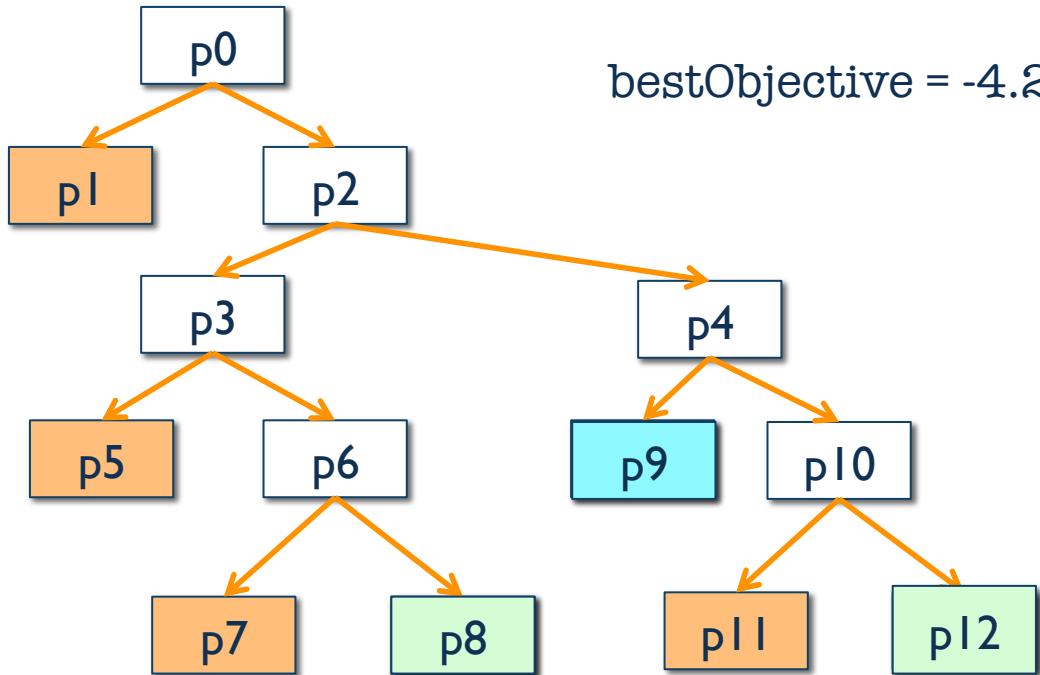
x4.val = 1

x5.val = 0

x6.val = 0

Optimal Value: -4.200000

bestObjective = -4.2



BRANCH-AND-BOUND METHOD

Heuristics

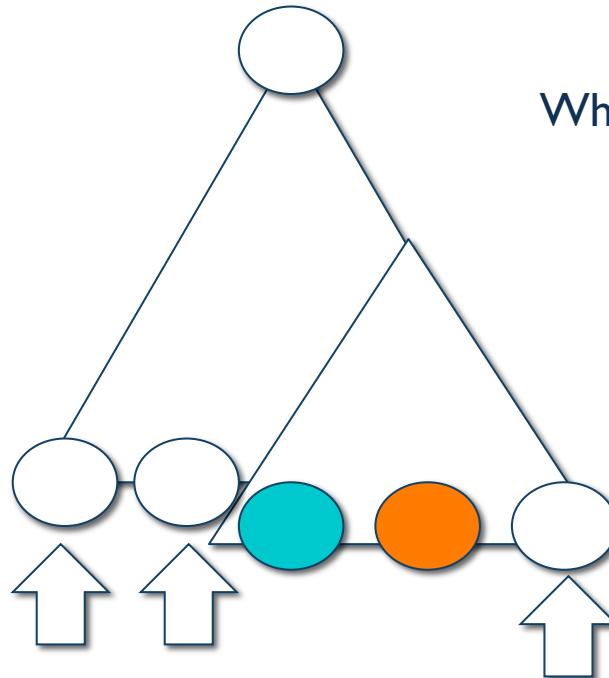
BnB choices to be made

- 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

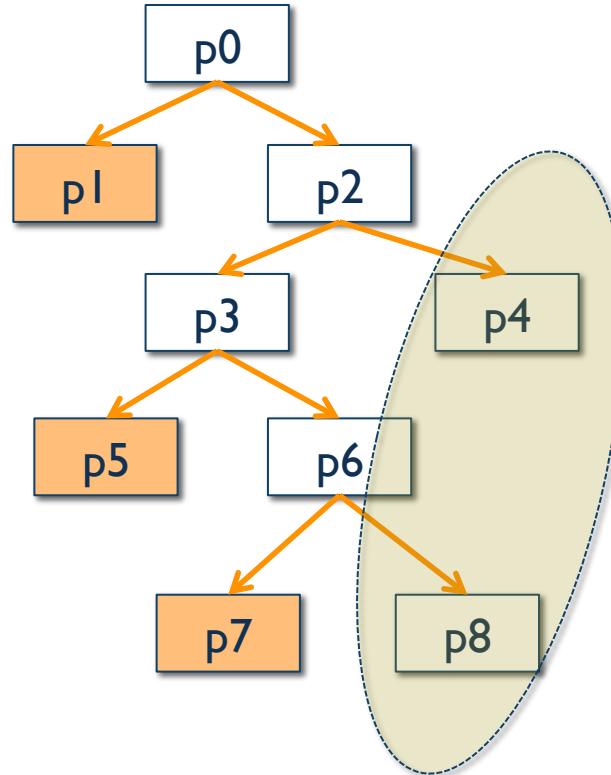


Which node should I examine first?

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

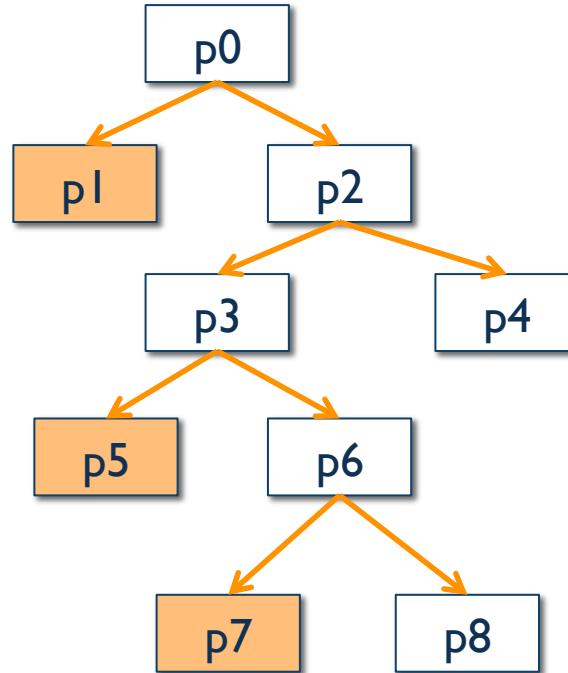
x3.val = 4.57142857142857

x4.val = 0

x5.val = 0.714285714285714

x6.val = 0.285714285714286

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 = 0.333333333333333
x5.val = 0.666666666666667
x6.val = 0.333333333333333
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

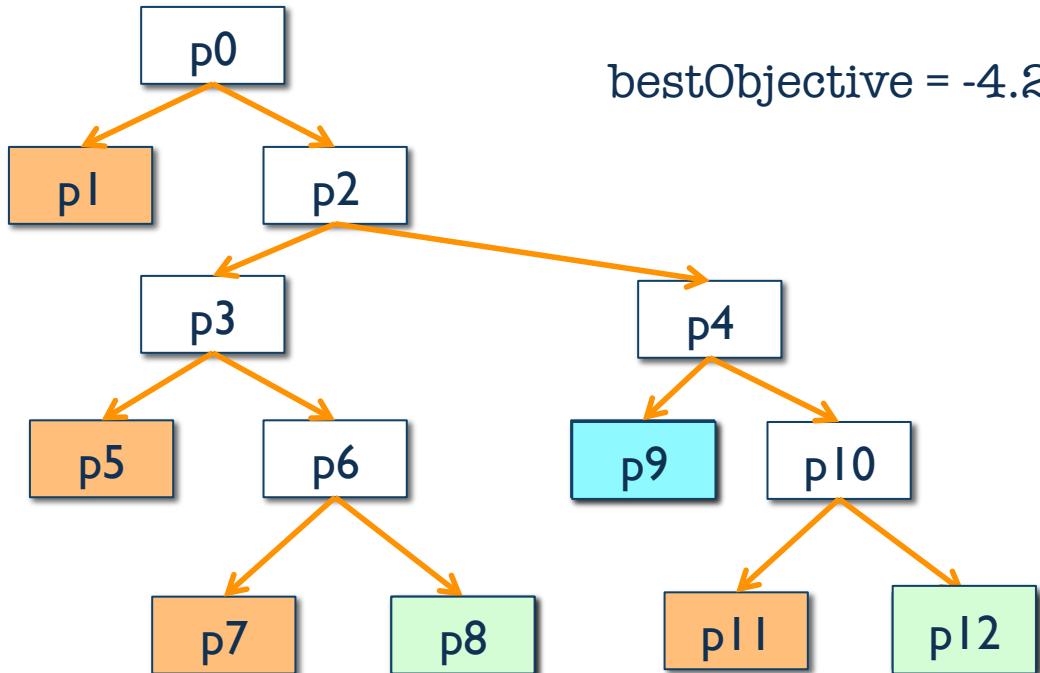
x4.val = 1

x5.val = 0

x6.val = 0

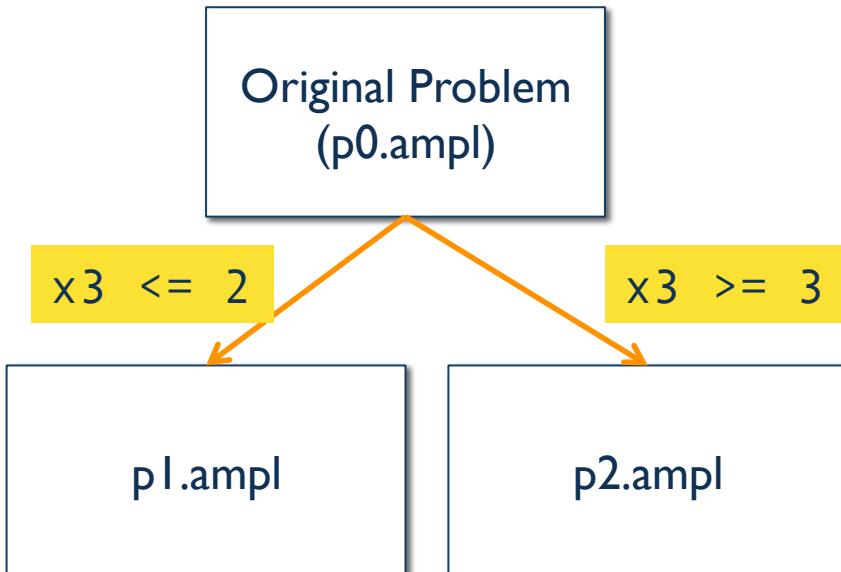
Optimal Value: -4.200000

bestObjective = -4.2



Initial Node

```
bestObjective := -Infinity
```



x1.val = 0.3333333333333333

x2.val = 0.6666666666666667

x3.val = 2.666666666666667

x4.val = 0

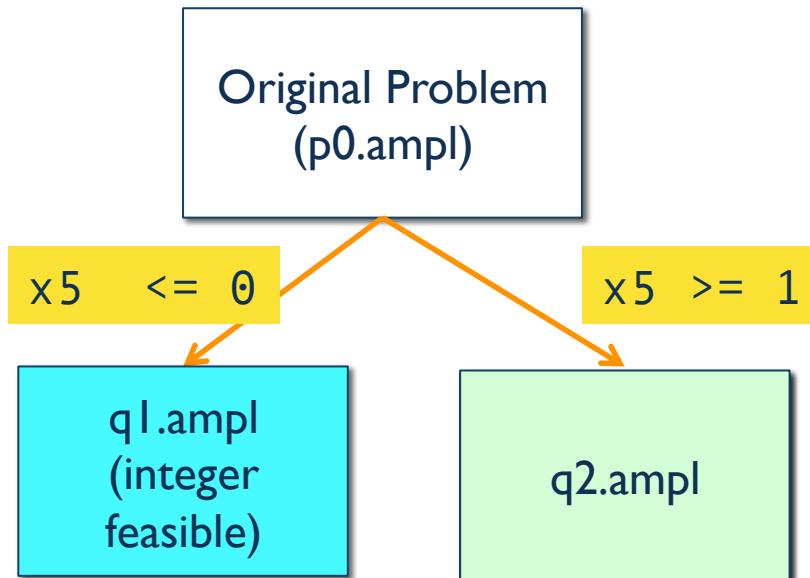
x5.val = 0.3333333333333333

x6.val = 0

Optimal Value: -3.333333

What if we chose a different branch variable..?

bestObjective := - Infinity



bestObjective := -4.2

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

How to select a branch variable?

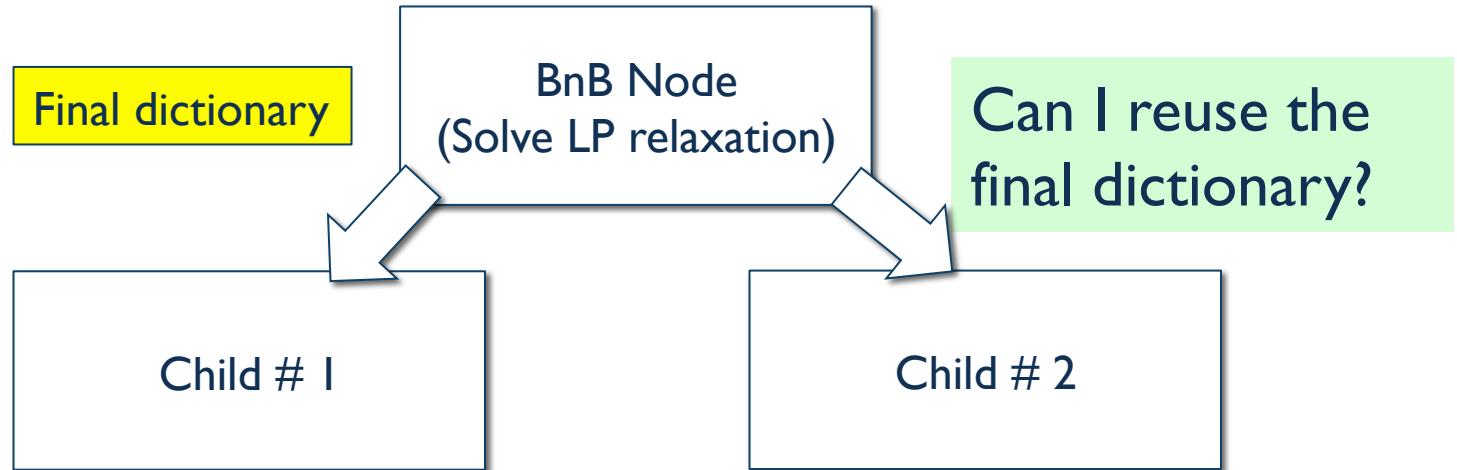
- The choice of a branch variable is very important.
- There is no single “best heuristic”.
- Often, some expertise with the problem domain informs the heuristic.

BRANCH-AND-BOUND

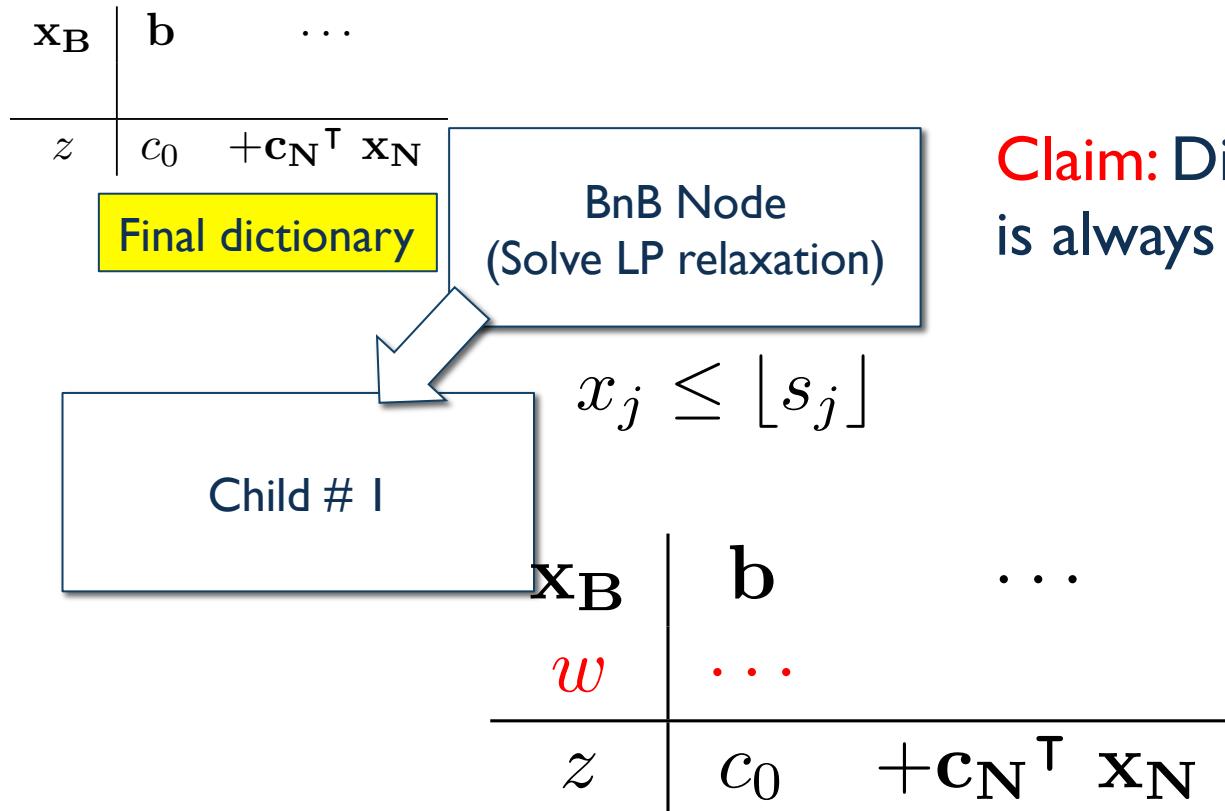
At the dictionary level

Thus far

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

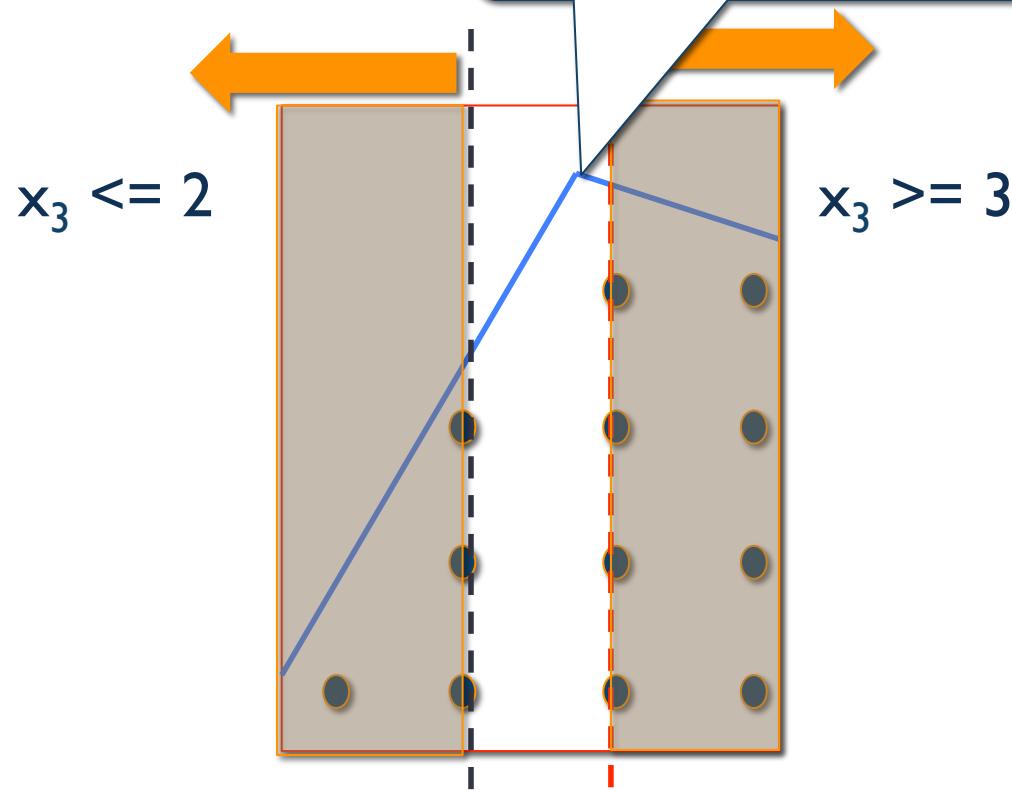


View from the dictionary



Proof (Pictorial)

Solution represented by
dictionary D and D'



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;
```

$$x_4 : 10 - x_1$$

$$x_5 : 10 - x_2$$

$$x_6 : 10 - x_3$$

$$x_7 : 1 + 2x_1 - 7x_2$$

$$x_8 : 3 - x_1 + 2x_2 - 5x_3$$

$$x_9 : 7 - x_1 - x_2 + 3x_3$$

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	$+0x_8$	$-x_9$	$-2x_3$

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

Dictionary After Branch

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

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	$+0.333333x_8 - 0.333333x_9 + 2.666667x_3$		
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?

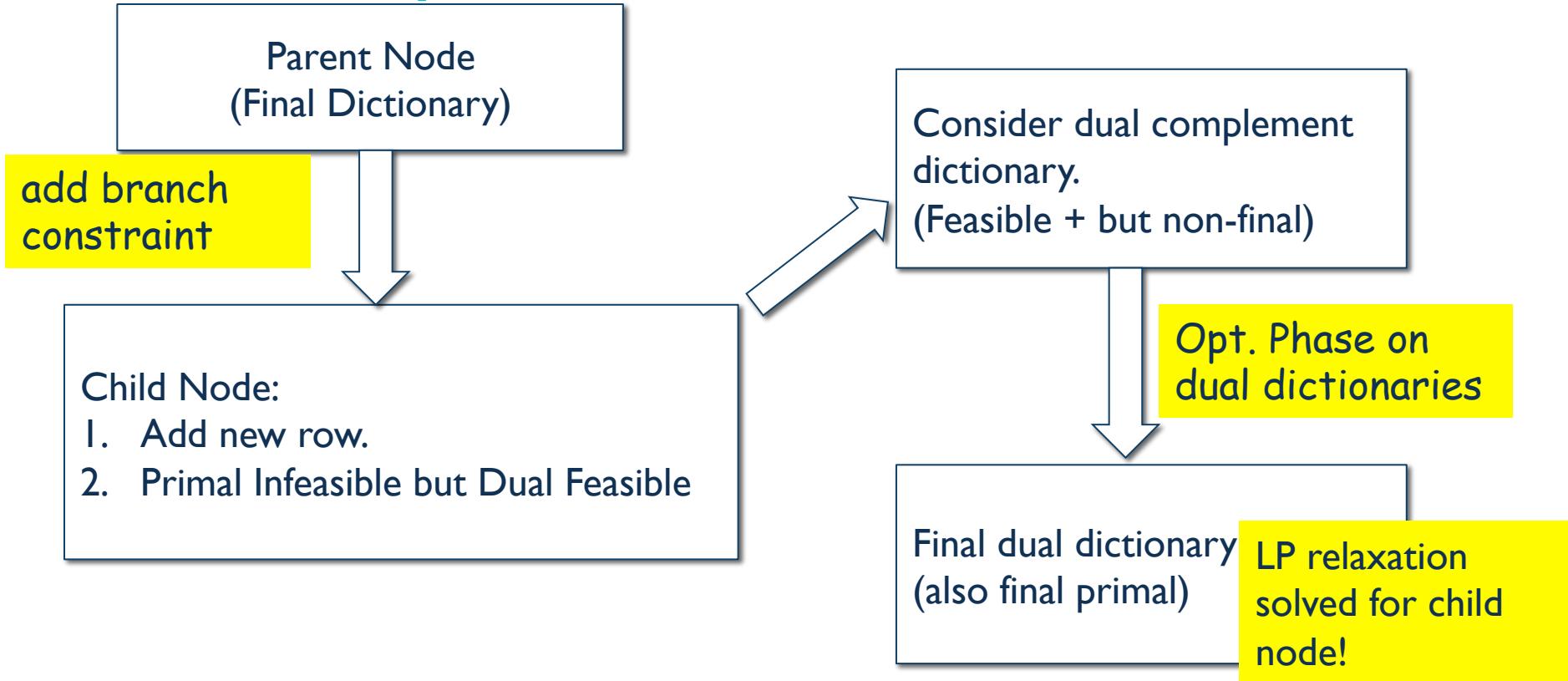
Dictionary After Branch

x_4	4.33333333333	$+0.333333x_8 + 0.666667x_9 - 0.333333x_3$		
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	$+0.333333x_8 - 0.333333x_9 + 2.666667x_3$		
x_{10}	-0.33333	$-0.333333x_8 - 0.666667x_9 + 0.333333x_3$		
z	7.0	$+0x_8$	$-x_9$	$-2x_3$

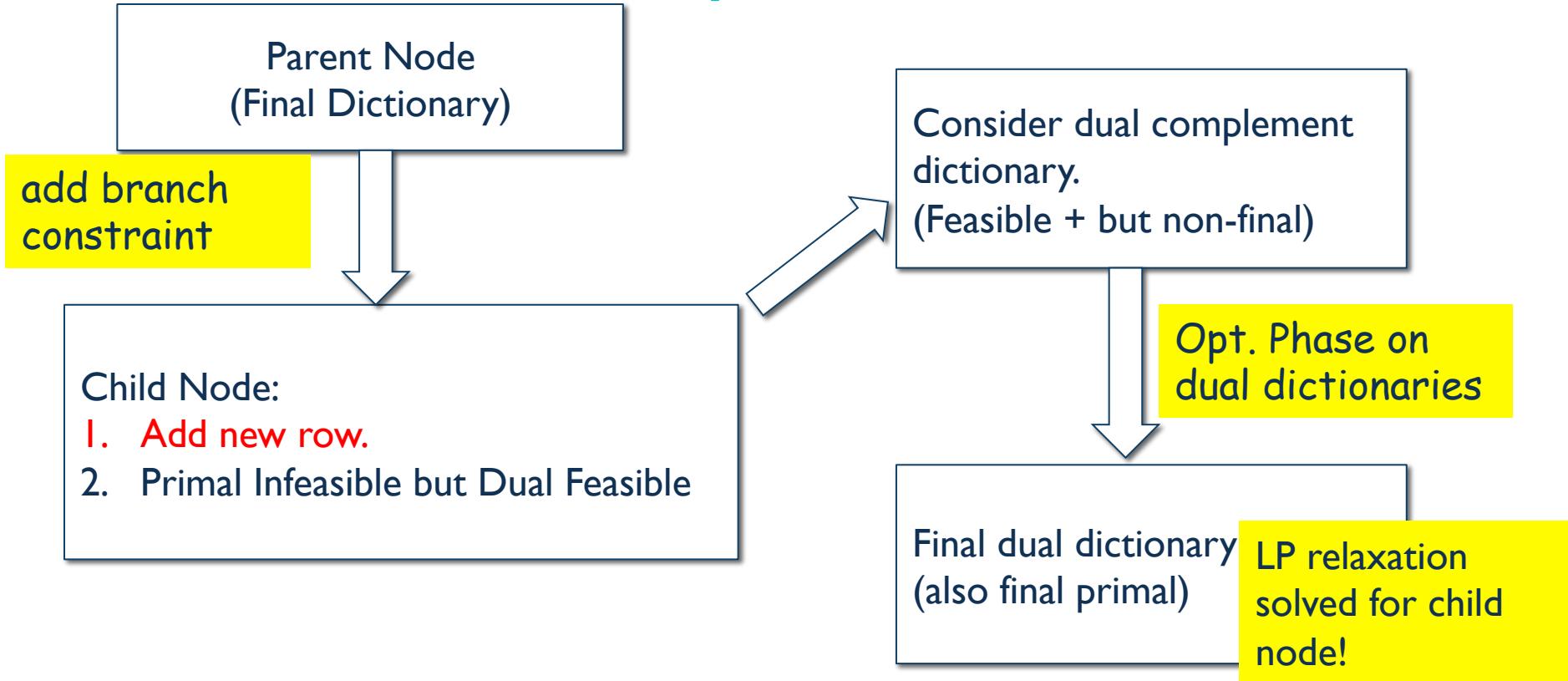
Dictionary becomes primal infeasible.

Dual Feasible Dictionary!!
But not dual final.

How to update after branch?



General Form Simplex



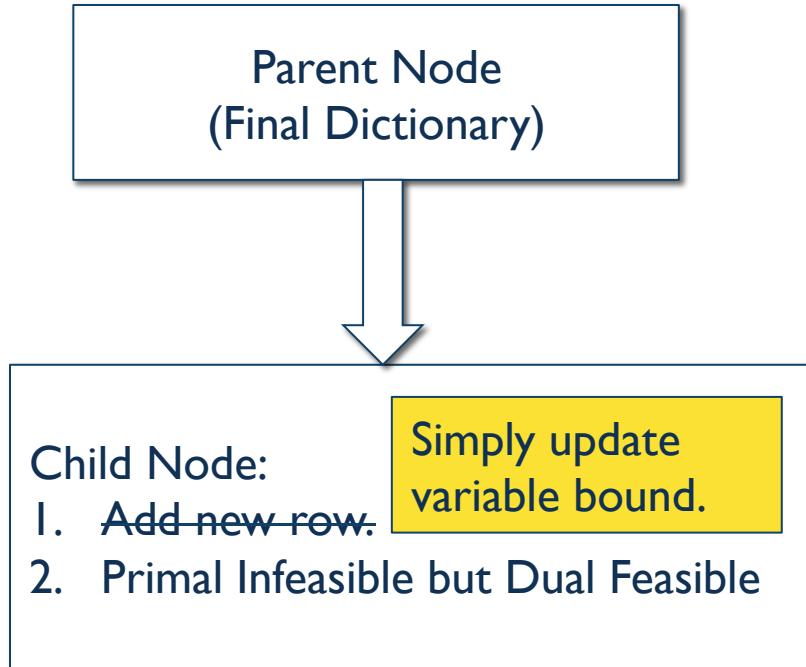
General Form Dictionary

- Give special treatment to bounds on variables.

$$l \leq x \leq u$$

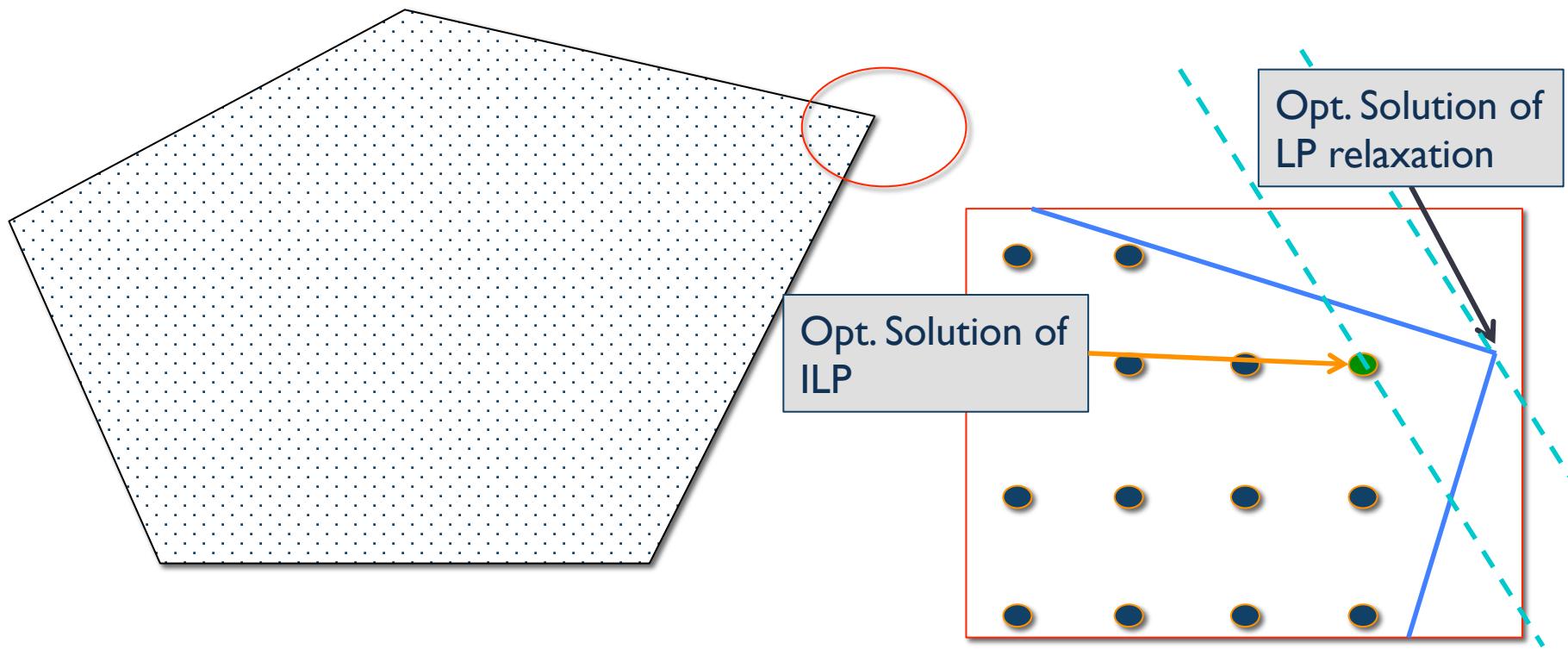
- Modify the Simplex algorithm in two ways:
 - Dictionaries now have special ways to track bounds.
 - Pivoting is modified.

Branch-and-Bound with General Form



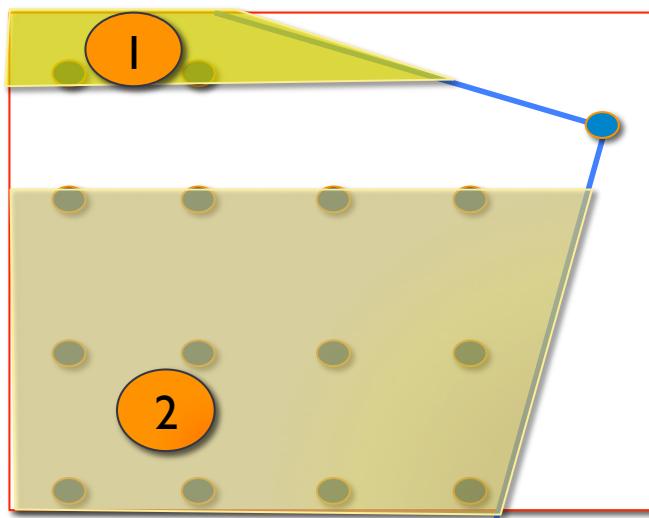
CUTTING PLANE METHOD

LP Relaxation for ILP

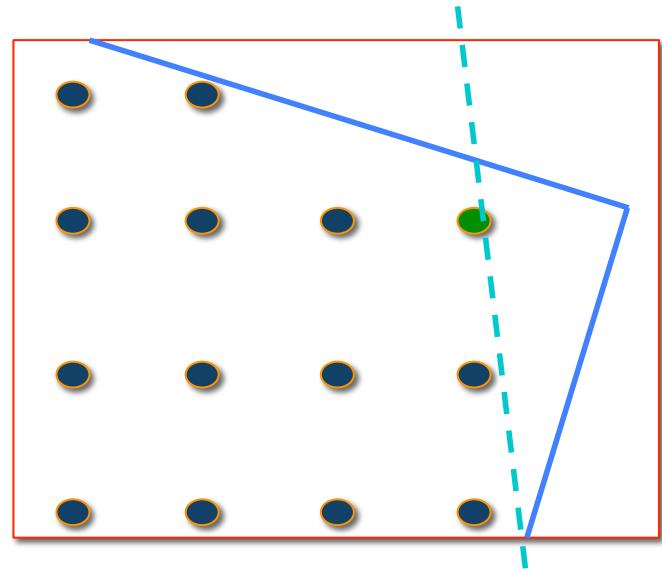


Removing a Fractional Solution

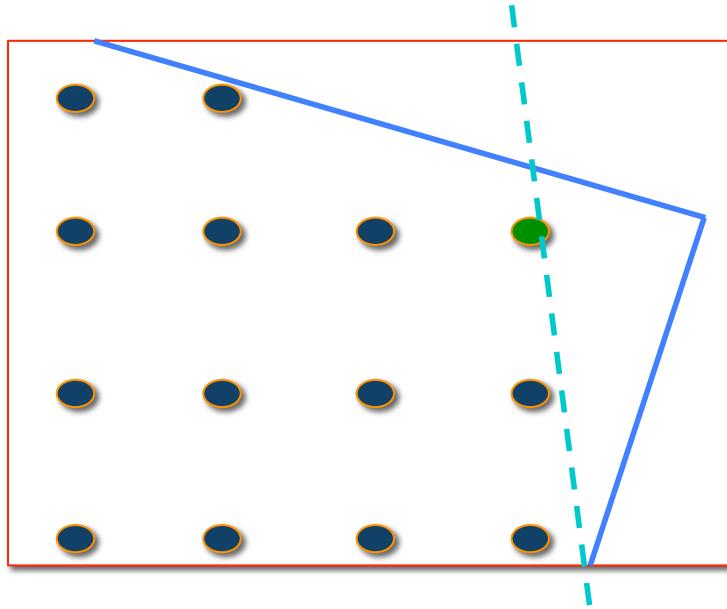
Branch and Bound



Cutting Plane Method.



Cutting Plane



How do we find a cutting plane?

GOMORY-CHVATAL CUTTING PLANE

Part I: Setting up the problem.

ILP in Standard Form

$$\max \quad c^T \ x$$

$$\text{s.t.} \quad Ax \leq b$$

$$x \geq 0$$

$$x \in \mathbb{Z}$$

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

Conversion to standard form

- 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 \geq to \leq by negating both sides.
- How do we make sure A, b, c are integers?

Reasonable assumption: original problem coefficients are rationals.

Integer Coefficients.

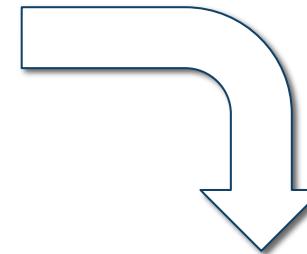
$$\begin{array}{lllll} \max & 2x_1 & +0.3x_2 & -0.1x_3 & \\ \text{s.t.} & 0.1x_1 & -2x_2 & -x_3 & \leq 0.25 \\ & 0.5x_1 & -2.6x_2 & +1.3x_3 & \leq 0.15 \\ & x_1, & x_2, & x_3 & \geq 0 \\ & x_1, & x_2, & x_3 & \in \mathbb{Z} \end{array}$$

Scale
constraints
+
Objective

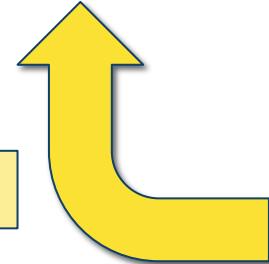
$$\begin{array}{lllll} \max & 2x_1 & +0.3x_2 & -0.1x_3 & \times 10 \\ \text{s.t.} & 0.1x_1 & -2x_2 & -x_3 & \leq 0.25 \times 20 \\ & 0.5x_1 & -2.6x_2 & +1.3x_3 & \leq 0.15 \times 100 \\ & x_1, & x_2, & x_3 & \geq 0 \\ & x_1, & x_2, & x_4 & \in \mathbb{Z} \end{array}$$

Conversion to Integer Coefficients.

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



Divide result by 10



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

Adding Slack Variables

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

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

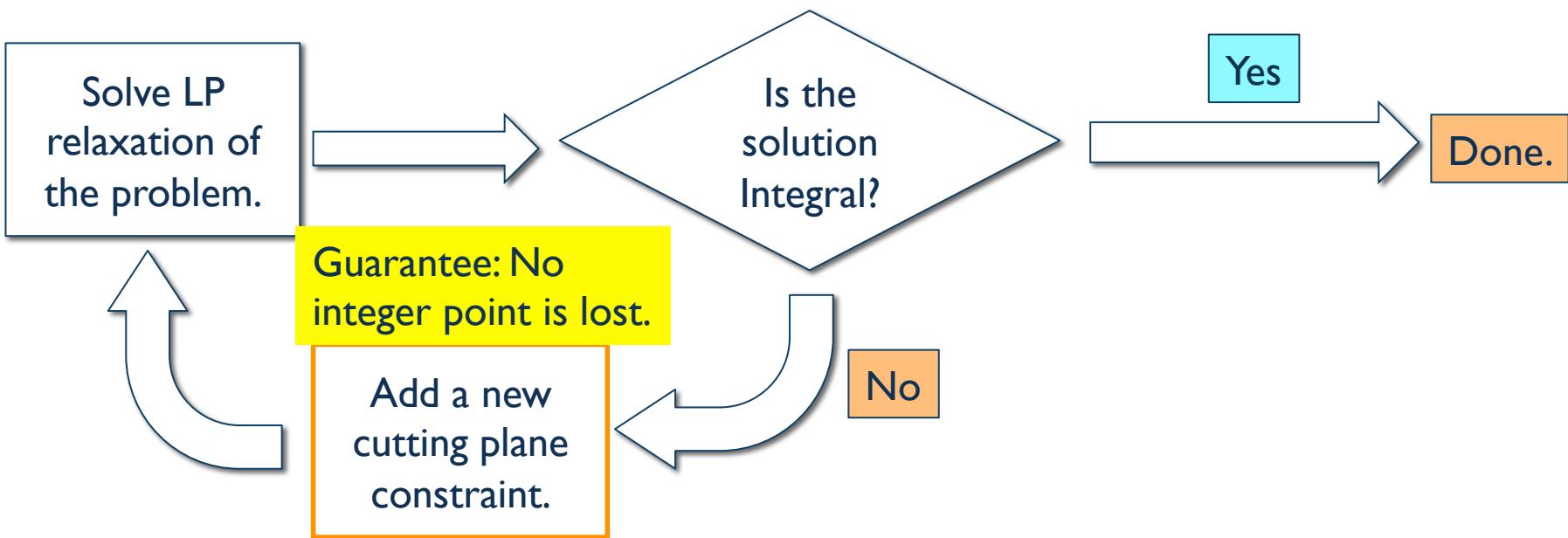
Summary

- 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.

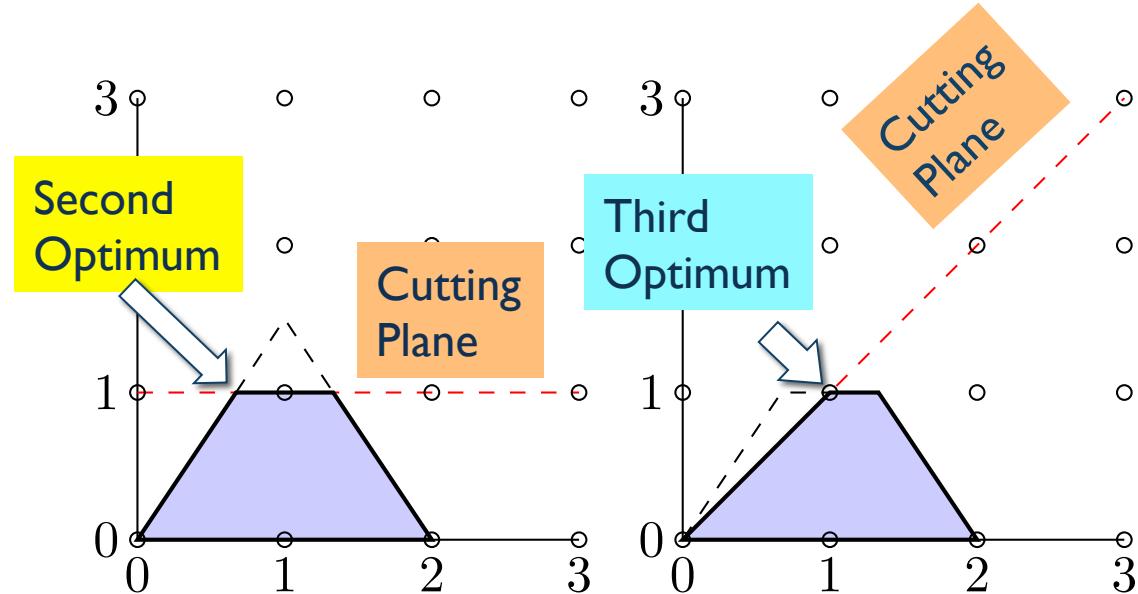
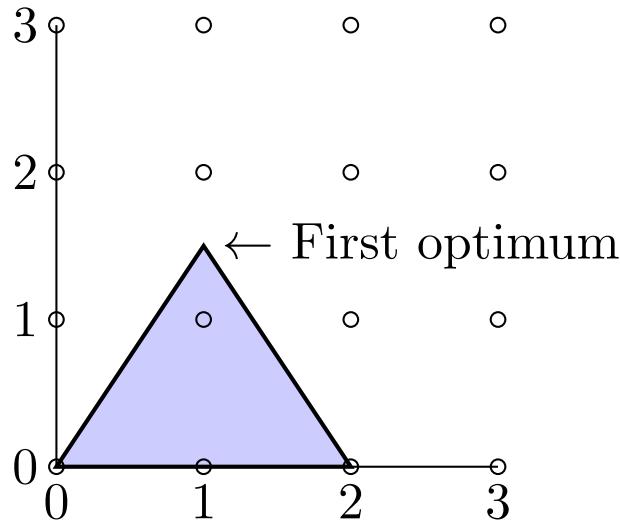
CUTTING PLANE METHOD

Part 2: Gomory-Chvatal Cuts

Overall method



Cutting Plane Visualization



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

Cutting Plane Method

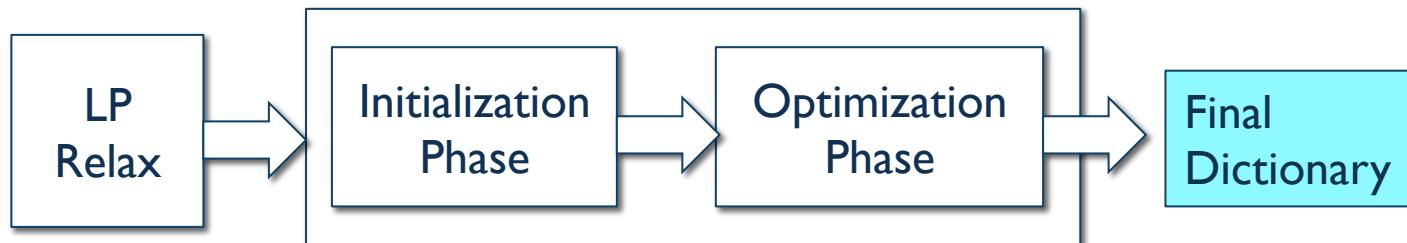
- Deriving the Cutting Plane
 - Gomory-Chvatal Cuts
- Integrating the method inside Simplex.
 - Using the dual dictionary to avoid initialization phase Simplex.

GOMORY-CHVATAL CUTS

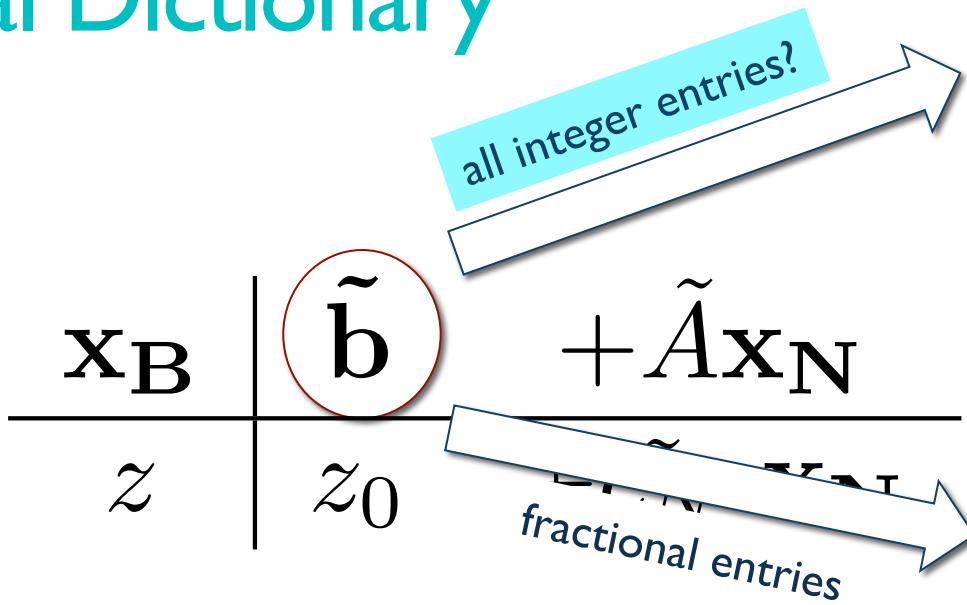
Overall Idea

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

I. Solve the LP relaxation using Simplex algorithm.



Final Dictionary



Optimal solution to
ILP found!!

Derive a cutting
plane

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 I

Step I

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
 $B \geq 0$

$$A + B = 1.2$$

Claim

($x_1 + 0.7x_3 + 0.5x_5$)

Conclusion:

- A I. Fractional part of B is 0.2
- B 2. $B \geq 0.2$

In other words,

- I. $B - 0.2$ is an integer.
- 2. $B - 0.2 \geq 0$

$$x_2 + 0.7x_3 + 0.5x_5) = 1 + 0.2$$

$\underbrace{}_B$

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)

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

$b_k \notin \mathbb{Z}$

Cutting Plane

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

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

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

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

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

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:

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

Therefore, we get

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

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

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

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

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

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

x_{B1}	=	b_1	$+ a_{11}x_{I1}$	$+ \cdots$	$+ a_{1j}x_{Ij}$	$+ \cdots$	$+ a_{1n}x_{In}$	
⋮								$b_k \notin \mathbb{Z}$
x_{Bk}	=	b_k	$+ a_{k1}x_{I1}$	$+ \cdots$	$+ a_{kj}x_{Ij}$	$+ \cdots$	$+ a_{kn}x_{In}$	
⋮								
x_{Bm}	=	b_m	$+ a_{m1}x_{I1}$	$+ \cdots$	$+ a_{mj}x_{Ij}$	$+ \cdots$	$+ a_{mn}x_{In}$	
z	=	c_0	$+ c_1x_{I1}$	$+ \cdots$	$+ c_jx_{Ij}$	$+ \cdots$	$+ c_nx_{In}$	

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	$-1.2x_2$	$-2.3x_3$	$-2.1x_5$

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

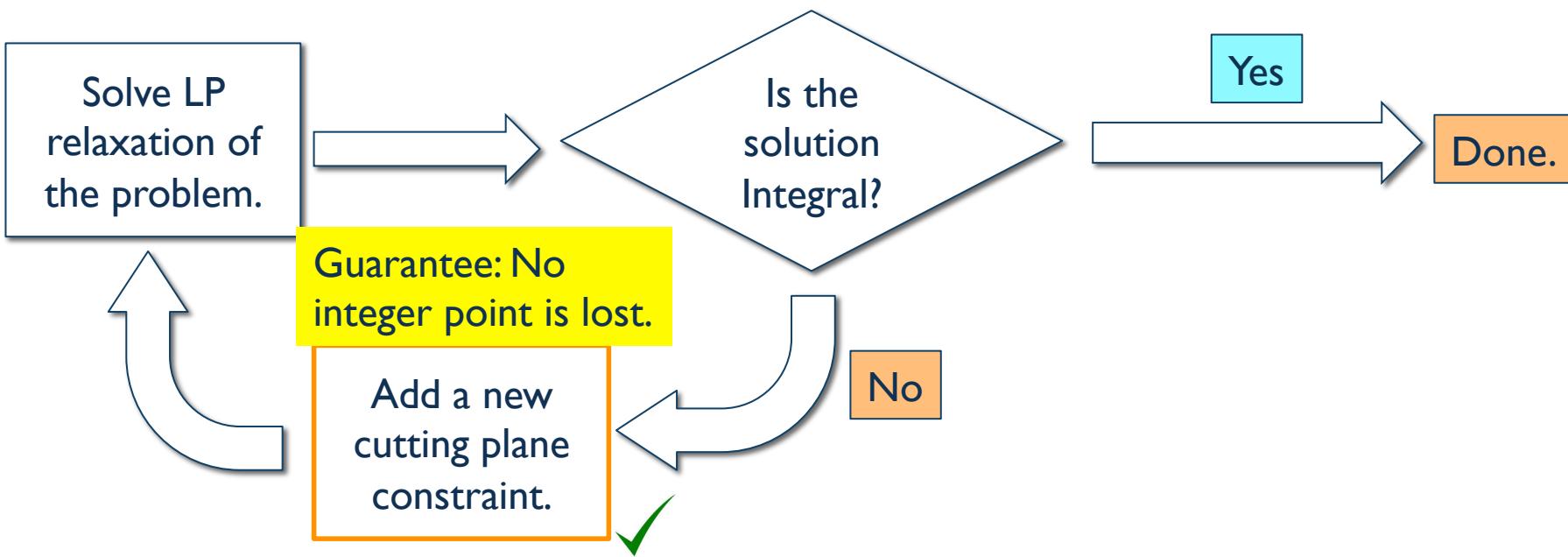
$$0.2x_2 + 0.3x_3 + 0.1x_5 \geq 0.7$$

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

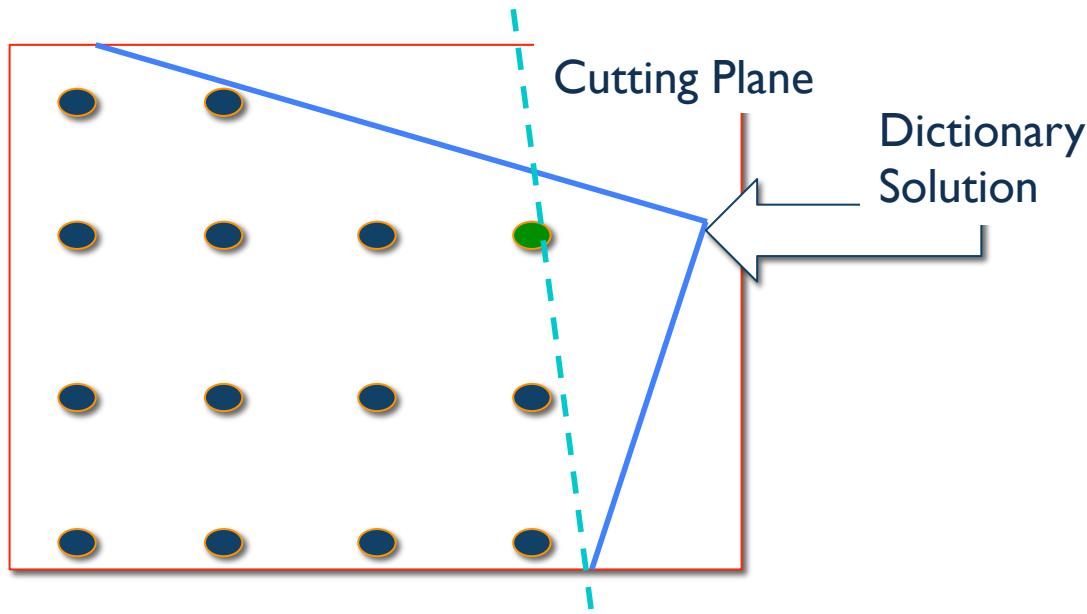
x_1	1.2	$-3.1x_2$	$+4.3x_3$	$-0.5x_5$
x_4	1	$-x_2$	$+x_3$	$-x_5$
x_6	2.5	$+1.3x_2$	$-2.1x_3$	$+x_5$
z	1.7	$-1.2x_2$	$-2.3x_3$	$-2.1x_5$

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

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.



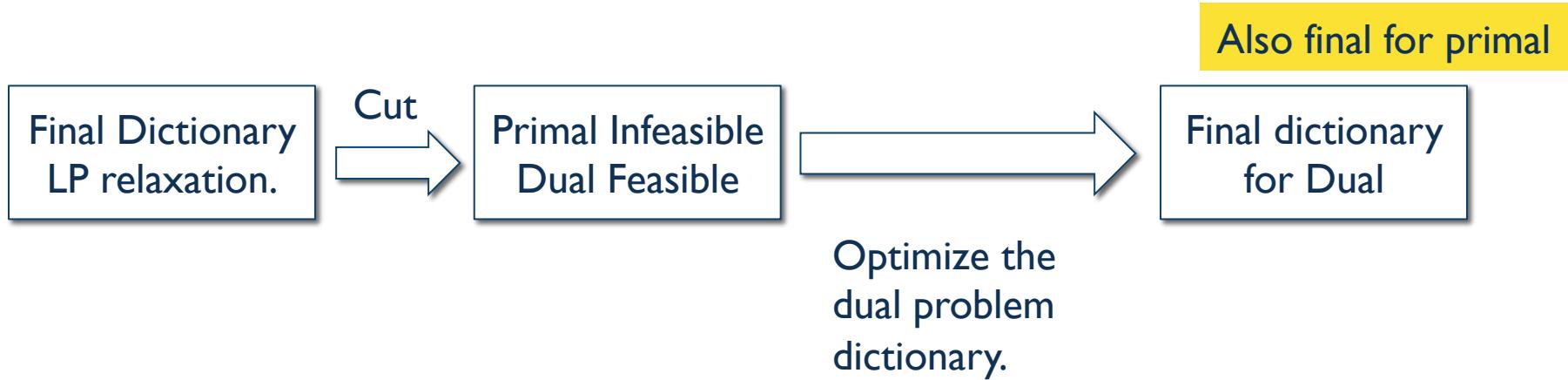
Cutting Plane: Example

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	$+0.1x_2$	$+0.7x_3$	$+0.5x_5$
<hr/> z	1.7	$-1.2x_2$	$-2.3x_3$	$-2.1x_5$

Dual dictionary
is feasible but
non-final.

Naïve Approach: Re-solve initialization phase Simplex.

Cutting Plane: Solving again after cut.



Cutting plane methods

Complete worked out example

Example (ILP)

```
var x1 integer >= 0, <= 10;
```

```
var x2 integer >= 0, <= 10;
```

```
var x3 integer >= 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;
```

$$x_4 : 10 - x_1$$

$$x_5 : 10 - x_2$$

$$x_6 : 10 - x_3$$

$$x_7 : 1 + 2x_1 - 7x_2$$

$$x_8 : 3 - x_1 + 2x_2 - 5x_3$$

$$x_9 : 7 - x_1 - x_2 + 3x_3$$

```
solve;
```

```
display x1, x2, x3;
```

```
end;
```

Slack variables are also integers.

First Iteration

Initial Dictionary

x_4	10	$-1x_1$				
x_5	10		$-1x_2$			
x_6	10			$-1x_3$		
x_7	1	$+2x_1$	$-7x_2$			
x_8	3	$-1x_1$	$+2x_2$	$-5x_3$		
x_9	7	$-1x_1$	$-1x_2$	$+3x_3$		
z	0		$+1x_1$	$+1x_2$	$-5x_3$	

Final Dictionary

x_4	4.33333	$+0.333x_8$	$+0.666667x_9$	$-0.333x_3$	
x_5	8.66667	$-0.333x_8$	$+0.333x_9$	$-2.6667x_3$	
x_6	10			$-1x_3$	
x_7	3	$-3x_8$	$+1x_9$	$-18x_3$	
x_1	5.66667	$-0.333x_8$	$-0.6667x_9$	$+0.333x_3$	
x_2	1.33333	$+0.333x_8$	$-0.333x_9$	$+2.6667x_3$	
z	7		$-1x_9$	$-2x_3$	

$$+0.333x_8 + 0.6667x_9 + 0.6667x_3 \geq 0.666667$$

Second Iteration

x_4	4.33333	$+0.333x_8 + 0.6667x_9 - 0.333x_3$		
x_5	8.66667	$-0.333x_8 + 0.333x_9 - 2.6667x_3$		
x_6	10		$-1x_3$	
x_7	3	$-3x_8$	$+1x_9$	$-18x_3$
x_1	5.66667	$-0.333x_8 - 0.6667x_9 + 0.333x_3$		
x_2	1.33333	$+0.333x_8 - 0.333x_9 + 2.6667x_3$		
x_{10}	-0.666667	$+0.333x_8 + 0.6667x_9 + 0.6667x_3$		
z	7		$-1x_9$	$-2x_3$

Dictionary after
 C_1 added.

Primal infeasible

Dual Dictionary

Feasible + Non-final

y_8	-0	$-0.333y_4 + 0.333y_5$		$+3y_7 + 0.333y_1 - 0.333y_2 - 0.333y_{10}$		
y_9	1	$-0.6667y_4 - 0.333y_5$		$-1y_7 + 0.6667y_1 + 0.333y_2 - 0.6667y_{10}$		
y_3	2	$+0.333y_4 + 2.6667y_5 + 1y_6$	$+18y_7 - 0.333y_1 - 2.6667y_2 - 0.6667y_{10}$			
z	-7	$-4.333y_4 - 8.6667y_5 - 10y_6 - 3y_7 - 5.6667y_1 - 1.333y_2 + 0.6667y_{10}$				

Naming convention for dual: x_j is complementary to y_j

Second Iteration (cont.)

Final
Dual
Dictionary

y_{10}	1.28571428571	$-1y_4 - 0.285714y_5$	$-1.285714y_9 + 1y_1 + 0.285714y_2 - 0.428571y_8$
y_7	0.142857142857	$-0.142857y_5$	$-0.142857y_9 + 0.142857y_2 + 0.285714y_8$
y_3	3.71428571429	$+1y_4 + 0.285714y_5 + 1y_6$	$-1.714286y_9 - 1y_1 - 0.285714y_2 + 5.428571y_8$
z	-6.57142857143	$-5y_4 - 8.428571y_5 - 10y_6 - 0.428571y_9 - 5y_1 - 1.571429y_2 - 1.142857y_8$	

Final
Primal
Dictionary

x_4	5	$+1x_{10}$	$-1x_3$
x_5	8.42857142857	$+0.285714x_{10} + 0.142857x_7 - 0.285714x_3$	
x_6	10		$-1x_3$
x_9	0.428571428571	$+1.285714x_{10} + 0.142857x_7 + 1.714286x_3$	
x_1	5	$-1x_{10}$	$+1x_3$
x_2	1.57142857143	$-0.285714x_{10} - 0.142857x_7 + 0.285714x_3$	
x_8	1.14285714286	$+0.428571x_{10} - 0.285714x_7 - 5.428571x_3$	
z	6.57142857143	$-1.285714x_{10} - 0.142857x_7 - 3.714286x_3$	

Cut C_2

$$0.285714x_{10} + 0.142857x_7 + 0.714286x_3 \geq 0.571428571429$$

Third Iteration

Dictionary after
C₂ added.

Primal Infeasible

x_4	5	$+1x_{10}$	$-1x_3$
x_5	8.42857142857	$+0.285714x_{10} + 0.142857x_7 - 0.285714x_3$	
x_6	10		$-1x_3$
x_9	0.428571428571	$+1.285714x_{10} + 0.142857x_7 + 1.714286x_3$	
x_1	5	$-1x_{10}$	$+1x_3$
x_2	1.57142857143	$-0.285714x_{10} - 0.142857x_7 + 0.285714x_3$	
x_8	1.14285714286	$+0.428571x_{10} - 0.285714x_7 - 5.428571x_3$	
x_{11}	-0.571428571429	$+0.285714x_{10} + 0.142857x_7 + 0.714286x_3$	
z	6.57142857143	$-1.285714x_{10} - 0.142857x_7 - 3.714286x_3$	

y_{10}	1.28571428571	$-1y_4 - 0.285714y_5$	$-1.285714y_9 + 1y_1 + 0.285714y_2 - 0.428571y_8 - 0.285714y_{11}$
y_7	0.142857142857	$-0.142857y_5$	$-0.142857y_9 + 0.142857y_2 + 0.285714y_8 - 0.142857y_{11}$
y_3	3.71428571429	$+1y_4 + 0.285714y_5 + 1y_6$	$-1.714286y_9 - 1y_1 - 0.285714y_2 + 5.428571y_8 - 0.714286y_{11}$
z	-6.57142857143	$-5y_4 - 8.428571y_5 - 10y_6 - 0.428571y_9 - 5y_1 - 1.571429y_2 - 1.142857y_8 + 0.571429y_{11}$	

Dual dictionary: Feasible + Non-final

Third Iteration (Cont.).

Dual Final
Dictionary

y_{10}	1	$-1y_4$	$-1y_9 + 1y_1$	$-1y_8 + 2y_7$	
y_{11}	1		$-1y_9$	$+1y_2 + 2y_8 - 7y_7$	
y_3	3	$+1y_4 + 1y_5$	$+1y_6$	$-1y_9 - 1y_1 - 1y_2 + 4y_8 + 5y_7$	
z	-6	$-5y_4 - 9y_5 - 10y_6 - 1y_9 - 5y_1 - 1y_2$		$-4y_7$	

Integral Solution

x_4	5	$+1x_{10}$	$-1x_3$
x_5	9		$+1x_{11} - 1x_3$
x_6	10		$-1x_3$
x_9	1	$+1x_{10} + 1x_{11} + 1x_3$	
x_1	5	$-1x_{10}$	$+1x_3$
x_2	1		$-1x_{11} + 1x_3$
x_8	-0	$+1x_{10} - 2x_{11} - 4x_3$	
x_7	4	$-2x_{10} + 7x_{11} - 5x_3$	
z	6	$-1x_{10} - 1x_{11} - 3x_3$	

Final Primal Dictionary

Final Answer

Final Primal Dictionary

x_4	5	$+1x_{10}$	$-1x_3$
x_5	9		$+1x_{11} - 1x_3$
x_6	10		$-1x_3$
x_9	1	$+1x_{10} + 1x_{11} + 1x_3$	
x_1	5	$-1x_{10}$	$+1x_3$
x_2	1		$-1x_{11} + 1x_3$
x_8	-0	$+1x_{10} - 2x_{11} - 4x_3$	
x_7	4	$-2x_{10} + 7x_{11} - 5x_3$	
z	6	$-1x_{10} - 1x_{11} - 3x_3$	

Integral Solution

$$x_1 = 5, x_2 = 1, x_3 = 0$$

ANSWER!!