Solving Mixed Integer Linear Programs

Algorithmic Methods for Mathematical Models

(AMMM)

Enric Rodríguez-Carbonell

October 21, 2020

Mixed Integer Linear Programs (1)

- Essentially, mixed integer linear programs are linear programs with the extra constraint that some variables can only take integer values
- Formally, a mixed integer linear program (MILP, MIP) is of the form

$$\begin{aligned}
&\min c^T x \\
&A_1 x \leq b_1 \\
&A_2 x = b_2 \\
&A_3 x \geq b_3 \\
&x_i \in \mathbb{Z} \quad \forall i \in \mathcal{I}
\end{aligned}$$

- The conditions $x_i \in \mathbb{Z} \quad \forall i \in \mathcal{I}$ are called the integrality constraints: they enforce that variables x_i with index $i \in \mathcal{I}$ can only take integer values
- Example:

$$\begin{aligned}
&\max \ x_1 + x_2 \\
&-2x_1 + 2x_2 \geq 1 \\
&-8x_1 + 10x_2 \leq 13 \\
&x_1, x_2 \geq 0 \\
&x_1 \in \mathbb{Z} && \text{(here, } \mathcal{I} = \{1\}\text{)}
\end{aligned}$$

Mixed Integer Linear Programs (2)

- If all variables need to be integer,
  it is called a (pure) integer linear program (ILP, IP)

- Example:

$$\begin{aligned}
&\max\ x_1 + x_2 \\
&-2x_1 + 2x_2 \geq 1 \\
&-8x_1 + 10x_2 \leq 13 \\
&x_1, x_2 \geq 0 \\
&x_1, x_2 \in \mathbb{Z}
\end{aligned}$$

(here, $\mathcal{I} = \{1, 2\}$)

- If all variables need to be 0 or 1 (aka binary or boolean variables),
  it is called a $0-1$ linear program

- Example:

$$\begin{aligned}
&\max\ x + y + 2z \\
&x + y + z \geq 1 \\
&-x + y \geq 0 \\
&0 \leq x, y, z \leq 1 \\
&x, y, z \in \mathbb{Z}
\end{aligned}$$

Applications of MIP

- Used in contexts where, e.g.:
  - it only makes sense to take integral quantities of certain goods or resources, e.g.:
    - men (human resources planning)
    - power stations (facility location)
  - binary decisions need to be taken
    - producing a product (production planning)
    - assigning a task to a worker (assignment problems)
    - assigning a slot to a course (timetabling)
- And many many more...

Complexity: LP vs. IP

- Including integer variables increases enourmously the modeling power, at the expense of making the problems more difficult to solve

- LP's can be solved in polynomial time with interior-point methods (ellipsoid method, Karmarkar's algorithm)

- Complexity theory tells us Integer Programming is NP-complete problem. So:
  - There is no known polynomial-time algorithm
  - There are little chances that one will ever be found
  - Even small problems may be hard to solve

- What follows is one of the many approaches (and one of the most successful) for attacking IP's

LP Relaxation of a MIP

- Given a MIP

$$
(IP) \qquad \begin{aligned} & \min c^T x \\ & A_1 x \le b_1 \\ & A_2 x = b_2 \\ & A_3 x \ge b_3 \\ & x_i \in \mathbb{Z} \quad \forall i \in \mathcal{I} \end{aligned}
$$

its linear relaxation is the LP obtained by dropping integrality constraints:

$$
(LP) \qquad \begin{aligned} & \min c^T x \\ & A_1 x \le b_1 \\ & A_2 x = b_2 \\ & A_3 x \ge b_3 \end{aligned}
$$

- Can we solve IP by solving LP? By rounding?

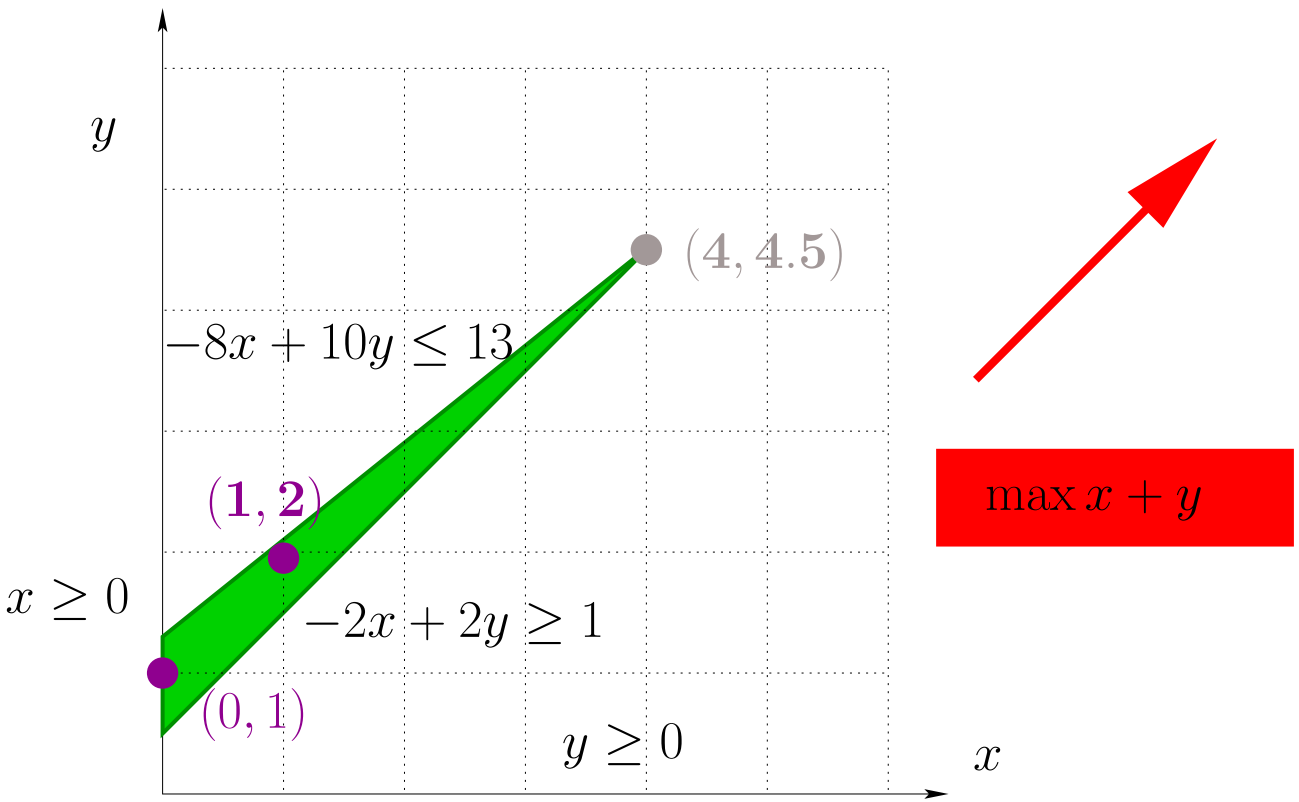
Branch & Bound

- Optimal solution of

$$
\begin{aligned}
\max\ \ x + y & \\
-2x + 2y &\geq 1 \\
-8x + 10y &\leq 13 \\
x, y \geq 0 & \\
x, y \in \mathbb{Z} &
\end{aligned}
$$

  is $(x, y) = (1, 2)$, with objective 3

- Optimal solution of LP relaxation is $(x, y) = (4, 4.5)$, with objective 9.5



Branch & Bound

- Optimal solution of
$$\begin{aligned} \max\ x + y & \\ -2x + 2y &\geq 1 \\ -8x + 10y &\leq 13 \\ x, y \geq 0 & \\ x, y \in \mathbb{Z} & \end{aligned}$$
is $(x, y) = (1, 2)$, with objective 3
- Optimal solution of LP relaxation is $(x, y) = (4, 4.5)$, with objective 9.5
- No direct way of getting from $(4, 4.5)$ to $(1, 2)$ by rounding!
- Something more elaborate is needed: branch & bound

Branch & Bound

- Assume variables are bounded, i.e., have lower and upper bounds
- Let P_0 be the initial problem, $\mathrm{LP}(P_0)$ be the LP relaxation of P_0
- If in optimal solution of $\mathrm{LP}(P_0)$ all integer variables take integer values then that solution is also solution to P_0

Branch & Bound

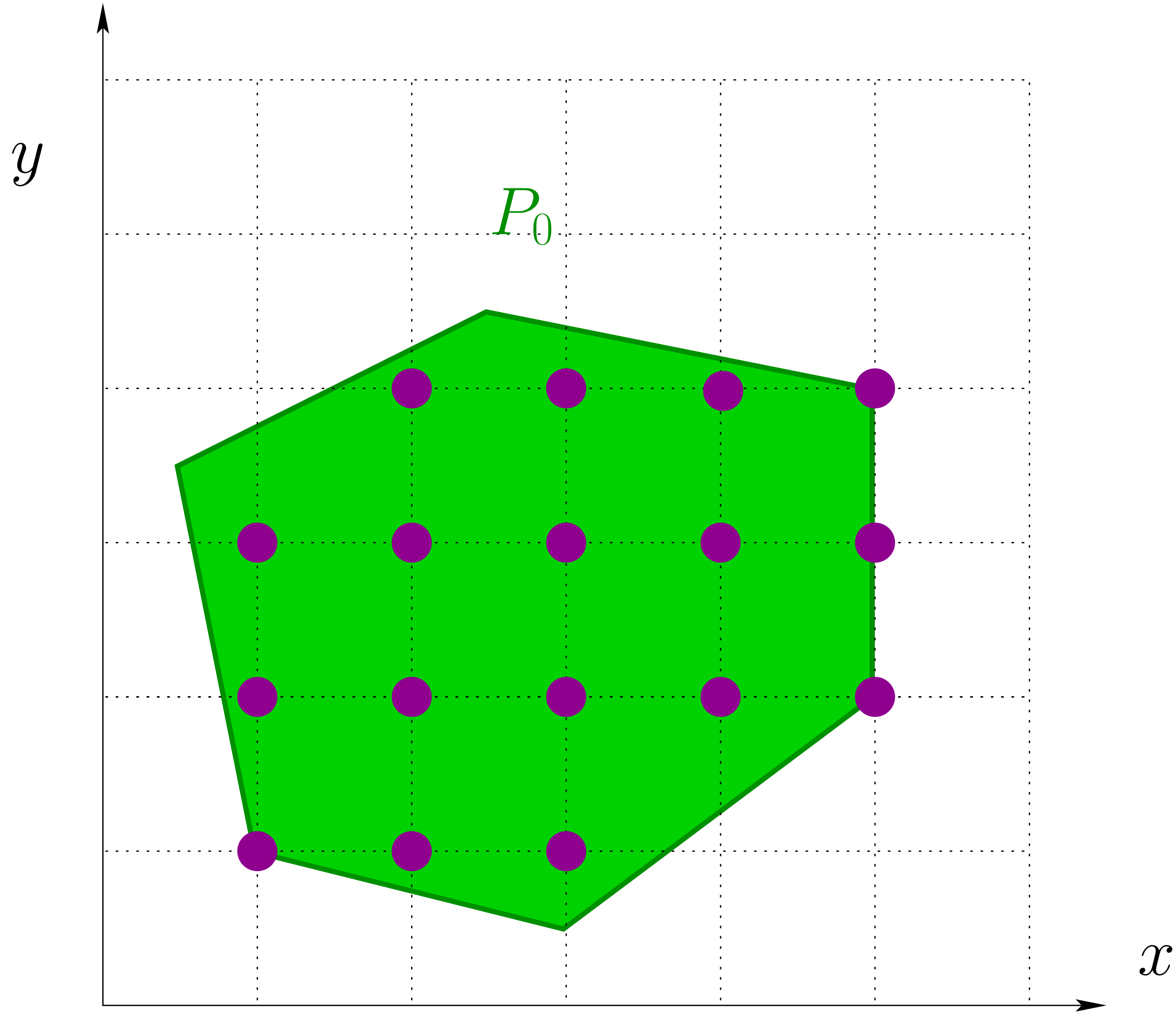
- Assume variables are bounded, i.e., have lower and upper bounds
- Let P_0 be the initial problem, $\mathrm{LP}(P_0)$ be the LP relaxation of P_0
- If in optimal solution of $\mathrm{LP}(P_0)$ all integer variables take integer values then that solution is also solution to P_0
- Else
  - Let x_j be an integer variable
    whose value β_j at optimal solution of $\mathrm{LP}(P_0)$ is such that $\beta_j \notin \mathbb{Z}$.

    Define

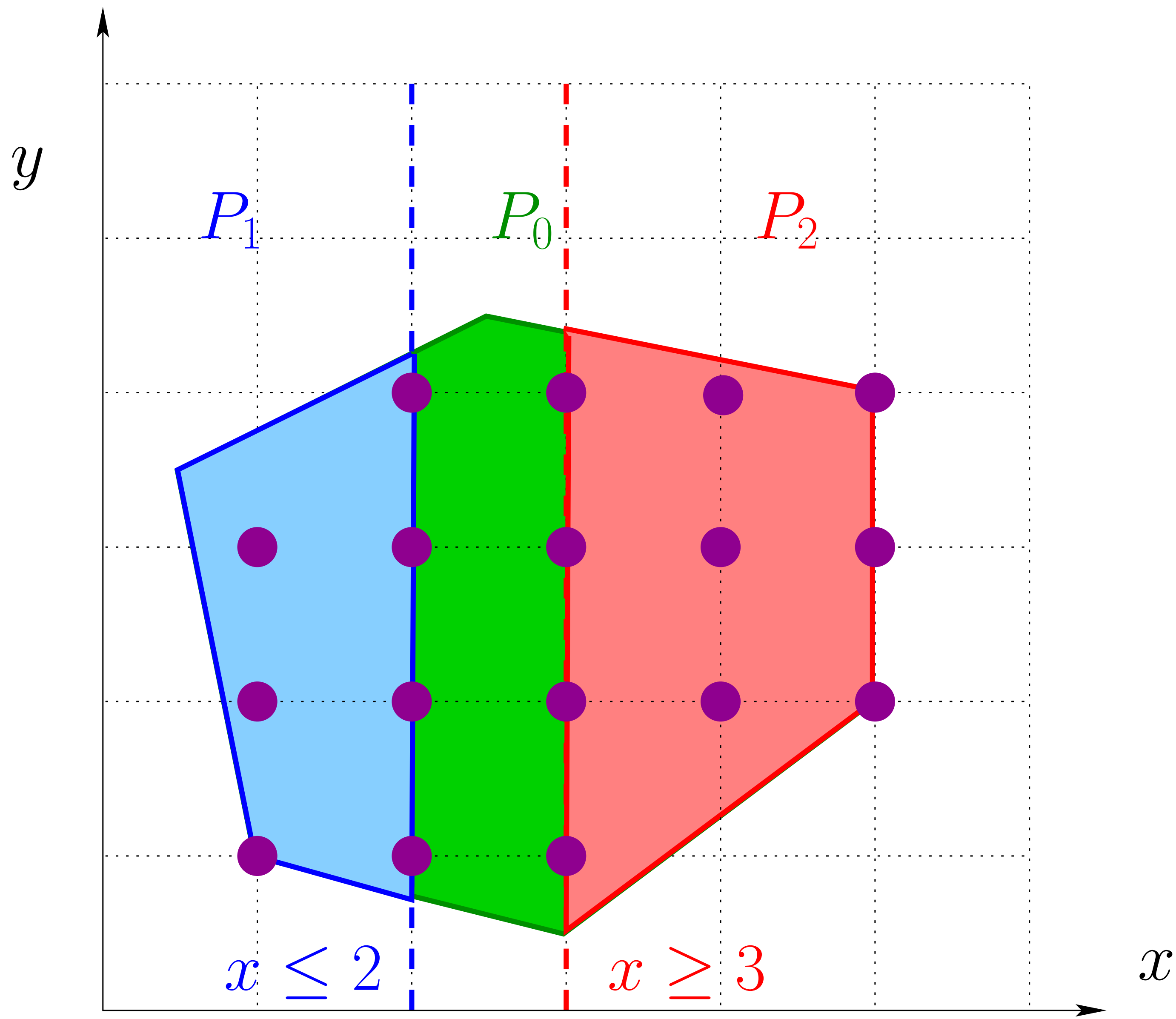
$$
\begin{array}{rcl}
P_1 & := & P_0 \wedge x_j \leq \lfloor \beta_j \rfloor \\
P_2 & := & P_0 \wedge x_j \geq \lceil \beta_j \rceil
\end{array}
$$

  - $\mathrm{feasibleSols}(P_0) = \mathrm{feasibleSols}(P_1) \cup \mathrm{feasibleSols}(P_2)$
  - Idea: solve P_1, solve P_2 and then take the best

Branch & Bound



Branch & Bound



Branch & Bound

- Let x_j be integer variable
  whose value β_j at optimal solution of $\mathrm{LP}(P_0)$ is such that $\beta_j \notin \mathbb{Z}$.

  Each of the problems

  $$P_1 := P_0 \wedge x_j \leq \lfloor \beta_j \rfloor \qquad P_2 := P_0 \wedge x_j \geq \lceil \beta_j \rceil$$

  can be solved recursively

- We can build a binary tree of subproblems
  whose leaves correspond to pending problems still to be solved

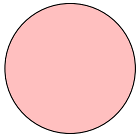
- This procedure terminates as integer vars have finite bounds and,
  at each split, the domain of x_j becomes strictly smaller

- If $\mathrm{LP}(P_i)$ has optimal solution where integer variables take integer values
  then solution is stored

- If $\mathrm{LP}(P_i)$ is infeasible then P_i can be discarded (pruned, fathomed)

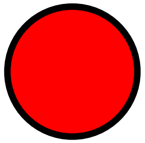
Example

$$
\begin{aligned}
&\min\ -x - y\\
&-2x + 2y \geq 1\\
&-8x + 10y \leq 13\\
&x, y \geq 0\\
&x, y \in \mathbb{Z}
\end{aligned}
$$



Example

$$\begin{aligned}
&\min\ -x - y \\
&-2x + 2y \geq 1 \\
&-8x + 10y \leq 13 \\
&x, y \geq 0 \\
&x, y \in \mathbb{Z}
\end{aligned}$$



Example

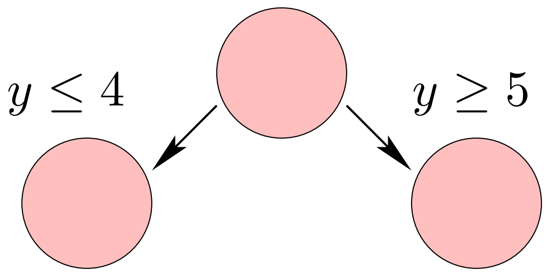
```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
End
====================================================================

CPLEX> optimize
Primal simplex - Optimal:  Objective = - 8.5000000000e+00
Solution time =    0.00 sec.  Iterations = 0 (0)
Deterministic time = 0.00 ticks  (0.37 ticks/sec)

CPLEX> display solution variables x
Variable Name           Solution Value
x                             4.000000

CPLEX> display solution variables y
Variable Name           Solution Value
y                             4.500000
```

Example

```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
End
=====================================================================

CPLEX> optimize
Primal simplex - Optimal:  Objective = - 8.5000000000e+00
Solution time =    0.00 sec.  Iterations = 0 (0)
Deterministic time = 0.00 ticks  (0.37 ticks/sec)

CPLEX> display solution variables x
Variable Name           Solution Value
x                             4.000000

CPLEX> display solution variables y
Variable Name           Solution Value
y                             4.500000
```

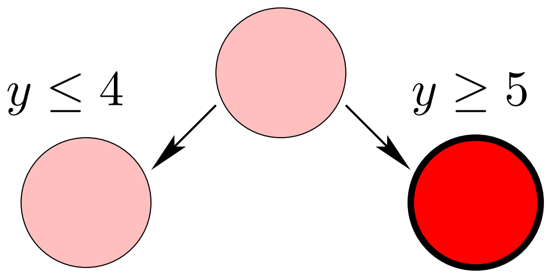
Example

$$\begin{aligned}
&\min\ -x-y\\
&-2x+2y\geq 1\\
&-8x+10y\leq 13\\
&x,y\geq 0\\
&x,y\in\mathbb{Z}
\end{aligned}$$



Example

$$\begin{aligned}
&\min\ -x-y\\
&-2x+2y \geq 1\\
&-8x+10y \leq 13\\
&x,y \geq 0\\
&x,y \in \mathbb{Z}
\end{aligned}$$

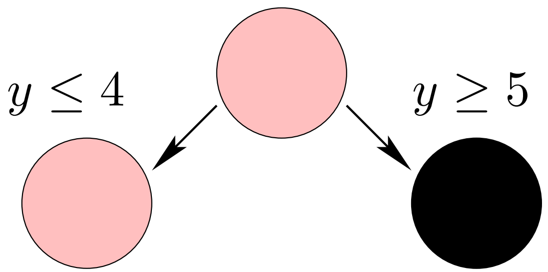


Example

```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
Bounds
y >= 5
End
=====================================================================

CPLEX> optimize
Bound infeasibility column 'x'.
Presolve time = 0.00 sec. (0.00 ticks)
Presolve - Infeasible.
Solution time =    0.00 sec.
Deterministic time = 0.00 ticks  (1.67 ticks/sec)
```

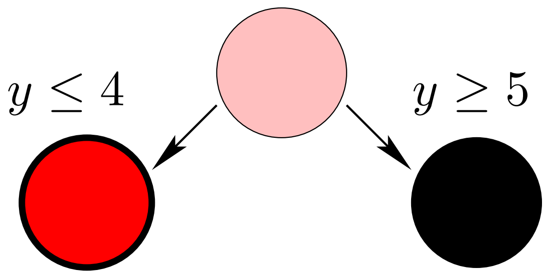
Example

$$\min\ -x - y$$
$$-2x + 2y \geq 1$$
$$-8x + 10y \leq 13$$
$$x, y \geq 0$$
$$x, y \in \mathbb{Z}$$



Example

$$\begin{aligned}
&\min\ -x-y \\
&-2x+2y \geq 1 \\
&-8x+10y \leq 13 \\
&x, y \geq 0 \\
&x, y \in \mathbb{Z}
\end{aligned}$$



Example

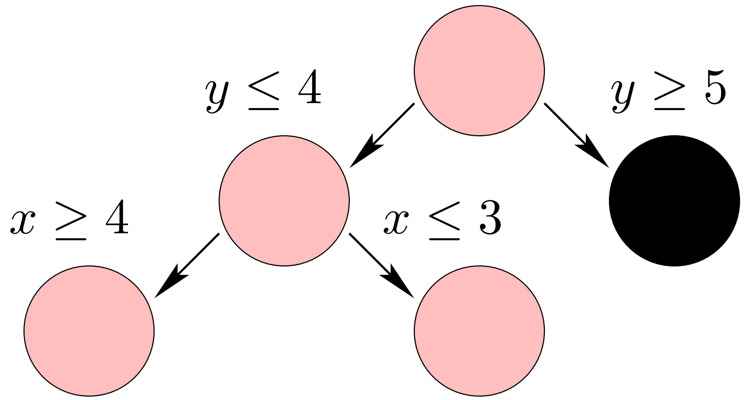
```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
Bounds
y <= 4
End
======================================================================

CPLEX> optimize
Dual simplex - Optimal:  Objective = - 7.5000000000e+00
Solution time =    0.00 sec.  Iterations = 0 (0)
Deterministic time = 0.00 ticks  (2.68 ticks/sec)

CPLEX> display solution variables x
Variable Name           Solution Value
x                             3.500000

CPLEX> display solution variables y
Variable Name           Solution Value
y                             4.000000
```

Example

```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
Bounds
y <= 4
End
============================================================

CPLEX> optimize
Dual simplex - Optimal:  Objective = - 7.5000000000e+00
Solution time =    0.00 sec.  Iterations = 0 (0)
Deterministic time = 0.00 ticks  (2.68 ticks/sec)

CPLEX> display solution variables x
Variable Name           Solution Value
x                             3.500000

CPLEX> display solution variables y
Variable Name           Solution Value
y                             4.000000
```

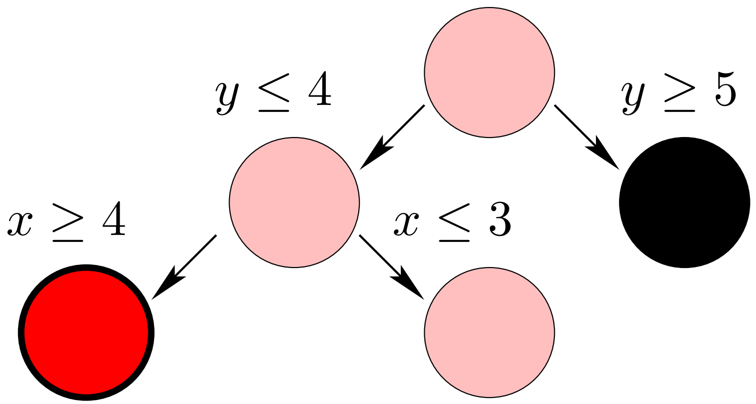
Example

$$
\begin{aligned}
&\min\ -x - y\\
&-2x + 2y \geq 1\\
&-8x + 10y \leq 13\\
&x, y \geq 0\\
&x, y \in \mathbb{Z}
\end{aligned}
$$



Example

$$\begin{aligned}
&\min\ -x - y\\
&-2x + 2y \geq 1\\
&-8x + 10y \leq 13\\
&x, y \geq 0\\
&x, y \in \mathbb{Z}
\end{aligned}$$

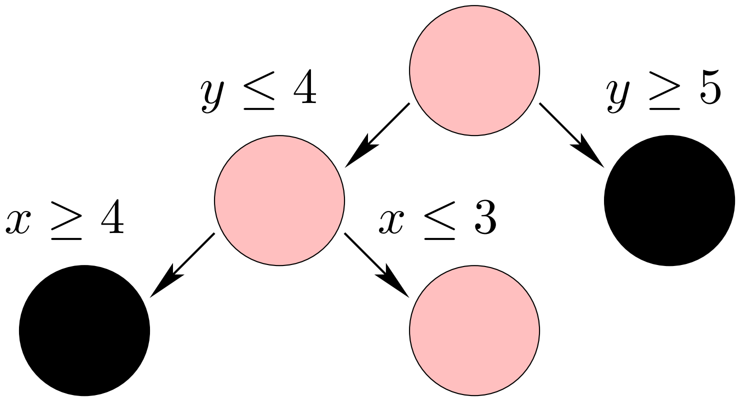


Example

```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
Bounds
x >= 4
y <= 4
End
=====================================================================

CPLEX> optimize
Row 'c1' infeasible, all entries at implied bounds.
Presolve time = 0.00 sec. (0.00 ticks)
Presolve - Infeasible.
Solution time =    0.00 sec.
Deterministic time = 0.00 ticks  (1.11 ticks/sec)
```

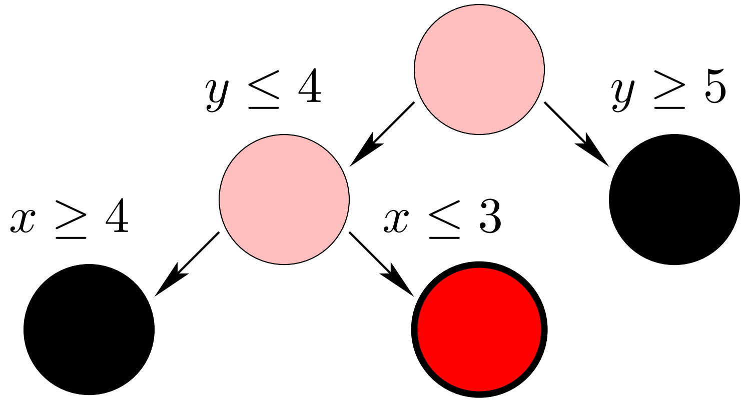
Example

$$
\begin{aligned}
&\min\ -x-y\\
&-2x+2y \geq 1\\
&-8x+10y \leq 13\\
&x, y \geq 0\\
&x, y \in \mathbb{Z}
\end{aligned}
$$



Example

$$\min\ -x - y$$
$$-2x + 2y \geq 1$$
$$-8x + 10y \leq 13$$
$$x, y \geq 0$$
$$x, y \in \mathbb{Z}$$



Example

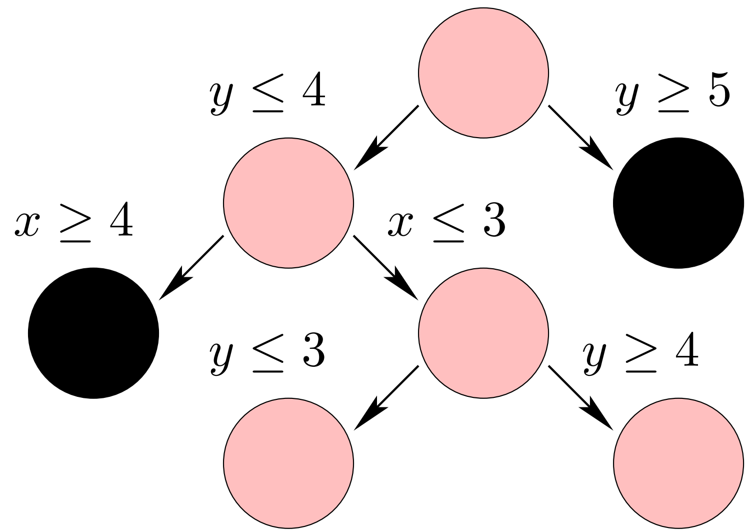
```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
Bounds
x <= 3
y <= 4
End
====================================================================

CPLEX> optimize
Dual simplex - Optimal:  Objective = - 6.7000000000e+00
Solution time =    0.00 sec.  Iterations = 0 (0)
Deterministic time = 0.00 ticks  (2.71 ticks/sec)

CPLEX> display solution variables x
Variable Name           Solution Value
x                             3.000000

CPLEX> display solution variables y
Variable Name           Solution Value
y                             3.700000
```

Example

```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
Bounds
x <= 3
y <= 4
End
====================================================================

CPLEX> optimize
Dual simplex - Optimal:  Objective = - 6.7000000000e+00
Solution time =    0.00 sec.  Iterations = 0 (0)
Deterministic time = 0.00 ticks  (2.71 ticks/sec)

CPLEX> display solution variables x
Variable Name           Solution Value
x                             3.000000

CPLEX> display solution variables y
Variable Name           Solution Value
y                             3.700000
```

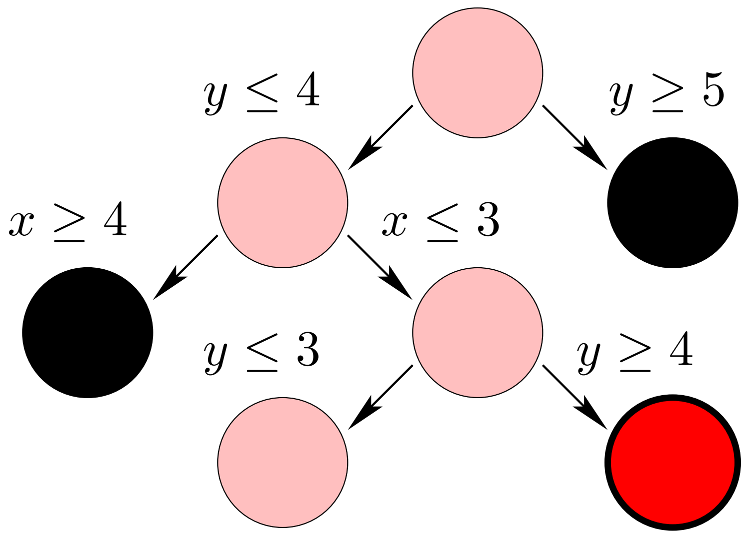
Example

$$
\begin{aligned}
&\min\ -x-y\\
&-2x+2y \geq 1\\
&-8x+10y \leq 13\\
&x, y \geq 0\\
&x, y \in \mathbb{Z}
\end{aligned}
$$



Example

$$
\begin{aligned}
&\min\ -x-y\\
&-2x+2y\geq 1\\
&-8x+10y\leq 13\\
&x,y\geq 0\\
&x,y\in\mathbb{Z}
\end{aligned}
$$

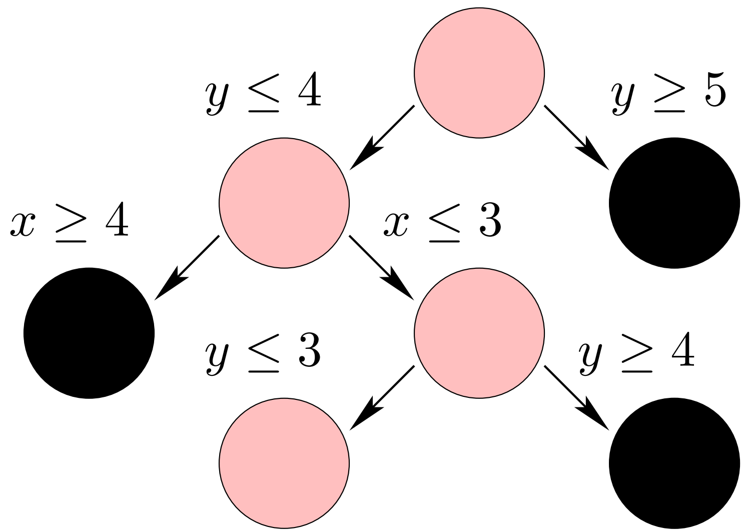


Example

```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
Bounds
x <= 3
y = 4
End
====================================================================

CPLEX> optimize
Bound infeasibility column 'x'.
Presolve time = 0.00 sec. (0.00 ticks)
Presolve - Infeasible.
Solution time =    0.00 sec.
Deterministic time = 0.00 ticks  (1.12 ticks/sec)
```

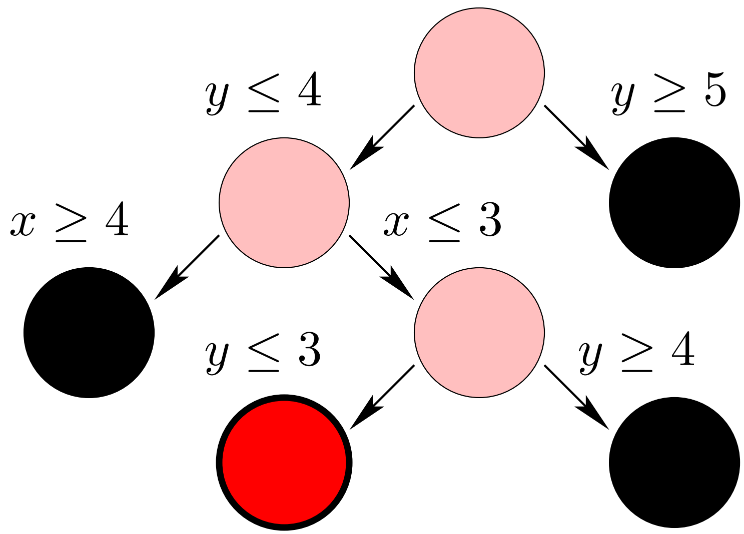
Example

$$\begin{aligned}
&\min\ -x-y\\
&-2x+2y\geq 1\\
&-8x+10y\leq 13\\
&x,y\geq 0\\
&x,y\in\mathbb{Z}
\end{aligned}$$



Example

$$\begin{aligned}
&\min\ -x - y \\
&-2x + 2y \geq 1 \\
&-8x + 10y \leq 13 \\
&x, y \geq 0 \\
&x, y \in \mathbb{Z}
\end{aligned}$$



Example

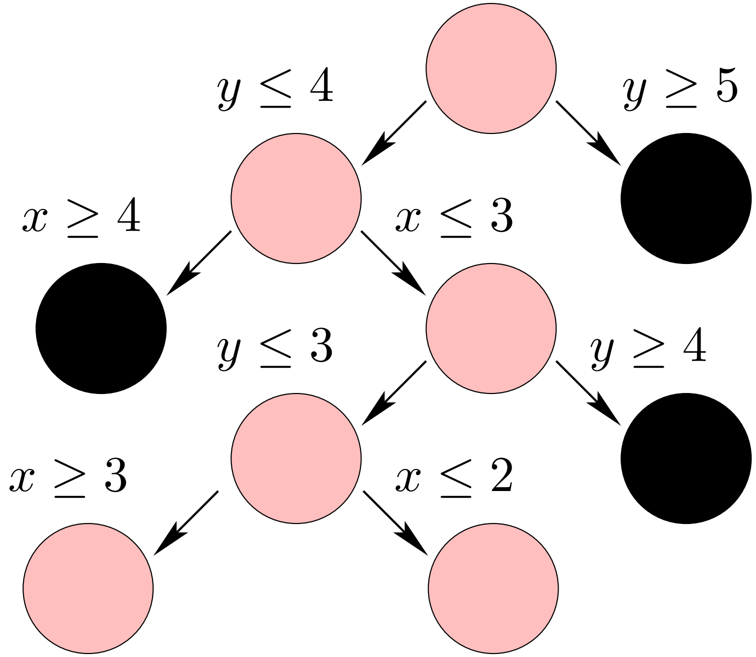
```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
Bounds
x <= 3
y <= 3
End
====================================================================

CPLEX> optimize
Dual simplex - Optimal:  Objective = - 5.5000000000e+00
Solution time =    0.00 sec.  Iterations = 0 (0)
Deterministic time = 0.00 ticks  (2.71 ticks/sec)

CPLEX> display solution variables x
Variable Name           Solution Value
x                             2.500000

CPLEX> display solution variables y
Variable Name           Solution Value
y                             3.000000
```

Example

```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
Bounds
x <= 3
y <= 3
End
====================================================================

CPLEX> optimize
Dual simplex - Optimal:  Objective = - 5.5000000000e+00
Solution time =    0.00 sec.  Iterations = 0 (0)
Deterministic time = 0.00 ticks  (2.71 ticks/sec)

CPLEX> display solution variables x
Variable Name           Solution Value
x                             2.500000

CPLEX> display solution variables y
Variable Name           Solution Value
y                             3.000000
```

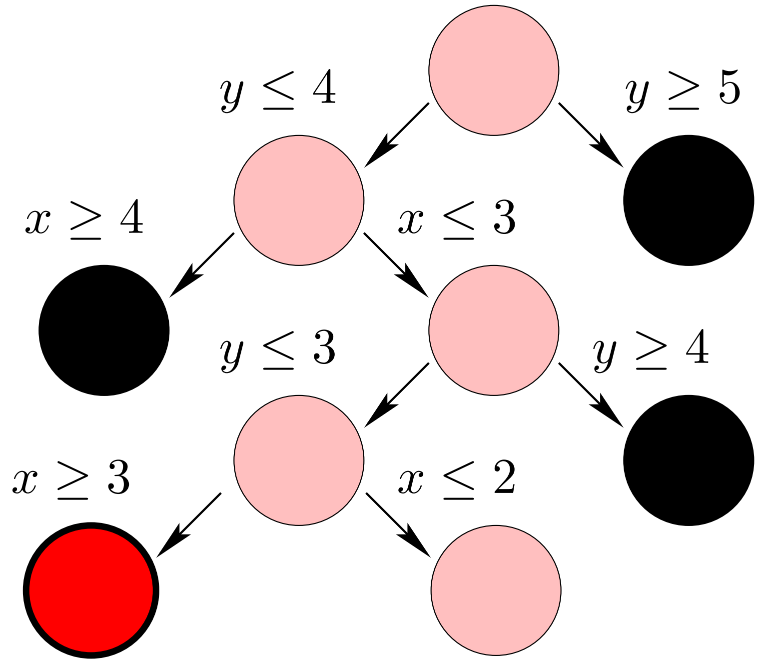
Example

$$
\begin{aligned}
&\min\ -x-y\\
&-2x+2y \geq 1\\
&-8x+10y \leq 13\\
&x,y \geq 0\\
&x,y \in \mathbb{Z}
\end{aligned}
$$



Example

$$\begin{aligned}
&\min\ -x-y\\
&-2x+2y \geq 1\\
&-8x+10y \leq 13\\
&x, y \geq 0\\
&x, y \in \mathbb{Z}
\end{aligned}$$

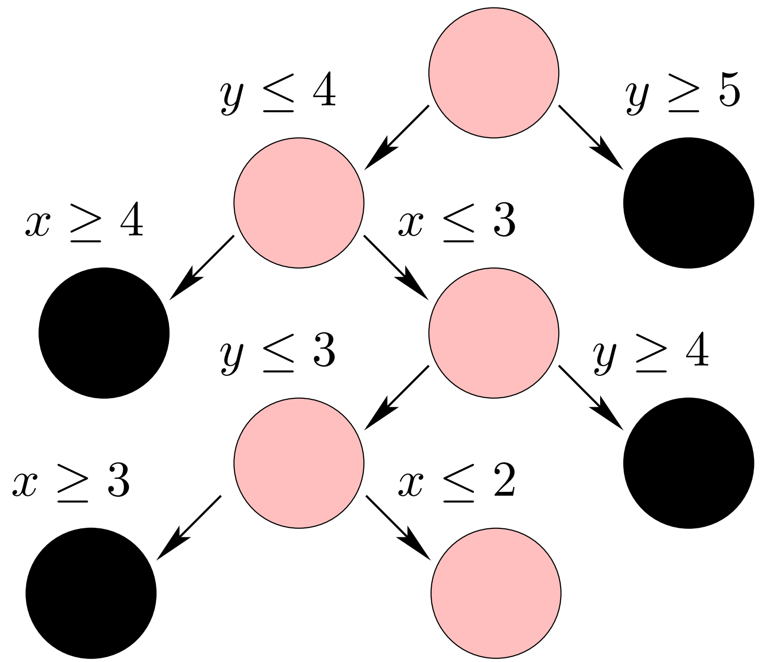


Example

```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
Bounds
x = 3
y <= 3
End
=====================================================================

CPLEX> optimize
Bound infeasibility column 'y'.
Presolve time = 0.00 sec. (0.00 ticks)
Presolve - Infeasible.
Solution time =    0.00 sec.
Deterministic time = 0.00 ticks  (1.11 ticks/sec)
```

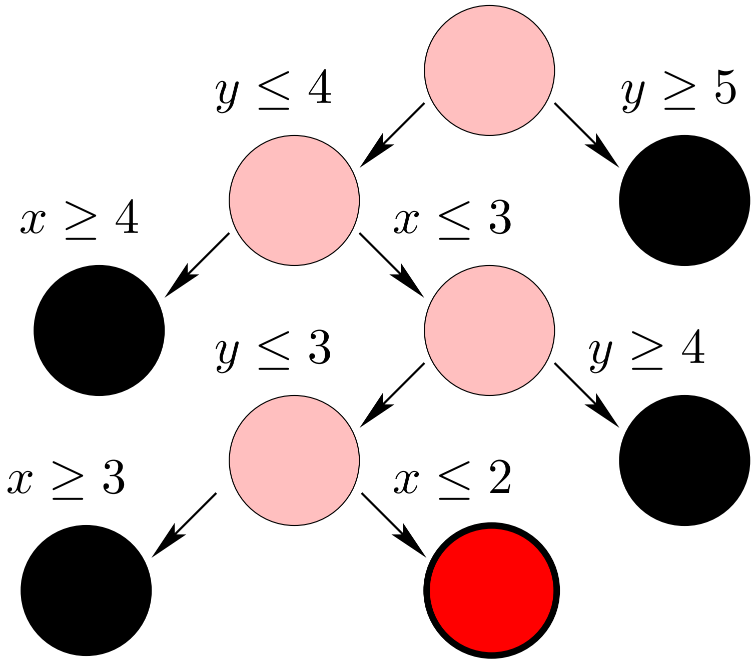
Example

$$\min \ -x - y$$
$$-2x + 2y \geq 1$$
$$-8x + 10y \leq 13$$
$$x, y \geq 0$$
$$x, y \in \mathbb{Z}$$



Example

$$\min \ -x - y$$
$$-2x + 2y \geq 1$$
$$-8x + 10y \leq 13$$
$$x, y \geq 0$$
$$x, y \in \mathbb{Z}$$



Example

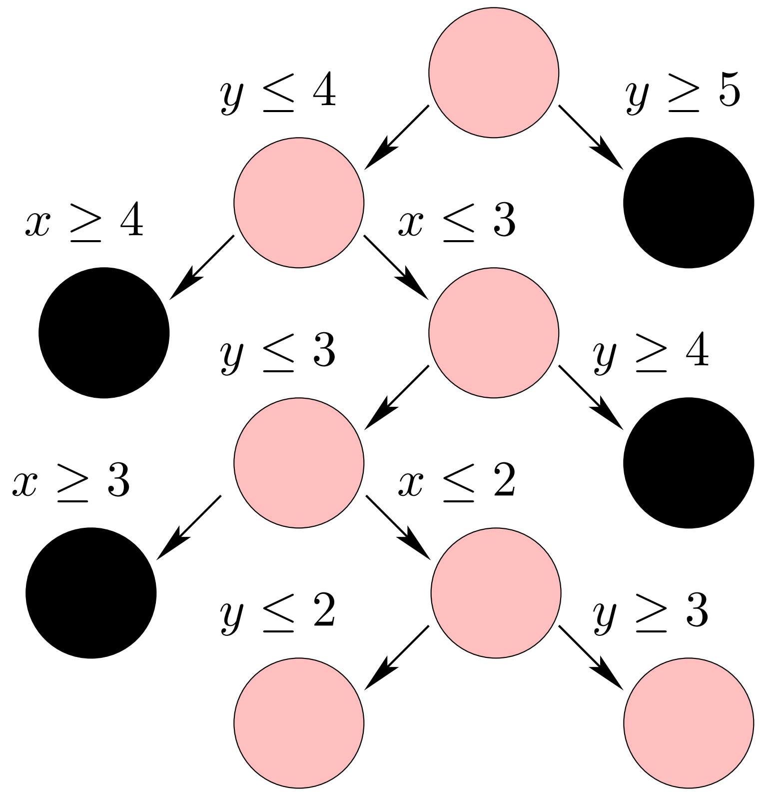
```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
Bounds
x <= 2
y <= 3
End
====================================================================

CPLEX> optimize
Dual simplex - Optimal:  Objective = - 4.9000000000e+00
Solution time =    0.00 sec.  Iterations = 0 (0)
Deterministic time = 0.00 ticks  (2.71 ticks/sec)

CPLEX> display solution variables x
Variable Name           Solution Value
x                             2.000000
CPLEX> display solution variables y
Variable Name           Solution Value
y                             2.900000
```

Example

```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
Bounds
x <= 2
y <= 3
End
====================================================================

CPLEX> optimize
Dual simplex - Optimal:  Objective = - 4.9000000000e+00
Solution time =    0.00 sec.  Iterations = 0 (0)
Deterministic time = 0.00 ticks  (2.71 ticks/sec)

CPLEX> display solution variables x
Variable Name           Solution Value
x                             2.000000
CPLEX> display solution variables y
Variable Name           Solution Value
y                             2.900000
```

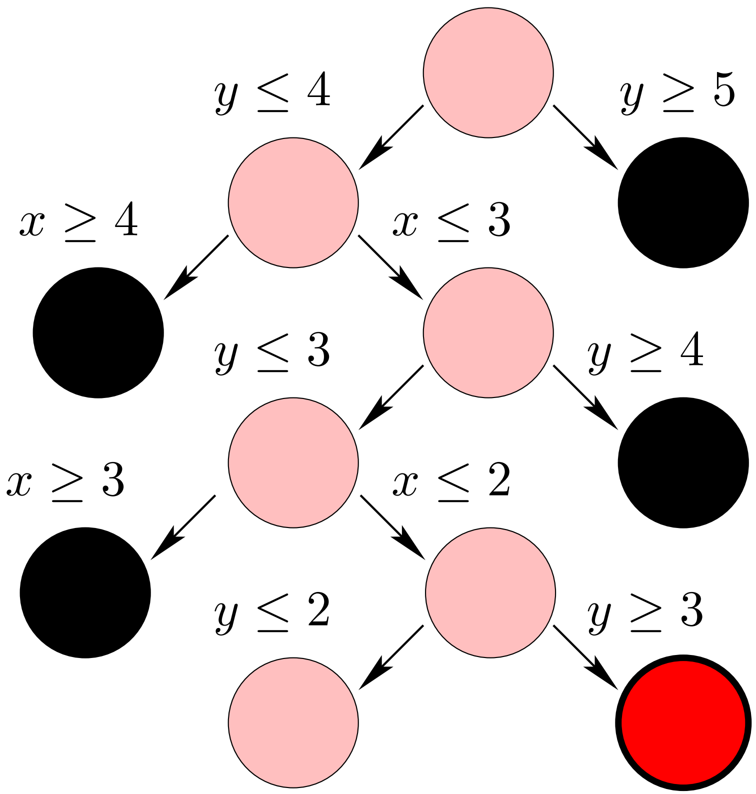
Example

$$
\begin{aligned}
&\min\ -x-y\\
&-2x+2y\geq 1\\
&-8x+10y\leq 13\\
&x,y\geq 0\\
&x,y\in\mathbb{Z}
\end{aligned}
$$



Example

$$\min\ -x-y$$
$$-2x+2y \geq 1$$
$$-8x+10y \leq 13$$
$$x, y \geq 0$$
$$x, y \in \mathbb{Z}$$

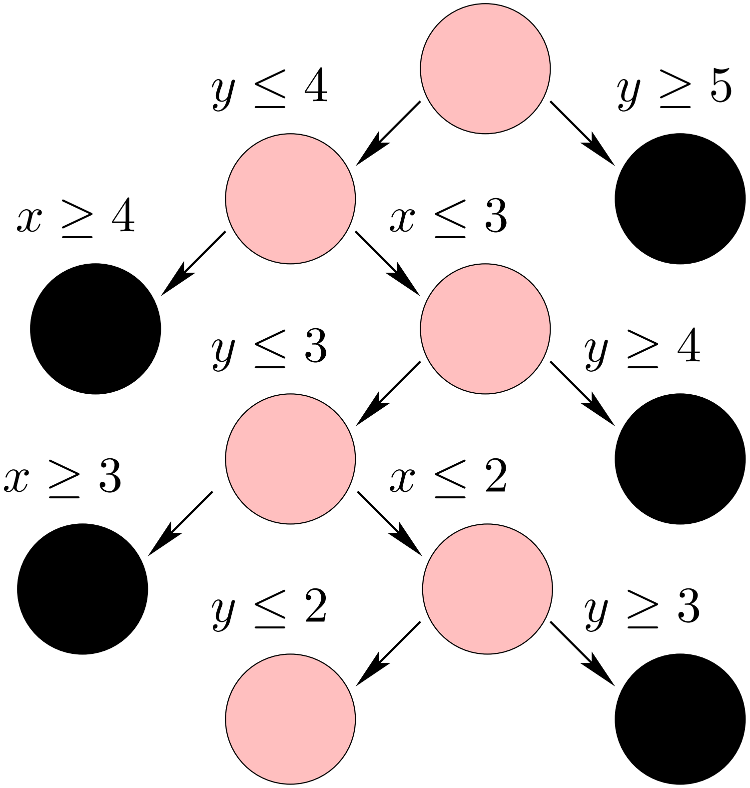


Example

```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
Bounds
x <= 2
y = 3
End
====================================================================

CPLEX> optimize
Bound infeasibility column 'x'.
Presolve time = 0.00 sec. (0.00 ticks)
Presolve - Infeasible.
Solution time =    0.00 sec.
Deterministic time = 0.00 ticks  (1.12 ticks/sec)
```

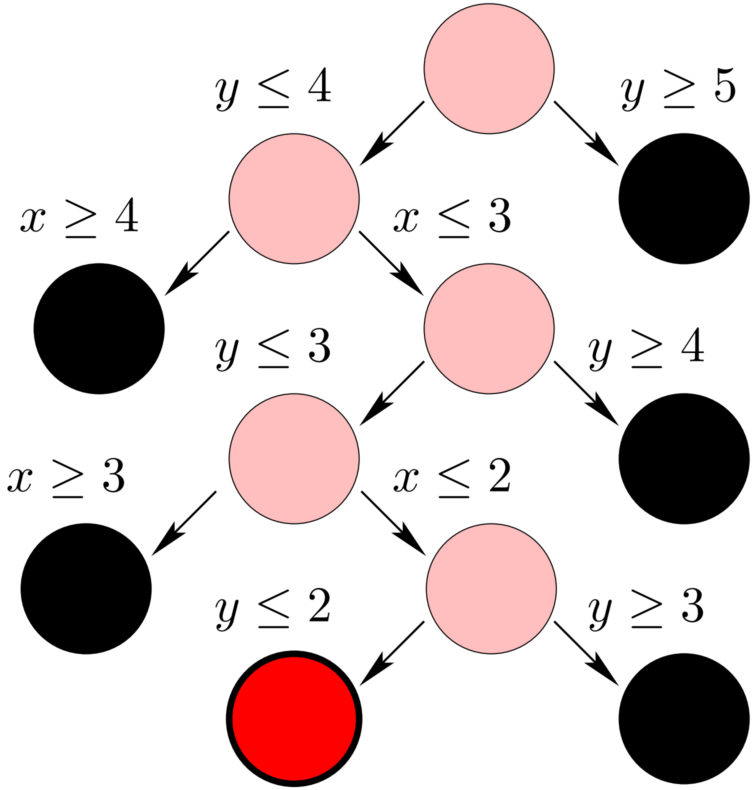
Example

$$\min\ -x - y$$
$$-2x + 2y \geq 1$$
$$-8x + 10y \leq 13$$
$$x, y \geq 0$$
$$x, y \in \mathbb{Z}$$



Example

$$\begin{aligned}
&\min\ -x - y \\
&-2x + 2y \geq 1 \\
&-8x + 10y \leq 13 \\
&x, y \geq 0 \\
&x, y \in \mathbb{Z}
\end{aligned}$$



Example

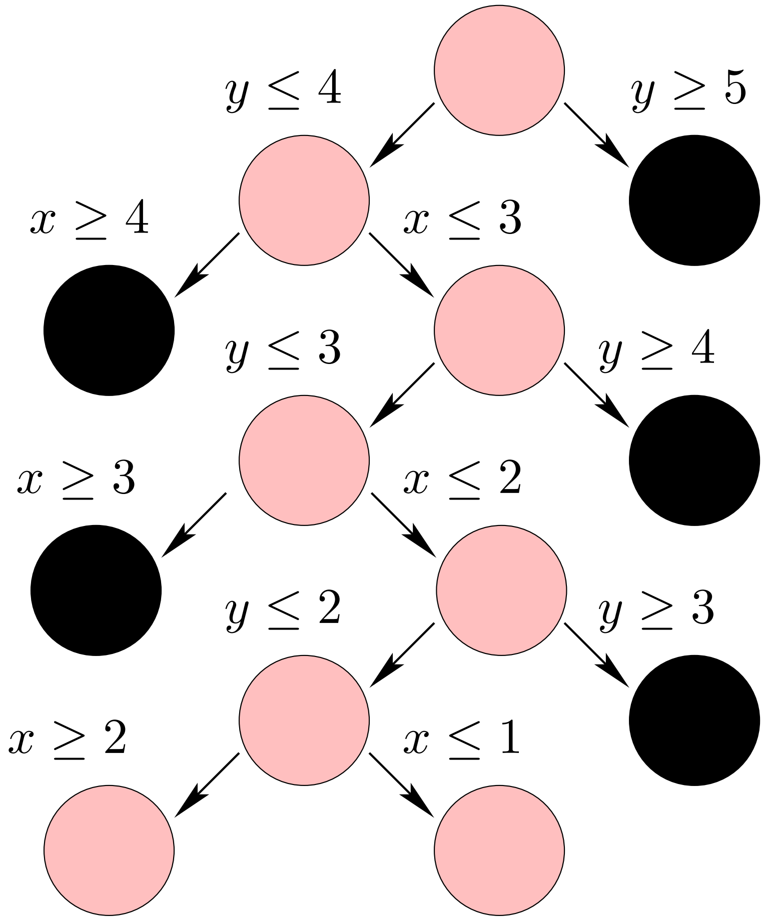
```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
Bounds
x <= 2
y <= 2
End
====================================================================

CPLEX> optimize
Dual simplex - Optimal:  Objective = - 3.5000000000e+00
Solution time =    0.00 sec.  Iterations = 0 (0)
Deterministic time = 0.00 ticks  (2.71 ticks/sec)

CPLEX> display solution variables x
Variable Name           Solution Value
x                             1.500000

CPLEX> display solution variables y
Variable Name           Solution Value
y                             2.000000
```

Example

```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
Bounds
x <= 2
y <= 2
End
====================================================================

CPLEX> optimize
Dual simplex - Optimal:  Objective = - 3.5000000000e+00
Solution time =    0.00 sec.  Iterations = 0 (0)
Deterministic time = 0.00 ticks  (2.71 ticks/sec)

CPLEX> display solution variables x
Variable Name           Solution Value
x                             1.500000

CPLEX> display solution variables y
Variable Name           Solution Value
y                             2.000000
```

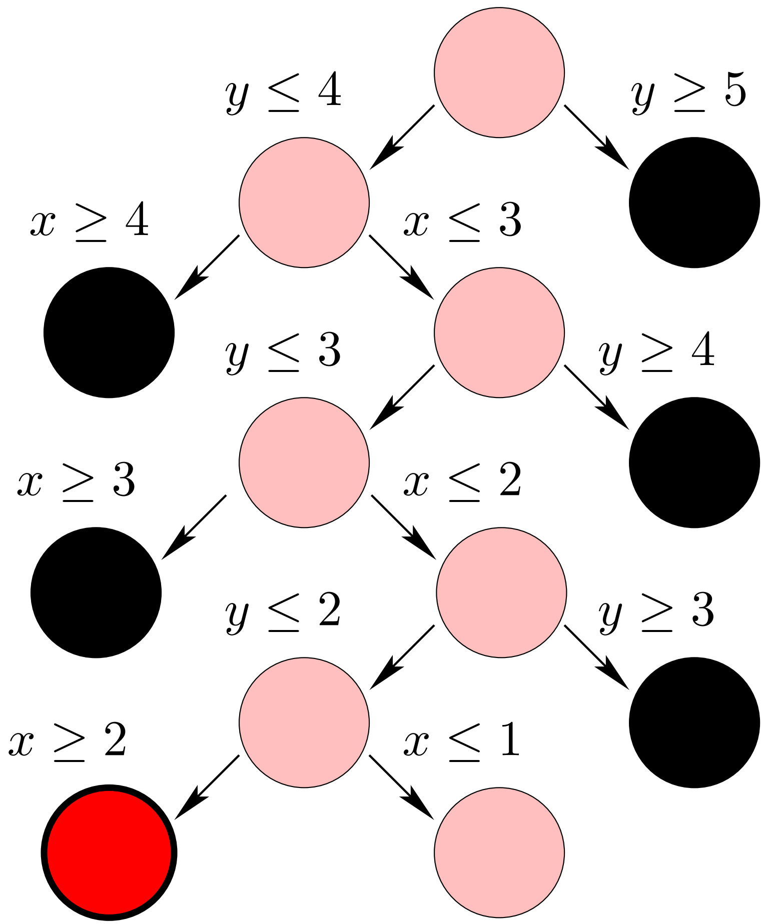
Example

$$\begin{aligned}
&\min\ -x-y\\
&-2x+2y \geq 1\\
&-8x+10y \leq 13\\
&x,y \geq 0\\
&x,y \in \mathbb{Z}
\end{aligned}$$



Example

$$\min\ -x-y$$
$$-2x+2y \geq 1$$
$$-8x+10y \leq 13$$
$$x,y \geq 0$$
$$x,y \in \mathbb{Z}$$

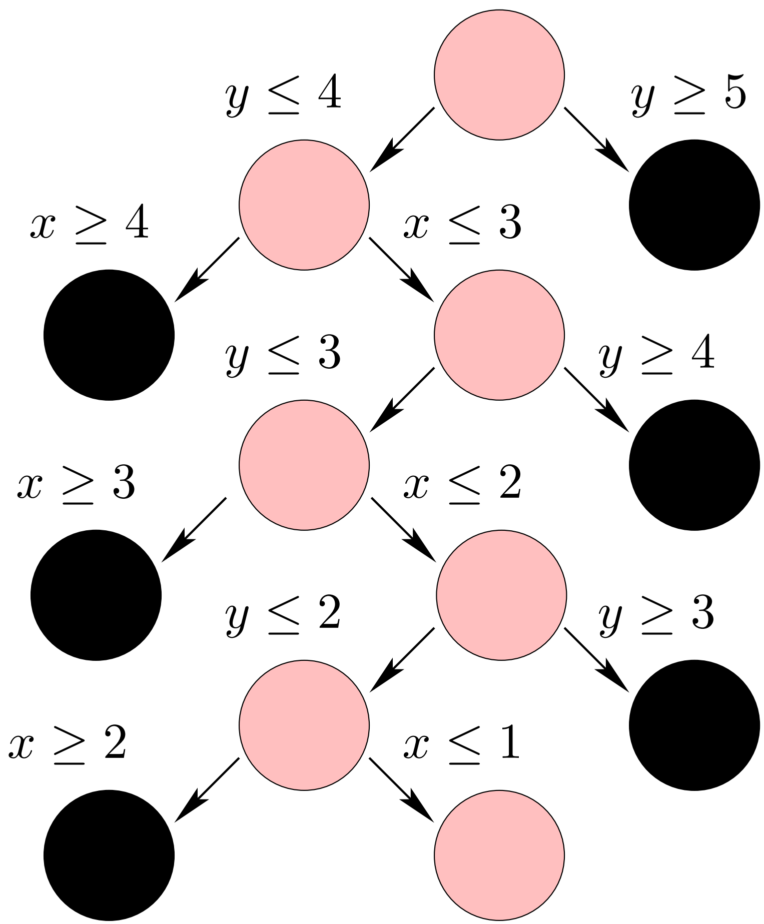


Example

```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
Bounds
x = 2
y <= 2
End
=====================================================================

CPLEX> optimize
Bound infeasibility column 'y'.
Presolve time = 0.00 sec. (0.00 ticks)
Presolve - Infeasible.
Solution time =    0.00 sec.
Deterministic time = 0.00 ticks  (1.11 ticks/sec)
```

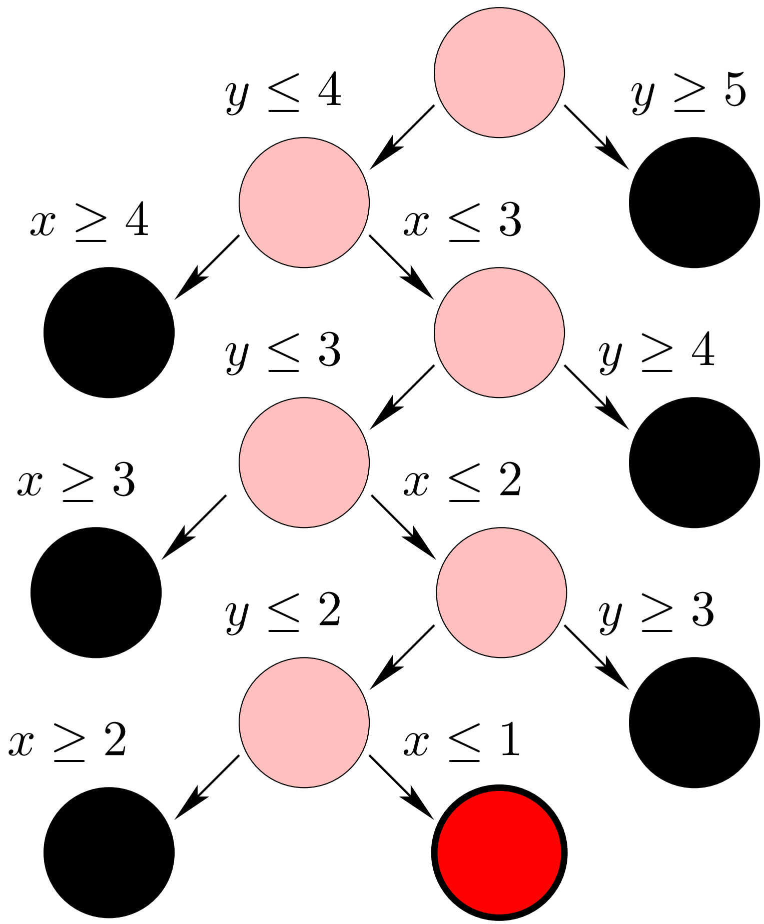
Example

$$\min\ -x-y$$
$$-2x+2y \geq 1$$
$$-8x+10y \leq 13$$
$$x, y \geq 0$$
$$x, y \in \mathbb{Z}$$



Example

$$
\begin{aligned}
&\min\ -x-y\\
&-2x+2y \geq 1\\
&-8x+10y \leq 13\\
&x,y \geq 0\\
&x,y \in \mathbb{Z}
\end{aligned}
$$

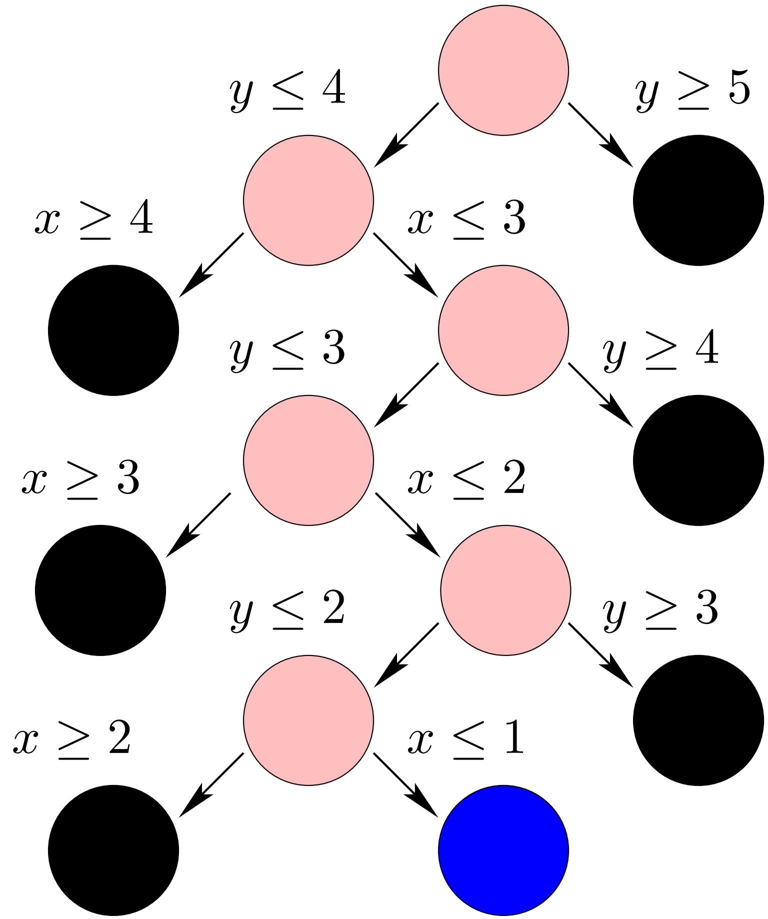


Example

```
Min - x - y
Subject To
-2 x +  2 y >= 1
-8 x + 10 y <= 13
Bounds
x <= 1
y <= 2
End
=====================================================================

CPLEX> optimize
Dual simplex - Optimal:  Objective = - 3.0000000000e+00
Solution time =    0.00 sec.  Iterations = 0 (0)
Deterministic time = 0.00 ticks  (2.40 ticks/sec)

CPLEX> display solution variables x
Variable Name           Solution Value
x                             1.000000

CPLEX> display solution variables y
Variable Name           Solution Value
y                             2.000000
```

Example

$$\min\ -x - y$$
$$-2x + 2y \geq 1$$
$$-8x + 10y \leq 13$$
$$x, y \geq 0$$
$$x, y \in \mathbb{Z}$$



Pruning in Branch & Bound

- We have already seen that if relaxation is infeasible,
  the problem can be pruned

- Now assume that
  a solution (satisfying the integrality constraints) has been previously found

- If solution has cost Z then any pending problem P_j whose relaxation has optimal value $\geq Z$ can be ignored, since

$$\text{cost}(P_j) \geq \text{cost}(\mathrm{LP}(P_j)) \geq Z$$

  The optimum will not be in any descendant of P_j!

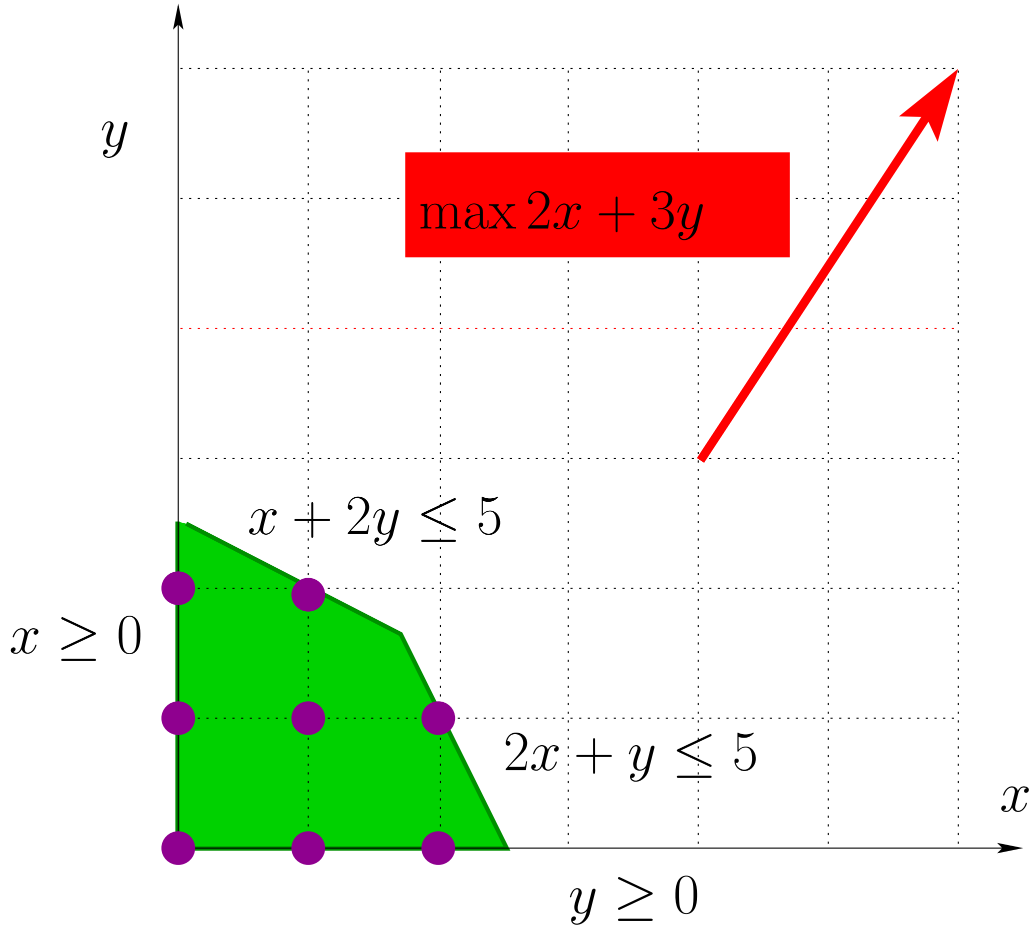
- This cost-based pruning of the search tree has a huge impact
  on the efficiency of Branch & Bound

Branch & Bound: Algorithm

```
S := {P_0}                                              /* set of pending problems */
Z := +∞                                                 /* best cost found so far */
while S ≠ ∅ do
    remove P from S
    solve LP(P)
    if LP(P) is feasible then                           /* if unfeasible P can be pruned */
        let β be optimal basic solution of LP(P)
        if β satisfies integrality constraints then
            if cost(β) < Z then store β; update Z
        else
            if cost(LP(P)) ≥ Z then continue            /* P can be pruned */
            let x_j be integer variable such that β_j ∉ ℤ
            S := S ∪ { P ∧ x_j ≤ ⌊β_j⌋,  P ∧ x_j ≥ ⌈β_j⌉ }
return Z
```

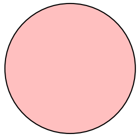
Example

$$\begin{aligned}
&\min\ -2x - 3y\\
&x + 2y \leq 5\\
&2x + y \leq 5\\
&x, y \geq 0\\
&x, y \in \mathbb{Z}
\end{aligned}$$



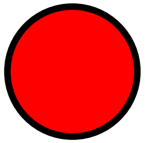
Example

$$
\begin{aligned}
&\min \ -2x - 3y \\
&x + 2y \leq 5 \\
&2x + y \leq 5 \\
&x, y \geq 0 \\
&x, y \in \mathbb{Z}
\end{aligned}
$$



Example

$$\begin{aligned}
&\min\ -2x - 3y\\
&x + 2y \leq 5\\
&2x + y \leq 5\\
&x, y \geq 0\\
&x, y \in \mathbb{Z}
\end{aligned}$$



Example

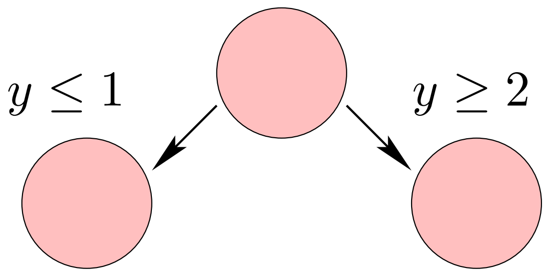
```
Min - 2 x - 3 y
Subject To
  x + 2 y <= 5
2 x +   y <= 5
End
====================================================================

CPLEX> optimize
Dual simplex - Optimal:  Objective = -4.1666666667e+00
Solution time =    0.01 sec.  Iterations = 2 (1)
Deterministic time = 0.00 ticks  (0.38 ticks/sec)

CPLEX> display solution variables x
Variable Name           Solution Value
x                             1.666667

CPLEX> display solution variables y
Variable Name           Solution Value
y                             1.666667
```

Example

```
Min - 2 x - 3 y
Subject To
  x + 2 y <= 5
2 x +   y <= 5
End
====================================================================

CPLEX> optimize
Dual simplex - Optimal:  Objective = -4.1666666667e+00
Solution time =    0.01 sec.  Iterations = 2 (1)
Deterministic time = 0.00 ticks  (0.38 ticks/sec)

CPLEX> display solution variables x
Variable Name           Solution Value
x                             1.666667

CPLEX> display solution variables y
Variable Name           Solution Value
y                             1.666667
```

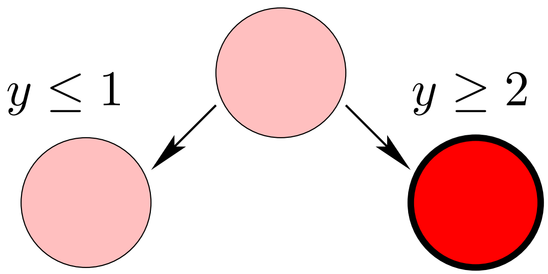
Example

$$\min\ -2x - 3y$$
$$x + 2y \leq 5$$
$$2x + y \leq 5$$
$$x, y \geq 0$$
$$x, y \in \mathbb{Z}$$



Example

$$\min\ -2x - 3y$$
$$x + 2y \leq 5$$
$$2x + y \leq 5$$
$$x, y \geq 0$$
$$x, y \in \mathbb{Z}$$



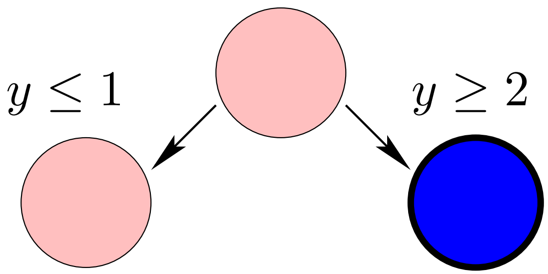
Example

```
Min - 2 x - 3 y
Subject To
  x + 2 y <= 5
2 x +   y <= 5
y >= 2
End
====================================================================

CPLEX> optimize
Dual simplex - Optimal:  Objective = -8.0000000000e+00
Solution time =    0.00 sec.  Iterations = 0 (0)
Deterministic time = 0.00 ticks  (1.62 ticks/sec)

CPLEX> display solution variables x
Variable Name           Solution Value
x                             1.000000

CPLEX> display solution variables y
Variable Name           Solution Value
y                             2.000000
```

Found solution with $x = 1$, $y = 2$ and objective value -8

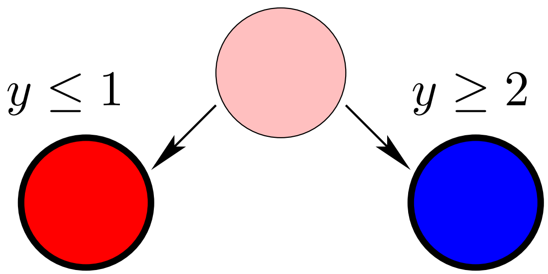
Example

$$\begin{aligned}
&\min \ -x - 3y \\
&x + 2y \leq 5 \\
&2x + y \leq 5 \\
&x, y \geq 0 \\
&x, y \in \mathbb{Z}
\end{aligned}$$



Example

$$\min\ -x - 3y$$
$$x + 2y \leq 5$$
$$2x + y \leq 5$$
$$x, y \geq 0$$
$$x, y \in \mathbb{Z}$$



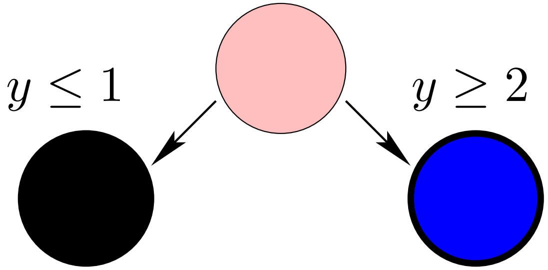
Example

```
Min - 2 x - 3 y
Subject To
  x + 2 y <= 5
2 x +   y <= 5
y <= 1
End
====================================================================

CPLEX> optimize
Dual simplex - Optimal:  Objective = -7.0000000000e+00
Solution time =    0.00 sec.  Iterations = 0 (0)
Deterministic time = 0.00 ticks  (1.71 ticks/sec)
```

Node can be pruned as $-7 \geq -8 =$ cost of best solution found so far!

Example

$$\min\ -x - 3y$$
$$x + 2y \leq 5$$
$$2x + y \leq 5$$
$$x, y \geq 0$$
$$x, y \in \mathbb{Z}$$



Heuristics in Branch & Bound

- Possible choices in Branch & Bound
  - Choice of the pending problem
    - Depth-first search
    - Breadth-first search
    - Best-first search: assuming a relaxation is solved when it is added to the set of pending problems, select the one with best cost value

Heuristics in Branch & Bound

- Possible choices in Branch & Bound
  - Choice of the pending problem
    - Depth-first search
    - Breadth-first search
    - Best-first search: assuming a relaxation is solved when it is added to the set of pending problems, select the one with best cost value
  - Choice of the branching variable: one that is
    - closest to halfway two integer values
    - most important in the model (e.g., 0-1 variable)
    - biggest in a variable ordering
    - the one with the largest coefficient
    - ...

Heuristics in Branch & Bound

- Possible choices in Branch & Bound
  - Choice of the pending problem
    - Depth-first search
    - Breadth-first search
    - Best-first search: assuming a relaxation is solved when it is added to the set of pending problems, select the one with best cost value
  - Choice of the branching variable: one that is
    - closest to halfway two integer values
    - most important in the model (e.g., 0-1 variable)
    - biggest in a variable ordering
    - the one with the largest coefficient
    - ...
- No known strategy is best for all problems!

Remarks on Branch & Bound

- If integer variables are not bounded, Branch & Bound may not terminate:

$$\begin{aligned} &\min\ 0 \\ &1 \leq 3x - 3y \leq 2 \\ &x, y \in \mathbb{Z} \end{aligned}$$

  is infeasible but Branch & Bound loops forever looking for solutions!

Remarks on Branch & Bound

- If integer variables are not bounded, Branch & Bound may not terminate:

$$\begin{aligned} &\min\ 0 \\ &1 \le 3x - 3y \le 2 \\ &x, y \in \mathbb{Z} \end{aligned}$$

  is infeasible but Branch & Bound loops forever looking for solutions!

- E.g., we first find a solution with $x = \frac{2}{3}$

Remarks on Branch & Bound

- If integer variables are not bounded, Branch & Bound may not terminate:

$$
\begin{aligned}
&\min\ 0 \\
&1 \leq 3x - 3y \leq 2 \\
&x, y \in \mathbb{Z}
\end{aligned}
$$

  is infeasible but Branch & Bound loops forever looking for solutions!

- E.g., we first find a solution with $x = \frac{2}{3}$
- In the subproblem with $x \geq 1$ we get a solution with $y = \frac{1}{3}$

Remarks on Branch & Bound

- If integer variables are not bounded, Branch & Bound may not terminate:

$$\begin{aligned} &\min\ 0 \\ &1 \leq 3x - 3y \leq 2 \\ &x, y \in \mathbb{Z} \end{aligned}$$

  is infeasible but Branch & Bound loops forever looking for solutions!

- E.g., we first find a solution with $x = \frac{2}{3}$
- In the subproblem with $x \geq 1$ we get a solution with $y = \frac{1}{3}$
- In the subproblem with $x \geq 1$, $y \geq 1$ we get a solution with $x = \frac{5}{3}$

Remarks on Branch & Bound

- If integer variables are not bounded, Branch & Bound may not terminate:

$$\begin{array}{l} \min\ 0 \\ 1 \le 3x - 3y \le 2 \\ x, y \in \mathbb{Z} \end{array}$$

  is infeasible but Branch & Bound loops forever looking for solutions!

- E.g., we first find a solution with $x = \frac{2}{3}$
- In the subproblem with $x \ge 1$ we get a solution with $y = \frac{1}{3}$
- In the subproblem with $x \ge 1$, $y \ge 1$ we get a solution with $x = \frac{5}{3}$
- In the subproblem with $x \ge 2$, $y \ge 1$ we get a solution with $y = \frac{4}{3}$

Remarks on Branch & Bound

- If integer variables are not bounded, Branch & Bound may not terminate:

$$\begin{aligned}&\min\ 0\\&1 \leq 3x - 3y \leq 2\\&x, y \in \mathbb{Z}\end{aligned}$$

  is infeasible but Branch & Bound loops forever looking for solutions!

- E.g., we first find a solution with $x = \frac{2}{3}$
- In the subproblem with $x \geq 1$ we get a solution with $y = \frac{1}{3}$
- In the subproblem with $x \geq 1$, $y \geq 1$ we get a solution with $x = \frac{5}{3}$
- In the subproblem with $x \geq 2$, $y \geq 1$ we get a solution with $y = \frac{4}{3}$
- In the subproblem with $x \geq 2$, $y \geq 2$ we get a solution with $x = \frac{8}{3}$

Remarks on Branch & Bound

- If integer variables are not bounded, Branch & Bound may not terminate:

$$\begin{aligned}&\min\ 0\\&1 \leq 3x - 3y \leq 2\\&x, y \in \mathbb{Z}\end{aligned}$$

  is infeasible but Branch & Bound loops forever looking for solutions!

- E.g., we first find a solution with $x = \frac{2}{3}$
- In the subproblem with $x \geq 1$ we get a solution with $y = \frac{1}{3}$
- In the subproblem with $x \geq 1$, $y \geq 1$ we get a solution with $x = \frac{5}{3}$
- In the subproblem with $x \geq 2$, $y \geq 1$ we get a solution with $y = \frac{4}{3}$
- In the subproblem with $x \geq 2$, $y \geq 2$ we get a solution with $x = \frac{8}{3}$
- ...

Cutting Planes

- Let us consider a MIP of the form

$$\begin{array}{l} \min \; c^T x \\ x \in S \end{array} \quad \text{where } S = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} A_1 x \le b_1 \\ A_2 x = b_2 \\ A_3 x \ge b_3 \\ x_i \in \mathbb{Z} \quad \forall i \in \mathcal{I} \end{array} \right\}$$

and its linear relaxation

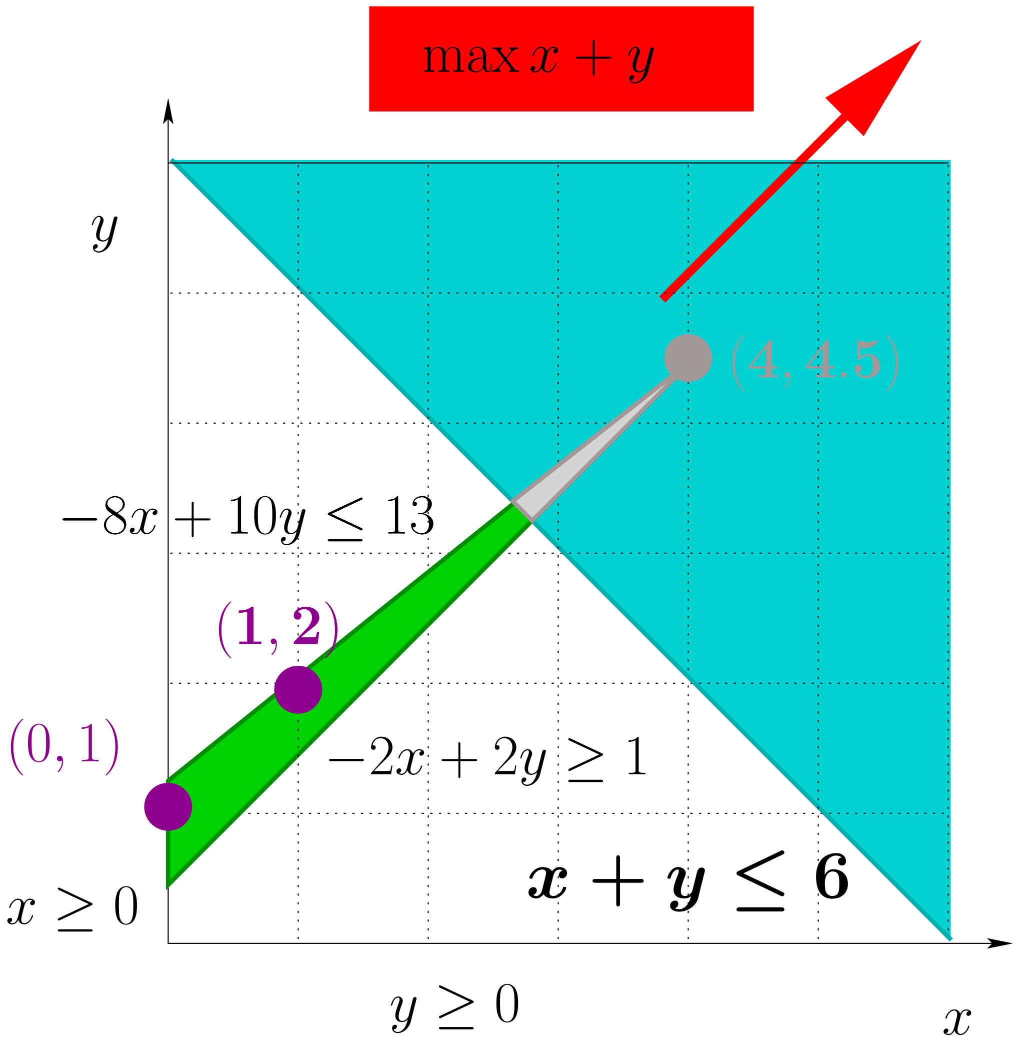
$$\begin{array}{l} \min \; c^T x \\ x \in P \end{array} \quad \text{where } P = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} A_1 x \le b_1 \\ A_2 x = b_2 \\ A_3 x \ge b_3 \end{array} \right\}$$

- Let β be such that $\beta \in P$ but $\beta \notin S$.

  A cut for β is a linear inequality $h^T x \le k$ such that

  - $h^T \sigma \le k$ for any $\sigma \in S$ (feasible solutions of the MIP respect the cut)
  - $h^T \beta > k$ (β does not respect the cut)

Cutting Planes



$$\begin{aligned} \max \ & x + y \\ -2x + 2y &\geq 1 \\ -8x + 10y &\leq 13 \\ x, y &\geq 0 \\ x, y &\in \mathbb{Z} \end{aligned}$$

$x + y \leq 6$ is a cut

Using Cuts for Solving MIP's

- Let $h^T x \leq k$ be a cut. Then the MIP

$$\begin{array}{l} \min \;\; c^T x \\ x \in S' \end{array} \quad \text{where } S' = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} A_1 x \leq b_1 \\ A_2 x = b_2 \\ A_3 x \geq b_3 \\ h^T x \leq k \\ x_i \in \mathbb{Z} \quad \forall i \in \mathcal{I} \end{array} \right\}$$

has the same set of feasible solutions S
but its LP relaxation is strictly more constrained

- Instead of splitting into subproblems (Branch & Bound),
one can add the cut and solve the relaxation of the new problem

- In practice cuts are used together with Branch & Bound:
If after adding some cuts no integer solution is found, then branch

This technique is called Branch & Cut

Gomory Cuts

- Let us see an example of how to derive cuts (namely, Gomory cuts):

$$\begin{aligned} \max\ & x + y \\ -2x + 2y &\geq 1 \\ -8x + 10y &\leq 13 \\ x, y &\geq 0 \\ x, y &\in \mathbb{Z} \end{aligned}$$

Let us transform the MIP into a minimization problem in canonical form:

$$\begin{aligned} \min\ & -x - y \\ -2x + 2y - s_1 &= 1 \\ -8x + 10y + s_2 &= 13 \\ x, y, s_1, s_2 &\geq 0 \\ x, y &\in \mathbb{Z} \end{aligned}$$

Variables (x, y) define an optimal basis for the relaxation:

$$\left\{ \begin{aligned} \min\ & -\tfrac{17}{2} + \tfrac{9}{2}s_1 + s_2 \\ x &= 4 - \tfrac{5}{2}s_1 - \tfrac{1}{2}s_2 \\ y &= \tfrac{9}{2} - 2s_1 - \tfrac{1}{2}s_2 \end{aligned} \right.$$

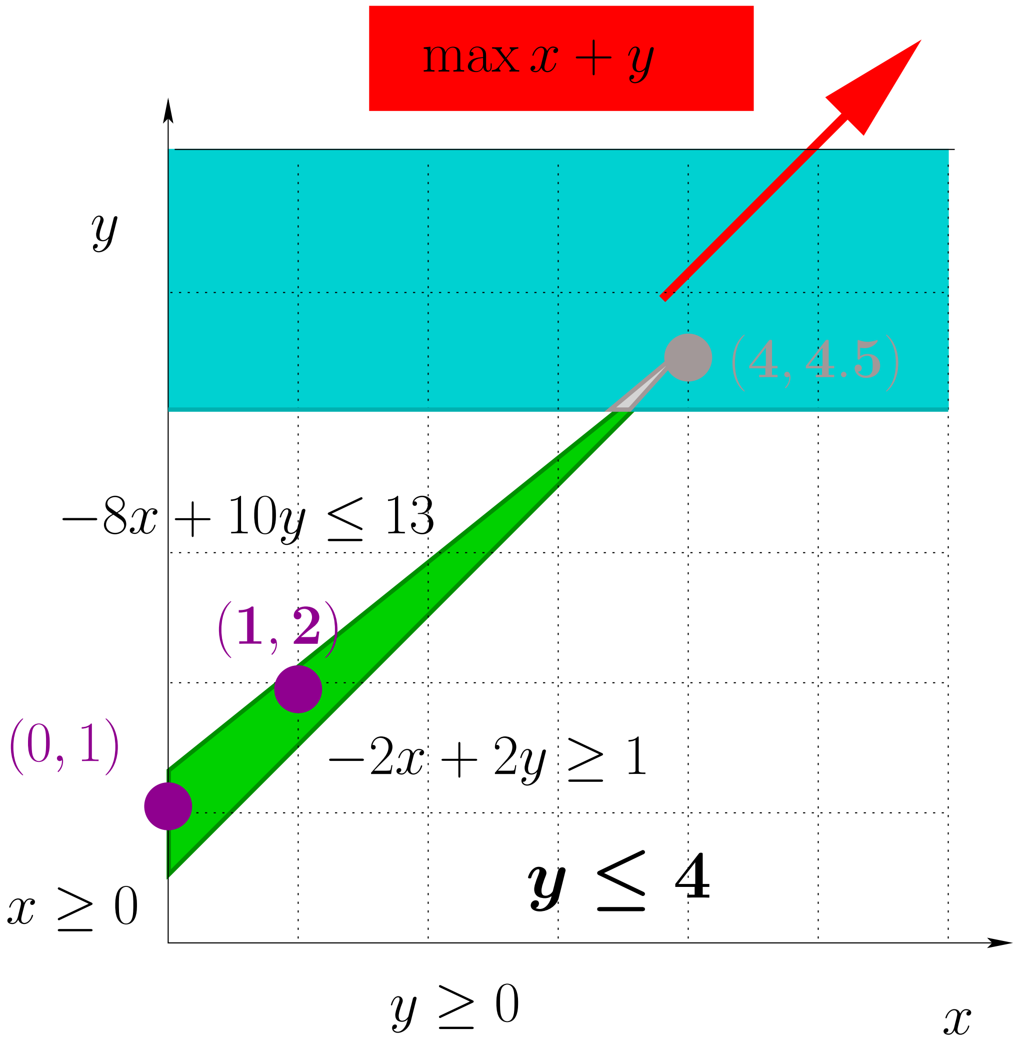
Gomory Cuts

- We can rewrite $y = \frac{9}{2} - 2s_1 - \frac{1}{2}s_2$ as

$$y - 4 = \frac{1}{2} - 2s_1 - \frac{1}{2}s_2$$

- We have that $-2s_1 - \frac{1}{2}s_2 \leq 0$ for any feasible solution
- So $\frac{1}{2} - 2s_1 - \frac{1}{2}s_2 \leq \frac{1}{2} < 1$
- But for any feasible solution, $y - 4 = \frac{1}{2} - 2s_1 - \frac{1}{2}s_2 < 1$ should be integer
- So for any feasible solution $y - 4 \leq 0$, i.e., $y \leq 4$
- Note that $y \leq 4$ is not satisfied by $x = 4, y = \frac{9}{2}$
  It is a cut (that cuts away $x = 4, y = \frac{9}{2}$)

Gomory Cuts



$$
\begin{array}{rcl}
\max \ x + y & & \\
-2x + 2y & \geq & 1 \\
-8x + 10y & \leq & 13 \\
x, y \geq 0 & & \\
x, y \in \mathbb{Z} & &
\end{array}
$$

$y \leq 4$ is a cut

Gomory Cuts

- To solve the MIP with Branch & Cut, now we will try to solve

$$
\begin{aligned}
\max \; x + y & \\
-2x + 2y & \geq 1 \\
-8x + 10y & \leq 13 \\
y & \leq 4 \\
x, y \geq 0 & \\
x, y \in \mathbb{Z} &
\end{aligned}
$$

- The process is repeated:

  1. solve the relaxation
  2. generate new cuts to be added
  3. ...

  until no new cuts can be generated and branch is applied,
  or an integer solution is found

Gomory Cuts

- In general, Gomory cuts can be generated as follows.
- Let us consider a basis B and let β be the associated basic solution. Note that for all $j \in \mathcal{R}$ we have $\beta_j = 0$
- Let x_i be a basic variable such that $i \in \mathcal{I}$ and $\beta_i \notin \mathbb{Z}$
- E.g., this happens in the optimal basis of the relaxation when the basic solution does not meet the integrality constraints
- Let the row of the tableau corresponding to x_i be of the form

$$x_i = \beta_i + \sum_{j \in \mathcal{R}} \alpha_{ij} x_j$$

Gomory Cuts

- Let $x \in S$. Then $x_i \in \mathbb{Z}$ and

$$x_i = \beta_i + \sum_{j \in \mathcal{R}} \alpha_{ij} x_j$$

$$x_i - \beta_i = \sum_{j \in \mathcal{R}} \alpha_{ij} x_j$$

- Let $\delta = \beta_i - \lfloor \beta_i \rfloor$. Then $0 < \delta < 1$
- Hence

$$\begin{aligned} x_i - \lfloor \beta_i \rfloor &= x_i - \beta_i + \beta_i - \lfloor \beta_i \rfloor \\ &= x_i - \beta_i + \delta \\ &= \delta + x_i - \beta_i \\ &= \delta + \sum_{j \in \mathcal{R}} \alpha_{ij} x_j \end{aligned}$$

Gomory Cuts

$$\delta = \beta_i - \lfloor \beta_i \rfloor \qquad x_i - \lfloor \beta_i \rfloor = \delta + \sum_{j \in \mathcal{R}} \alpha_{ij} x_j$$

- Let us define

$$\mathcal{R}^+ = \{j \in \mathcal{R} \mid \alpha_{ij} \geq 0\} \qquad \mathcal{R}^- = \{j \in \mathcal{R} \mid \alpha_{ij} < 0\}$$

- Assume $\sum_{j \in \mathcal{R}} \alpha_{ij} x_j \geq 0$.

Gomory Cuts

$$\delta = \beta_i - \lfloor \beta_i \rfloor \qquad x_i - \lfloor \beta_i \rfloor = \delta + \textstyle\sum_{j \in \mathcal{R}} \alpha_{ij} x_j$$

- Let us define

$$\mathcal{R}^+ = \{j \in \mathcal{R} \mid \alpha_{ij} \geq 0\} \qquad \mathcal{R}^- = \{j \in \mathcal{R} \mid \alpha_{ij} < 0\}$$

- Assume $\sum_{j \in \mathcal{R}} \alpha_{ij} x_j \geq 0$.

  Then $\delta + \sum_{j \in \mathcal{R}} \alpha_{ij} x_j > 0$ and $x_i - \lfloor \beta_i \rfloor \in \mathbb{Z}$ imply

$$\delta + \sum_{j \in \mathcal{R}} \alpha_{ij} x_j \geq 1$$

$$\sum_{j \in \mathcal{R}^+} \alpha_{ij} x_j \geq \sum_{j \in \mathcal{R}} \alpha_{ij} x_j \geq 1 - \delta$$

$$\sum_{j \in \mathcal{R}^+} \frac{\alpha_{ij}}{1 - \delta} x_j \geq 1$$

Gomory Cuts

$$\delta = \beta_i - \lfloor \beta_i \rfloor \qquad x_i - \lfloor \beta_i \rfloor = \delta + \textstyle\sum_{j\in\mathcal{R}} \alpha_{ij} x_j$$

- Let us define

$$\mathcal{R}^+ = \{j \in \mathcal{R} \mid \alpha_{ij} \geq 0\} \qquad \mathcal{R}^- = \{j \in \mathcal{R} \mid \alpha_{ij} < 0\}$$

- Assume $\sum_{j\in\mathcal{R}} \alpha_{ij} x_j \geq 0$.

  Then $\delta + \sum_{j\in\mathcal{R}} \alpha_{ij} x_j > 0$ and $x_i - \lfloor \beta_i \rfloor \in \mathbb{Z}$ imply

$$\delta + \sum_{j\in\mathcal{R}} \alpha_{ij} x_j \geq 1$$

$$\sum_{j\in\mathcal{R}^+} \alpha_{ij} x_j \geq \sum_{j\in\mathcal{R}} \alpha_{ij} x_j \geq 1 - \delta$$

$$\sum_{j\in\mathcal{R}^+} \frac{\alpha_{ij}}{1-\delta} x_j \geq 1$$

  Moreover $\sum_{j\in\mathcal{R}^-} \left(\frac{-\alpha_{ij}}{\delta}\right) x_j \geq 0$

Gomory Cuts

$$\delta = \beta_i - \lfloor \beta_i \rfloor \qquad x_i - \lfloor \beta_i \rfloor = \delta + \sum_{j \in \mathcal{R}} \alpha_{ij} x_j$$

- Let us define

$$\mathcal{R}^+ = \{j \in \mathcal{R} \mid \alpha_{ij} \geq 0\} \qquad \mathcal{R}^- = \{j \in \mathcal{R} \mid \alpha_{ij} < 0\}$$

- Assume $\sum_{j \in \mathcal{R}} \alpha_{ij} x_j < 0$.

Gomory Cuts

$$\delta = \beta_i - \lfloor \beta_i \rfloor \qquad x_i - \lfloor \beta_i \rfloor = \delta + \textstyle\sum_{j \in \mathcal{R}} \alpha_{ij} x_j$$

- Let us define

$$\mathcal{R}^+ = \{ j \in \mathcal{R} \mid \alpha_{ij} \geq 0 \} \qquad \mathcal{R}^- = \{ j \in \mathcal{R} \mid \alpha_{ij} < 0 \}$$

- Assume $\sum_{j \in \mathcal{R}} \alpha_{ij} x_j < 0$.

  Then $\delta + \sum_{j \in \mathcal{R}} \alpha_{ij} x_j < 1$ and $x_i - \lfloor \beta_i \rfloor \in \mathbb{Z}$ imply

$$\delta + \sum_{j \in \mathcal{R}} \alpha_{ij} x_j \leq 0$$

$$\sum_{j \in \mathcal{R}^-} \alpha_{ij} x_j \leq \sum_{j \in \mathcal{R}} \alpha_{ij} x_j \leq -\delta$$

$$\sum_{j \in \mathcal{R}^-} \left(\frac{-\alpha_{ij}}{\delta} \right) x_j \geq 1$$

Gomory Cuts

$$\delta = \beta_i - \lfloor \beta_i \rfloor \qquad x_i - \lfloor \beta_i \rfloor = \delta + \textstyle\sum_{j \in \mathcal{R}} \alpha_{ij} x_j$$

- Let us define

$$\mathcal{R}^+ = \{ j \in \mathcal{R} \mid \alpha_{ij} \geq 0 \} \qquad \mathcal{R}^- = \{ j \in \mathcal{R} \mid \alpha_{ij} < 0 \}$$

- Assume $\sum_{j \in \mathcal{R}} \alpha_{ij} x_j < 0$.

  Then $\delta + \sum_{j \in \mathcal{R}} \alpha_{ij} x_j < 1$ and $x_i - \lfloor \beta_i \rfloor \in \mathbb{Z}$ imply

$$\delta + \sum_{j \in \mathcal{R}} \alpha_{ij} x_j \leq 0$$

$$\sum_{j \in \mathcal{R}^-} \alpha_{ij} x_j \leq \sum_{j \in \mathcal{R}} \alpha_{ij} x_j \leq -\delta$$

$$\sum_{j \in \mathcal{R}^-} \left(\frac{-\alpha_{ij}}{\delta} \right) x_j \geq 1$$

  Moreover $\sum_{j \in \mathcal{R}^+} \frac{\alpha_{ij}}{1-\delta} x_j \geq 0$

Gomory Cuts

- In any case

$$\sum_{j \in \mathcal{R}^-} \left(\frac{-\alpha_{ij}}{\delta}\right) x_j + \sum_{j \in \mathcal{R}^+} \frac{\alpha_{ij}}{1-\delta} x_j \geq 1$$

for any $x \in S$.

However, when $x = \beta$ this inequality is not satisfied (set $x_j = 0$ for $j \in \mathcal{R}$)