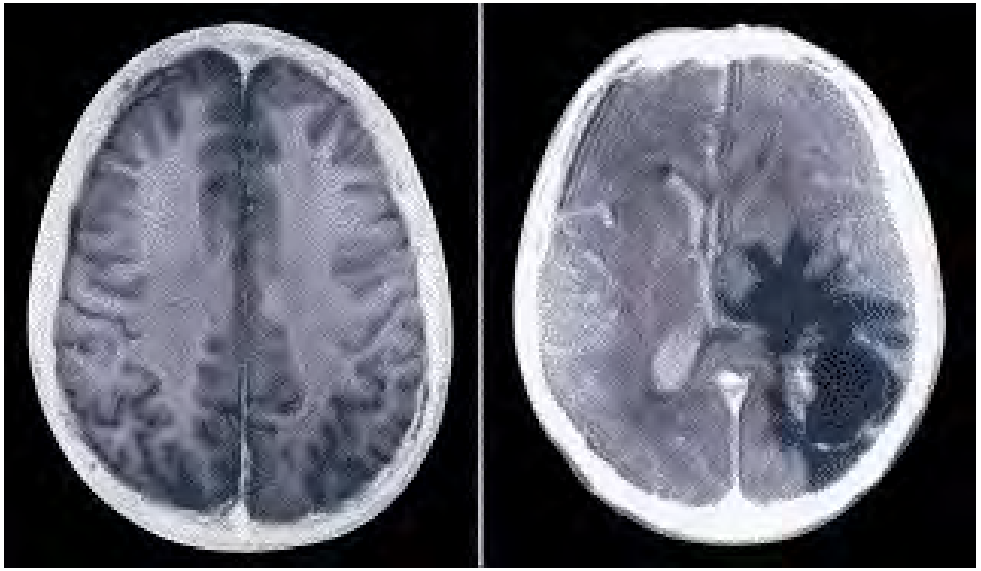
Mixed Integer Optimization

Andrés Ramos (Andres.Ramos@comillas.edu)
Pedro Sánchez (Pedro.Sanchez@comillas.edu)
Sonja Wogrin (Sonja.Wogrin@comillas.edu)

CONTENT

Mixed integer programming problem (MIP)

- Many times, we need integer or binary variables. Some decisions can not be modeled with continuous variables
  - ✓ Investment decisions
  - ✓ Connection of a machine
  - ✓ Location of a warehouse
  - ✓ Selection of a product
  - ✓ Where do you apply radiotherapy to maximize the impact on cancerous cells and minimize the damage to other cells?



Introduction

- A mixed integer programming problem (MIP) is a linear problem (LP) with some integer variables
  - ✓ Mixed integer linear programming problem (MILP)
    - $x \in R^{+}, y \in Z^{+}$
  - ✓ Pure integer programming (PIP)
    - $x \in Z^{+}$
  - ✓ Binary programming (0-1 MIP, 0-1 IP, BIP)
    - $x \in \{0,1\}$: assignment variables, logical variables
- More difficult to solve than LP
- First solution algorithm by Ralph Gomory in 1958

CONTENT

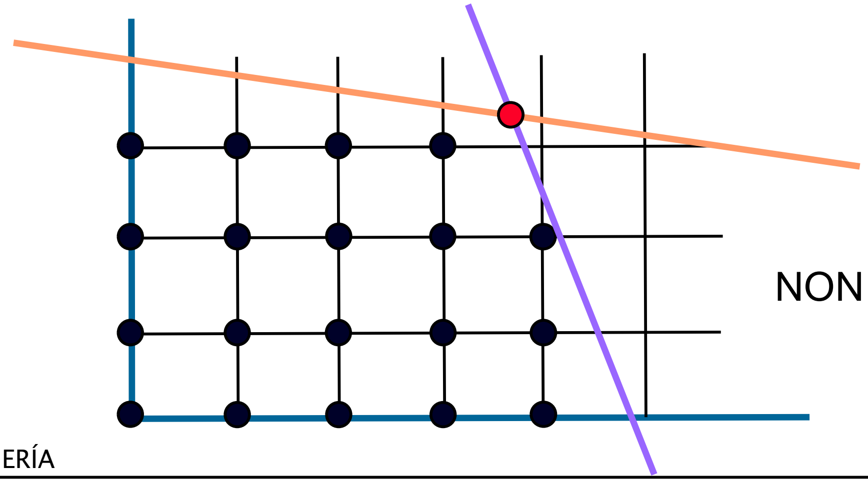
- INTRODUCTION
- ➢ SOLUTION METHODS
- BRANCH AND BOUND
- DUALITY (master)
- PREPROCESSING (master)
- BRANCH AND CUT METHOD (master)

Solution methods

- Linear relaxation and discretization
- Exhaustive enumeration
- Branch and bound
- Cutting plane method
- Branch and cut

Linear relaxation and discretization (i)

- Relaxed problem: integer variables are allowed to take continue values
- If the solution satisfies the integrality constraints, then it is the optimum of the integer problem
  - ✓ Integer polytope: all extreme points are integer.
  - ✓ Coincide with the convex hull of the solution.
  - ✓ It is an integer if matrix A is totally unimodular (every square submatrix has determinant 1, 0, or -1).
  - ✓ Transportation problem, assignment problem, minimum cost flow.



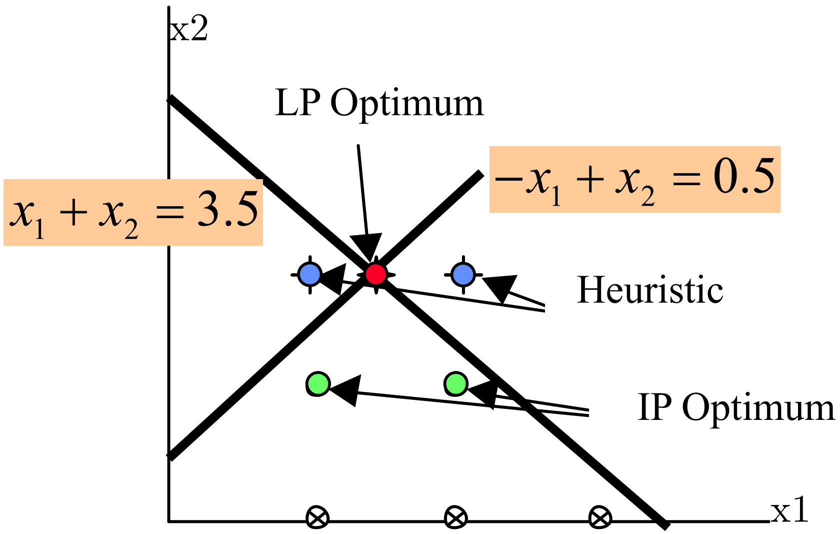
NON integer polytope

Linear relaxation and discretization (ii)

- The integer solution is NOT necessarily the solution of a relaxed problem heuristically discretized (rounded to the closer integer values).
  - ✓ Approximatede solution if integer variables take large values
  - ✓ Possible loss of optimality
  - ✓ Possible loss of feasibility
- Metaheuristic methods are an alternative to classical optimization methods (genetic algorithms, simulated annealing, etc.)

Discretization: loss of feasibility

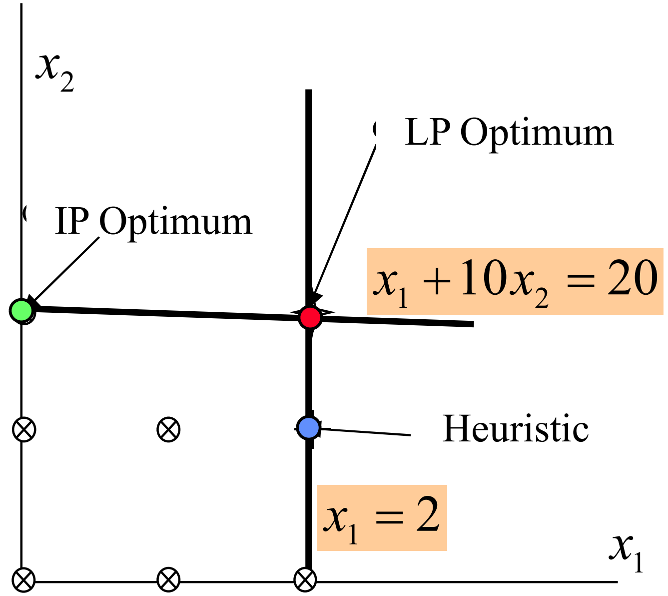
$$\max \quad x_2$$
$$-x_1 + x_2 \le 0.5$$
$$x_1 + x_2 \le 3.5$$
$$x_1, x_2 \ge 0 \quad x_i \in \mathbf{Z}$$



- LP solution: (1.5,2)
- Discretized solutions: (1,2) or (2,2), are infeasible
- Optimal integer solutions: (1,1) or (2,1)

Discretization: loss of optimality

$$\max \; x_1 + 5x_2$$
$$x_1 + 10x_2 \le 20$$
$$x_1 \le 2$$
$$x_1, x_2 \ge 0 \quad x_i \in \mathbf{Z}$$



- LP solution: (2,9/5)
- Discretized solution: (2,1)
- Optimal integer solution: (0,2)

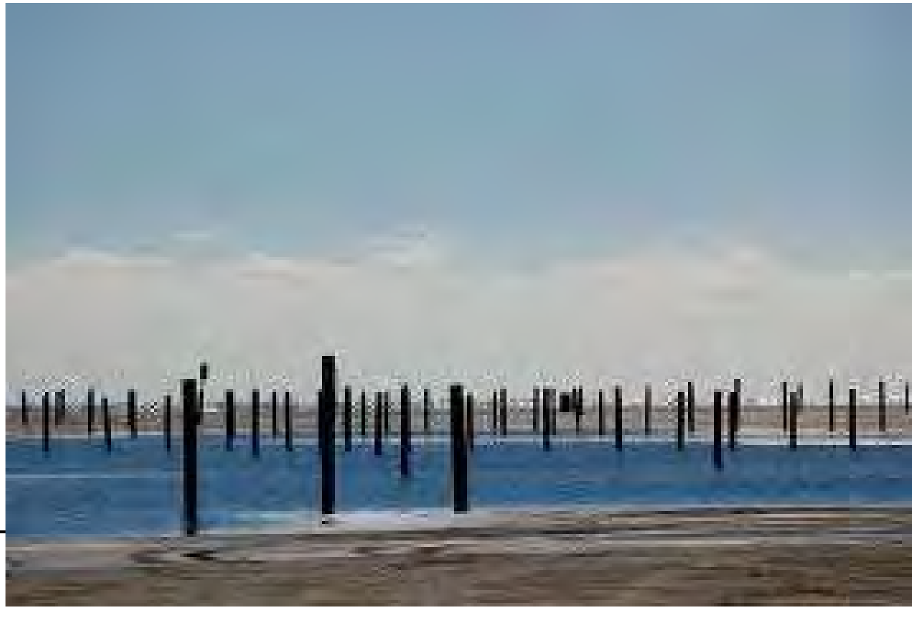
Exhaustive enumeration

- It is not possible, given that the number of solutions grows exponentially

- In a BIP problem of n variables, there 2^n possible solutions

Solution methods

- **Loss of convexity** of the feasible region. Interior points can not be expressed as a linear combination of the extreme points.
- **Loss of the mathematical power associated with continuous variables** (derivatives, optimality conditions, sensitivities, etc.).
- The solution of an MIP problem is more difficult than an LP problem's. It requires **more computation time and more memory**.



CONTENT

- ❑ INTRODUCTION
- ❑ SOLUTION METHODS
- ➢ BRANCH AND BOUND
- ❑ DUALITY (master)
- ❑ PREPROCESSING (master)
- ❑ BRANCH AND CUT METHOD (master)

Concepts

- Implicit enumeration of all the feasible integer solutions.
- Use the principle of divide and conquer.
  - ✓ Divide (branch) the set of integer solutions in decreasing disjunctive subsets.
  - ✓ Determine (bound) the value of the best solution of the subset.
    - In a maximization problem, a lower bound of the optimal MIP solution is the greatest feasible integer solution found so far
    - In a maximization problem, an upper bound of the optimal MIP solution is the optimal RMIP solution
  - ✓ Prune (delete) the branch of the tree if the bound says that this branch can not contain the optimal solution

Procedure (i)

1. Initialization

- ✓ Initialize the lower bound of the o.f. $z^* = -\infty$, in maximization problems
- ✓ Solve the problem relaxation (usually linear relaxation). This is the root node.
- ✓ Apply branch or prune and optimality criterion to the complete problem.
- ✓ If the problem can not be labeled as pruned, begin a complete iteration

2. Iteration

- ✓ Branch
  - ❑ Select one of the nodes among the non-explored ones (remaining nodes). See selection criteria
  - ❑ Select one integer variable with continuous value in the optimal solution of the relaxed problem. See selection criteria

Procedure (ii)

- Be x_j^* the optimal value of the relaxed problem
- Branch in two branches, including the constraints $\begin{cases} x_j \le \lfloor x_j^* \rfloor \\ x_j \ge \lfloor x_j^* \rfloor + 1 \end{cases}$

  being $\lfloor x_j^* \rfloor$ integer part of x_j^*
- For binary variables, the value is fixed to 0 or 1. Each binary variable can only be branched once.
- Each time a constraint is added (sensitivity analysis by dual simplex method). The primal problem becomes infeasible. The dual problem becomes feasible but not optimal.
- Each branch deletes the optimal solution of the previous problem (ancestor).

✓ Bounding

- For each node determines the o.f. z

Procedure (ii)

- ✓ Prune
  - ❑ Try to prune branches of the tree. Apply the following pruning criteria for a maximization problem:
    1. Solution worse than the current integer solution $z \leq z^*$, being z^* the value of the o.f. for the current integer solution. Prune the branch.
    2. Integer solution is better than the current one $z > z^*$. $z^* = z$ New current integer solution

       Apply criterion 1 to all the branches not pruned with the current integer solution.
    3. Infeasible. Prune the branch.

3. Optimality criterion
   - ✓ Stop when there are no problems not analyzed. The current integer solution is the optimal one.
   - ✓ If not, do another iteration.

Example

$$
\begin{array}{llllll}
\max & z = 4x_1 - 2x_2 + 7x_3 - x_4 \\
x_1 & & +5x_3 & & \leq 10 \\
x_1 & +x_2 & -x_3 & & \leq 1 \\
6x_1 & -5x_2 & & & \leq 0 \\
-x_1 & & +2x_3 & -2x_4 & \leq 3 \\
x_j \geq 0 & & j = 1,\ldots,4 \\
x_j \text{ integer} & & j = 1,\ldots,3
\end{array}
$$

Solution procedure (i)

1. Initialization
   - ✓ Solve the relaxed problem $z = 14.25$ $(x_1, x_2, x_3, x_4) = (1.25, 1.5, 1.75, 0)$
2. Iteration 1
   - ✓ Branch with the first continuous variable that had to be integer x_1 .
   - ✓ Branch 1: $x_1 \le 1$ $z = 14.2$ $(x_1, x_2, x_3, x_4) = (1, 1.2, 1.8, 0)$
     Any descendant solution will have $z \le 14.2$
   - ✓ Branch 2: $x_1 \ge 2$ infeasible. Prune the branch
3. Iteration 2
   - ✓ Branch with the first continuous variable that had to be integer, x_2 .
   - ✓ Branch 3: $x_1 \le 1$, $x_2 \le 1$ $z = 14.1\widehat{6}$ $(x_1, x_2, x_3, x_4) = (0.8\widehat{3}, 1, 1.8\widehat{3}, 0)$
     Any descendant solution will have $z \le 14.1\widehat{6}$
   - ✓ Branch 4: $x_1 \le 1$, $x_2 \ge 2$ $z = 12.1\widehat{6}$ $(x_1, x_2, x_3, x_4) = (0.8\widehat{3}, 2, 1.8\widehat{3}, 0)$
     Any descendant solution will have $z \le 12.1\widehat{6}$

Solution procedure (ii)

4. Iteration 3

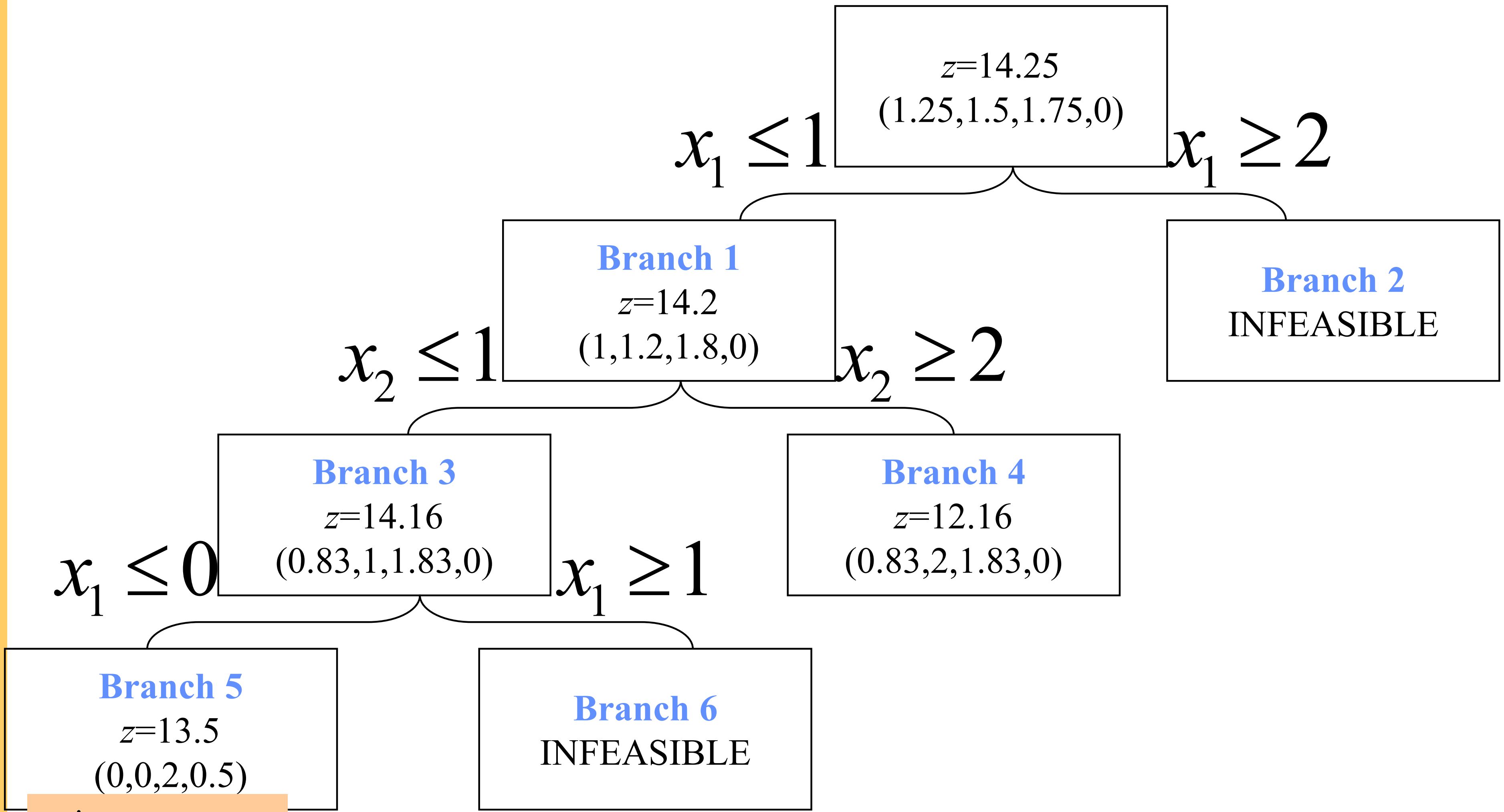
- ✓ Select branch 3 for having the best objective function.
- ✓ Branch with variable x_1 .
- ✓ Branch 5: $x_1 \le 1$, $x_2 \le 1$, $x_1 \le 0$ $\quad z = 13.5 \quad (x_1, x_2, x_3, x_4) = (0, 0, 2, 0.5)$

  First integer solution of an MIP problem $\quad z^* = 13.5$
- ✓ Branch 6: $x_1 \le 1$, $x_2 \le 1$, $x_1 \ge 1$ infeasible
- ✓ Branch 4 can be pruned given that its o.f. is lower (in maximization) than the current integer solution.

5. Optimality criterion

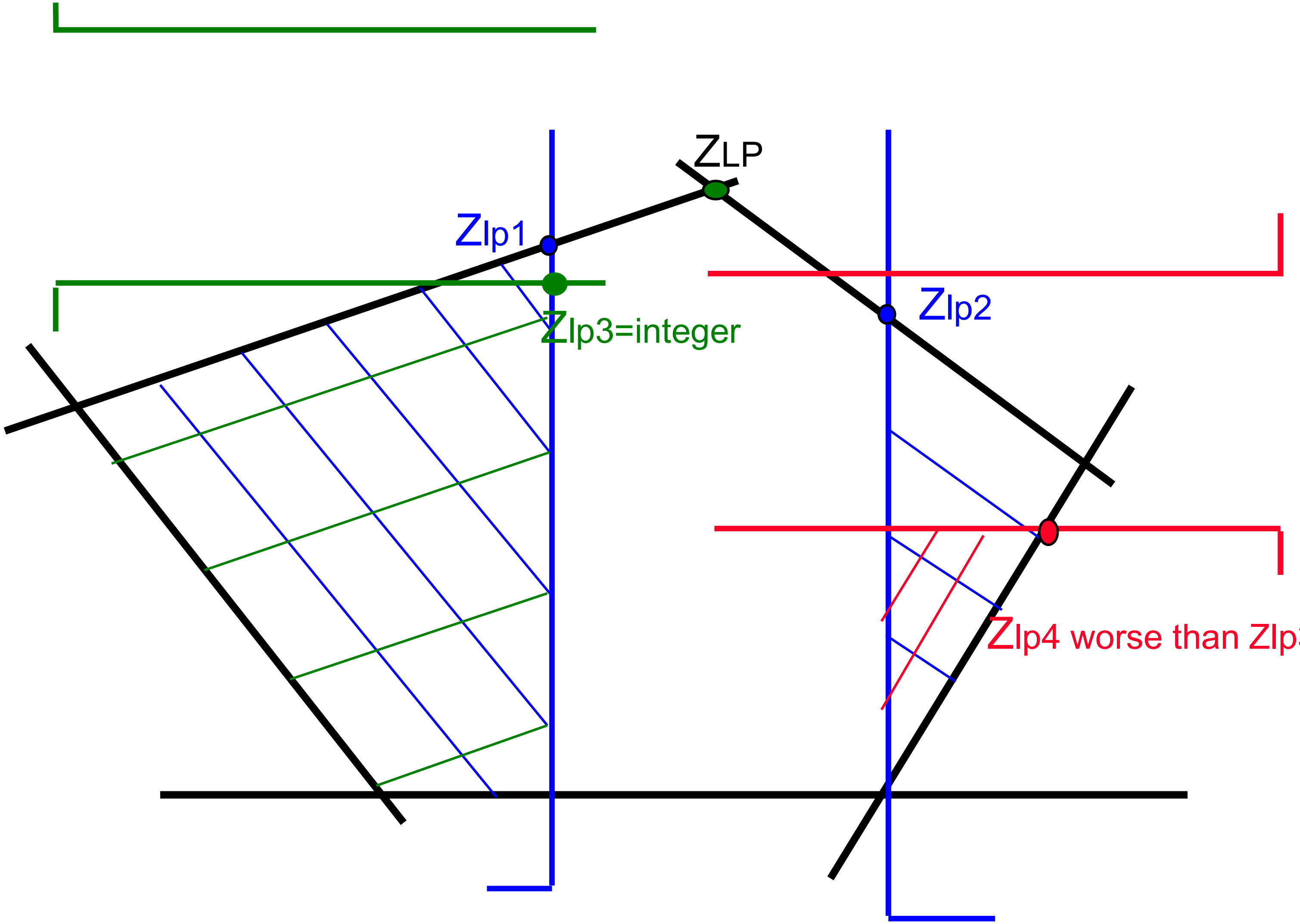
- ✓ Optimal solution given that there is no unexplored branches.

Tree



$z^* = 13.5$ $(x_1, x_2, x_3, x_4) = (0, 0, 2, 0.5)$

Geometrical interpretation



Linear problem (LP). Example 1

$$\max_{x,y} 3x + 2y$$

$$x + y \leq 11$$

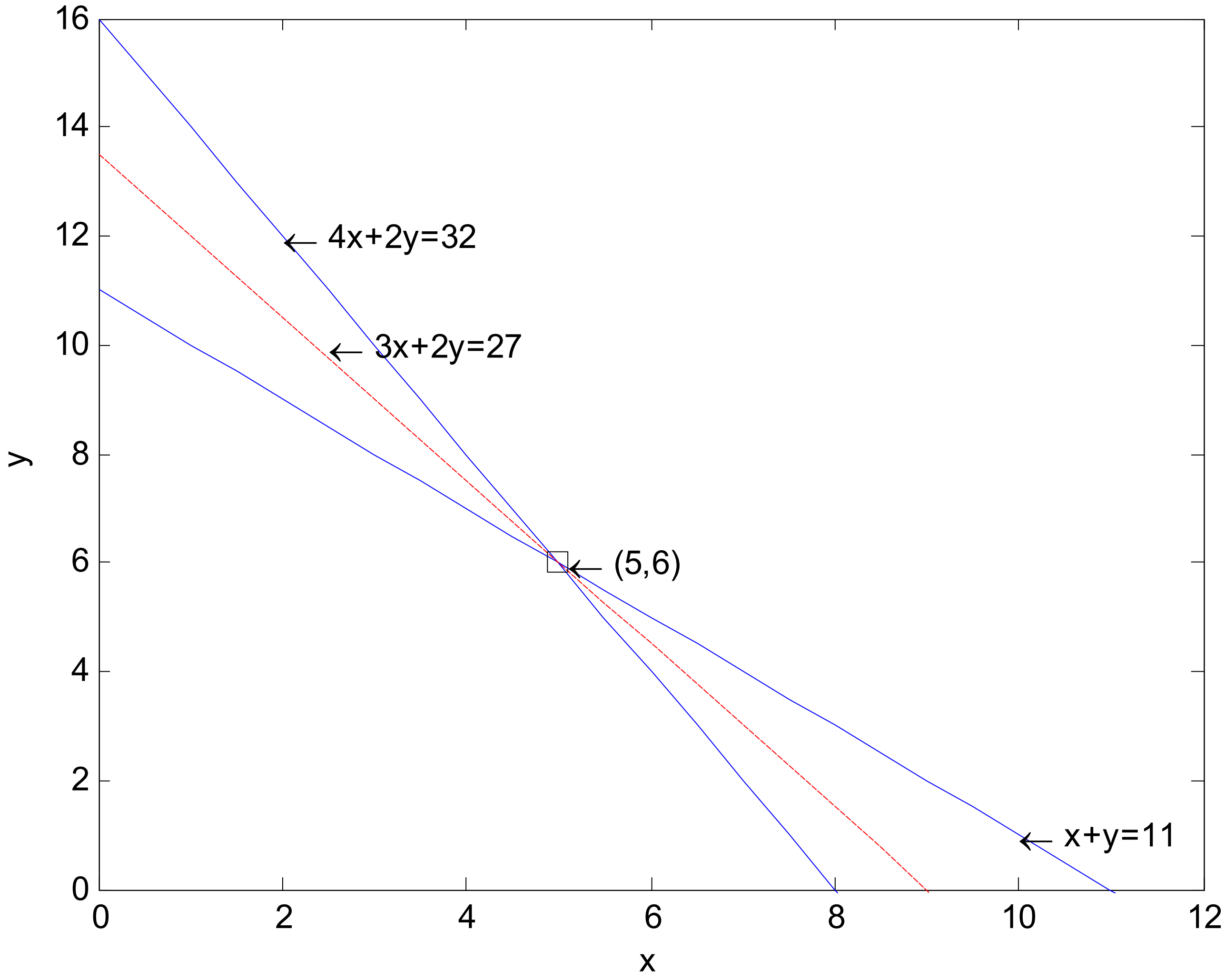
$$4x + 2y \leq 32$$

$$x, y \geq 0$$

LP solution

$$z = 27$$

$$(x^*, y^*) = (5,6)$$



Pure integer problem (PIP). Example 1

$$\max_{x,y} 3x + 2y$$

$$x + y \leq 11$$

$$4x + 2y \leq 32$$

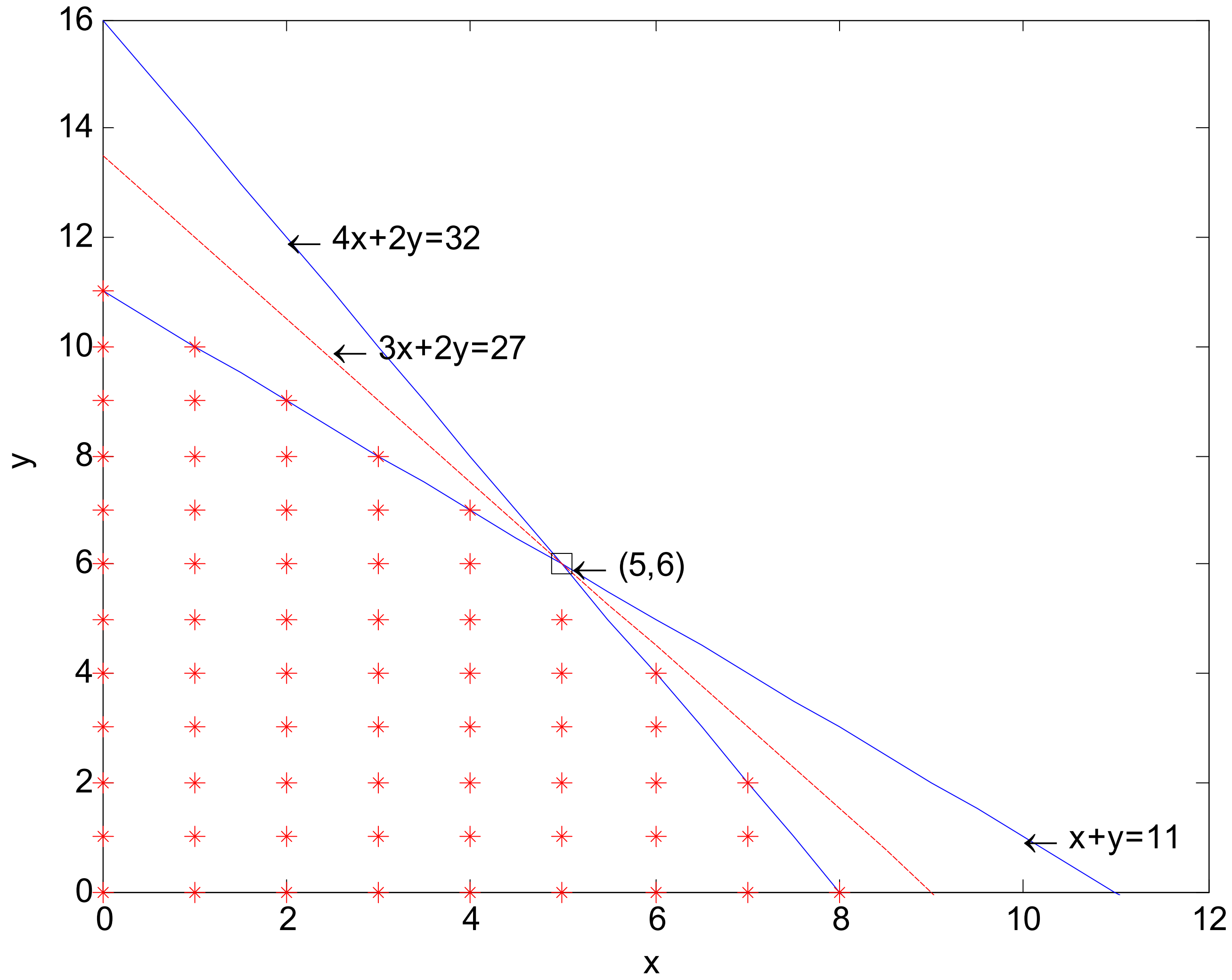
$$x, y \geq 0$$

$$x, y \in ¢^{+}$$

PIP solution

$$z = 27$$

$$(x^*, y^*) = (5,6)$$



Pure integer problem (PIP). Linear relaxation. Example 2

$$\max_{x,y} 3x + 2y$$

$$x + y \leq 11.5$$

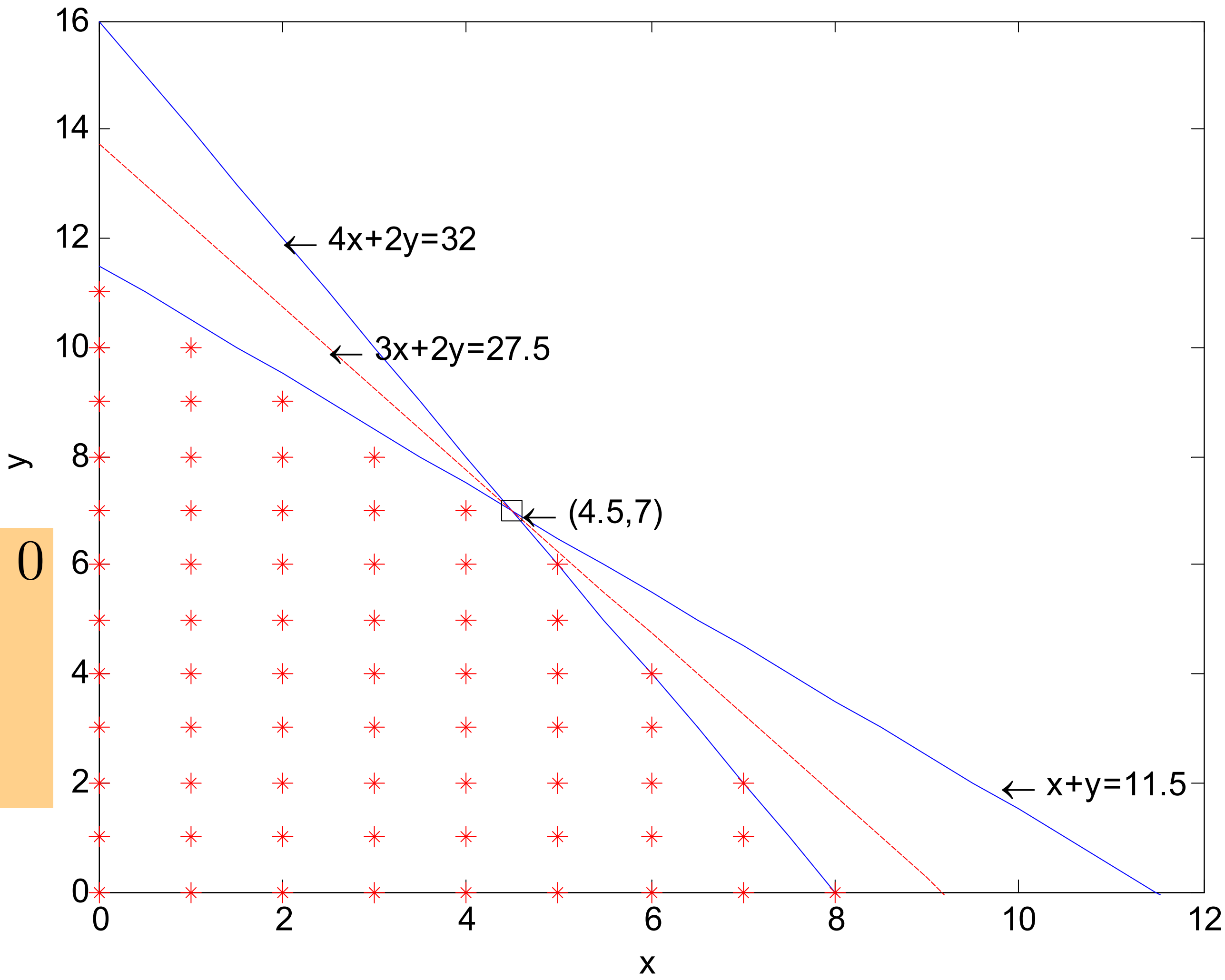
$$4x + 2y \leq 32$$

$$x, y \geq 0$$

LP relaxation. Problem 0

$$z = 27.5$$

$$(x^*, y^*) = (4.5, 7)$$



Pure integer problem (PIP). Branch and Bound. Example 2

$$\max_{x,y} 3x + 2y$$
$$x + y \leq 11.5$$
$$4x + 2y \leq 32$$
$$x \leq 4$$
$$x, y \geq 0$$

LP Problem 1

$z = 27$

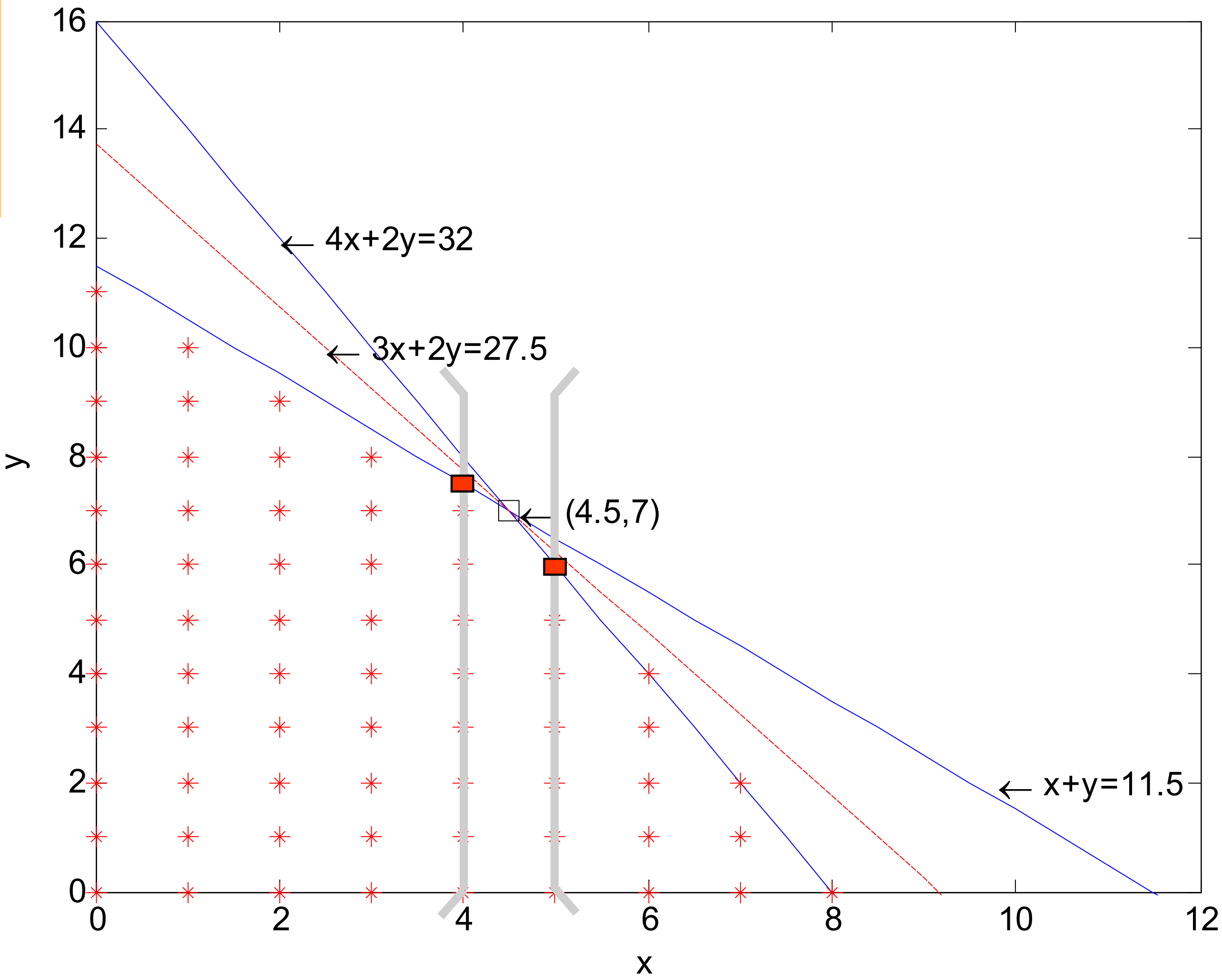
$(x^*, y^*) = (4, 7.5)$

$$\max_{x,y} 3x + 2y$$
$$x + y \leq 11.5$$
$$4x + 2y \leq 32$$
$$x \geq 5$$
$$x, y \geq 0$$

LP Problem 2

$z = 27$

$(x^*, y^*) = (5, 6)$



Pure integer problem (PIP). Branch and Bound. Example 2

$$\max_{x,y} 3x + 2y$$
$$x + y \leq 11.5$$
$$4x + 2y \leq 32$$
$$x \leq 4$$
$$y \geq 8$$
$$x, y \geq 0$$

LP Problem 3

$z = 26.5$

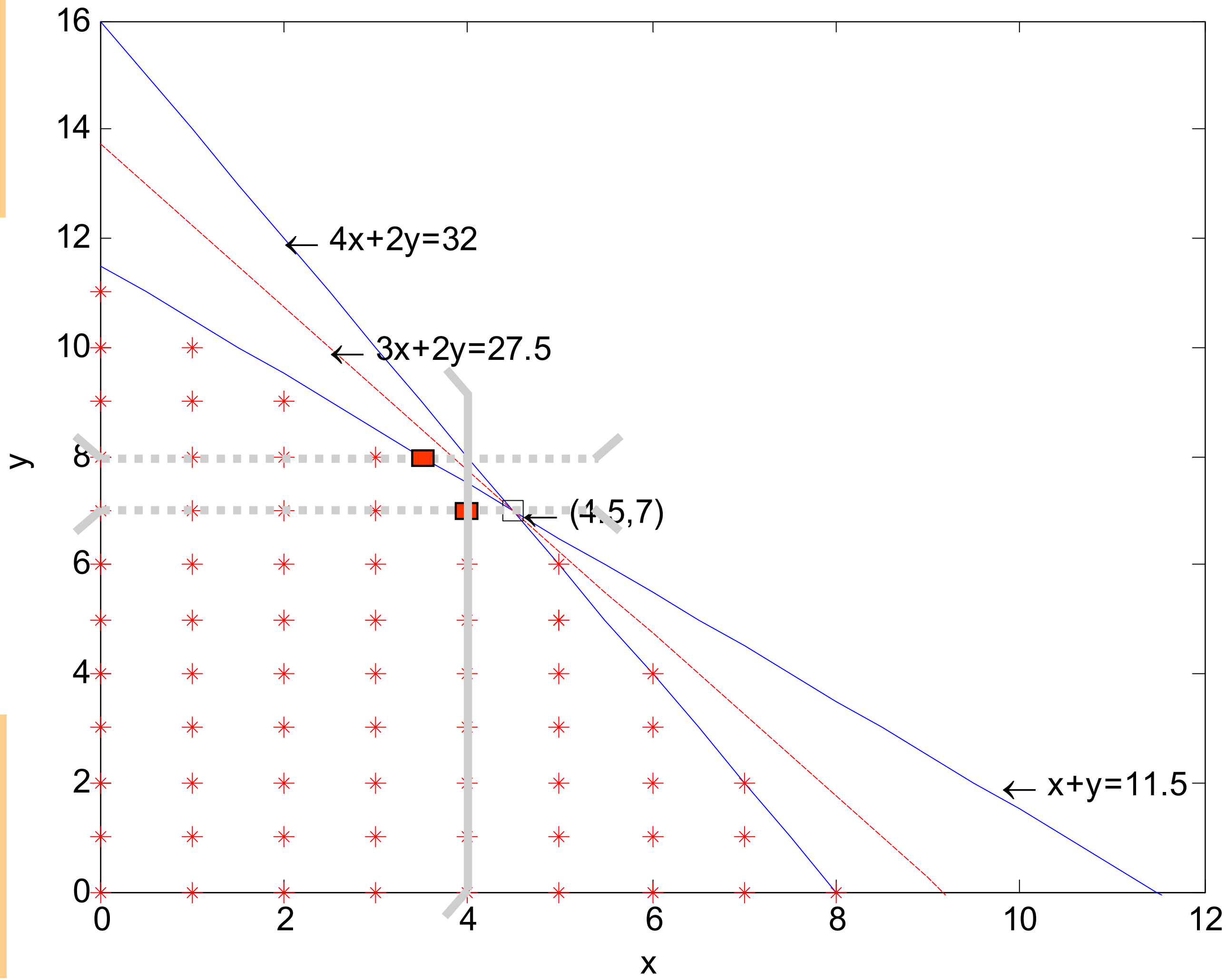
$(x^*, y^*) = (3.5, 8)$

$$\max_{x,y} 3x + 2y$$
$$x + y \leq 11.5$$
$$4x + 2y \leq 32$$
$$x \leq 4$$
$$y \leq 7$$
$$x, y \geq 0$$

LP Problem 4

$z = 26$

$(x^*, y^*) = (4, 7)$



Pure integer problem (PIP). Example 2

$$\max_{x,y} 3x + 2y$$

$$x + y \leq 11.5$$

$$4x + 2y \leq 32$$

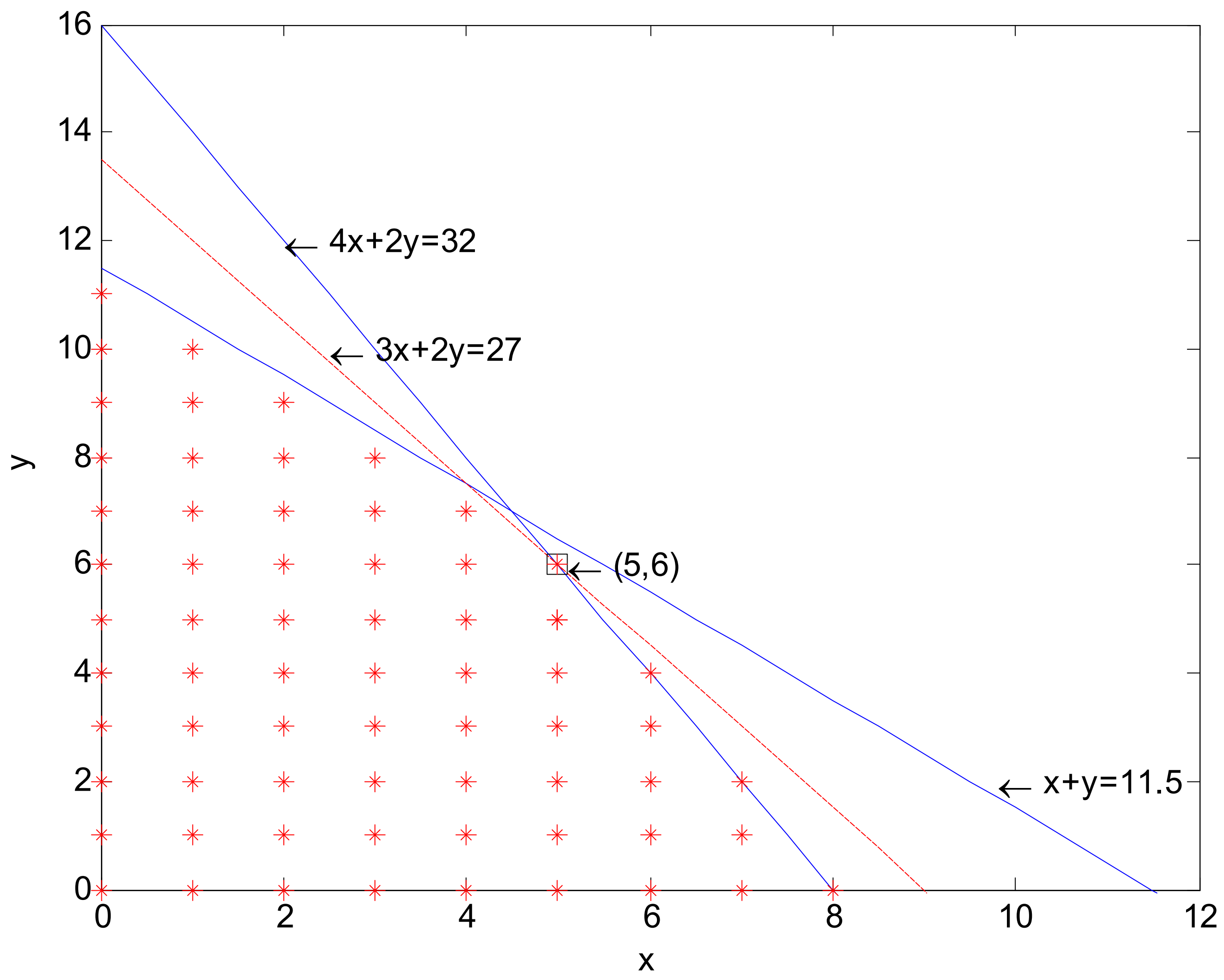
$$x, y \geq 0$$

$$x, y \in ¢^{+}$$

PIP solution

$$z = 27$$

$$(x^*, y^*) = (5,6)$$



Basic strategies

1. Emphasis in feasibility

- ✓ Branch and explore the tree fixing recursively the continuous variables closer to their integer value in the selected node (*greedy strategy*)

2. Emphasis in optimality

- ✓ Assuming that an integer solution is available, prove that is optimal. Select the variables which have strong impact in the objective function to discard branches of the tree as soon as possible.

Strategy implementation

1. Select a variable to branch
   - ✓ Firstly found
   - ✓ Lower or upper integer infeasibility

2. Select a branch to solve
   - ✓ The most recent. Good to re-optimize with the simplex method.
   - ✓ The closest or remotest objective function (worst or best bound)

Relaxation of the pruning criterion

- It is the difference between solving the problem with an optimal solution within a known tolerance or NOT solving the problem. Crucial for solving real problems.
- Stopping criterion without fully exploring the tree.

  Pruning criterion (for maximization) within a certain tolerance of a better (but not significantly better) non-integer solution than the current integer solution

  - ✓ Relative $z^* \leq z \leq z^*(1+\alpha)$
  - ✓ Absolute $z^* \leq z \leq z^* + \beta$

  α (relative tolerance error, e.g., 10^{-3}) **OptCR**

  β (absolute tolerance error), **OptCA**

CONTENT

- INTRODUCTION
- SOLUTION METHODS
- BRANCH AND BOUND
- ➢ DUALITY (master)
- PREPROCESSING (master)
- BRANCH AND CUT METHOD (master)

Pure integer problem (PIP). Dual variables

- We know how to obtain the dual variables of an LP problem. They are calculated at the same time as the optimal linear solution
- But we do not know how to obtain these dual variables in an MIP problem because we have solved many LP problems for reaching the optimal integer solution

Pure integer problem (PIP). Dual variables. Example 2

$$\max_{x,y} 3x + 2y$$

$$x + y \leq 11.5$$

$$4x + 2y \leq 32$$

$$x, y \geq 0$$

$$x, y \in ¢^{+}$$

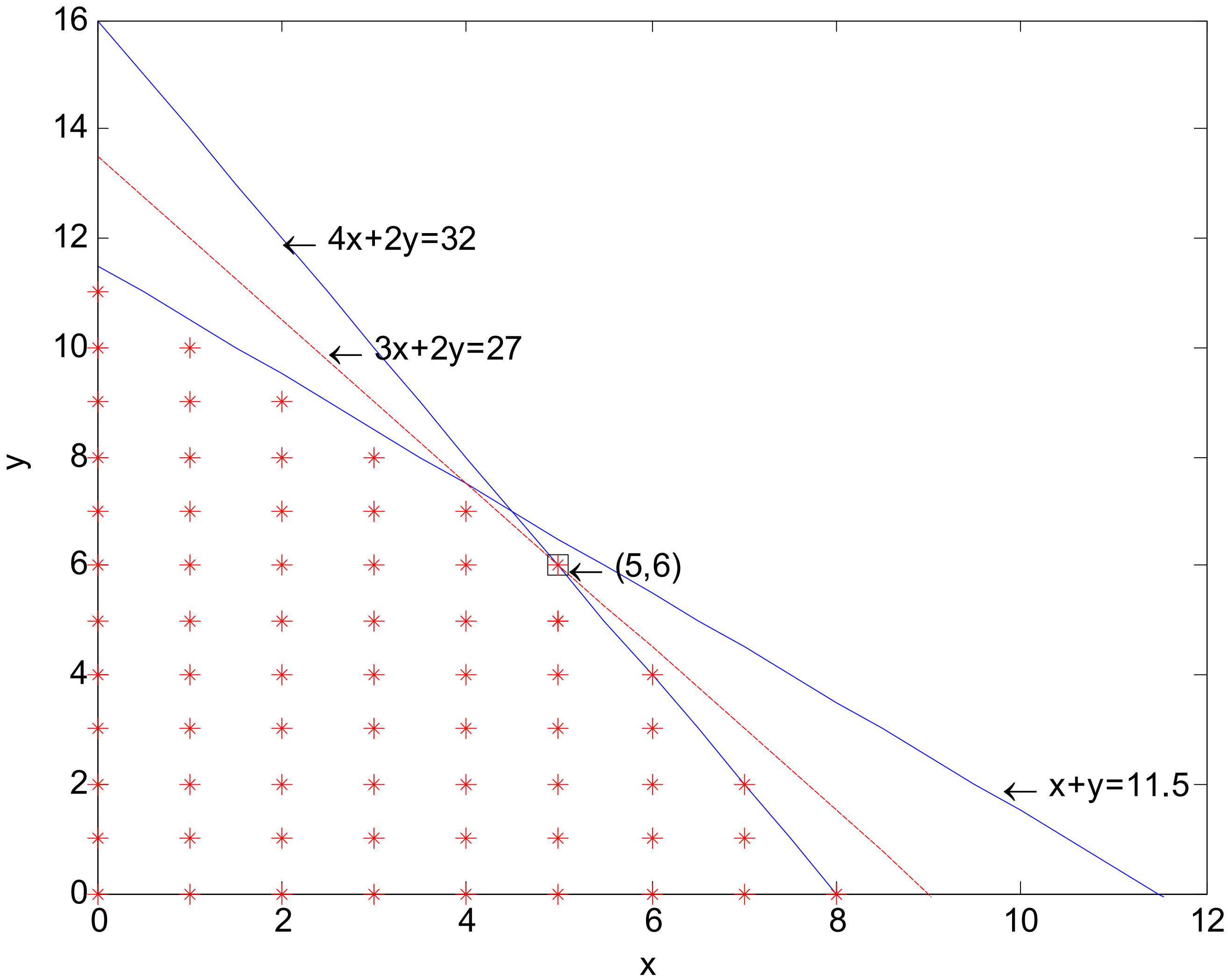
PIP solution

$$z = 27$$

$$(x^*, y^*) = (5,6)$$

$$(\pi_1^*, \pi_2^*) = (0,0)$$

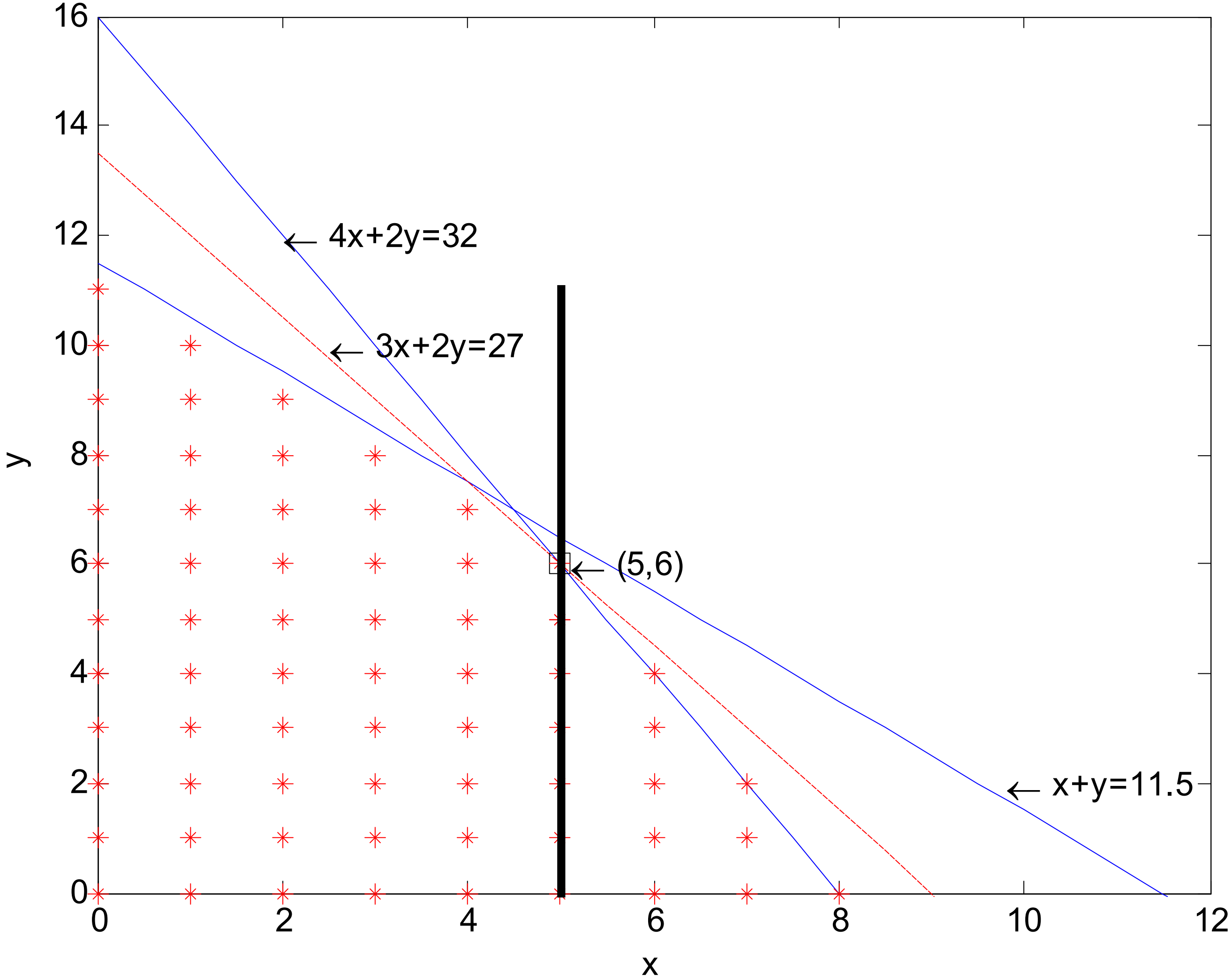
Obtain dual variables manually (re-solving the model)



Pure integer problem (PIP). Dual variables. Example 2

$$\max_{x,y} 3x + 2y$$
$$x + y \leq 11.5$$
$$4x + 2y \leq 32$$
$$x = 5$$
$$x, y \geq 0$$

Let's try fixing one variable



LP solution

$$z = 27$$
$$(x^*, y^*) = (5,6)$$
$$(\pi_1^*, \pi_2^*) = (0,1)$$

Pure integer problem (PIP). Dual variables. Example 2

$$\max_{x,y} 3x + 2y$$

$$x + y \leq 11.5$$

$$4x + 2y \leq 32$$

$$y = 6$$

$$x, y \geq 0$$

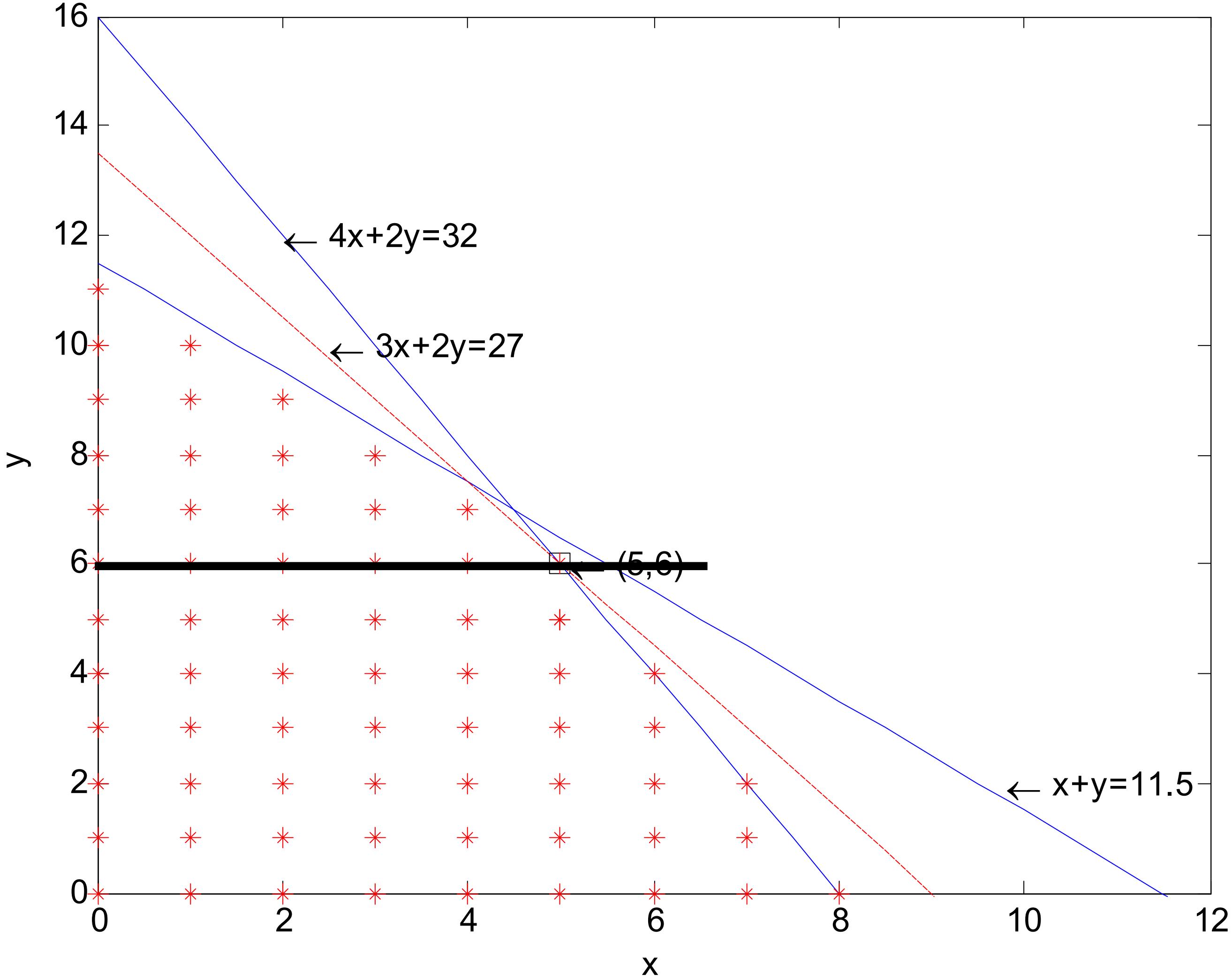
LP solution

$$z = 27$$

$$(x^*, y^*) = (5,6)$$

$$(\pi_1^*, \pi_2^*) = (0, 0.75)$$

Let's try fixing the other variable



Pure integer problem (PIP). Dual variables. Example 2

$$\max_{x,y} 3x + 2y$$

$$x + y \leq 11.5$$

$$4x + 2y \leq 32$$

$$x = 5$$

$$y = 6$$

$$x, y \geq 0$$

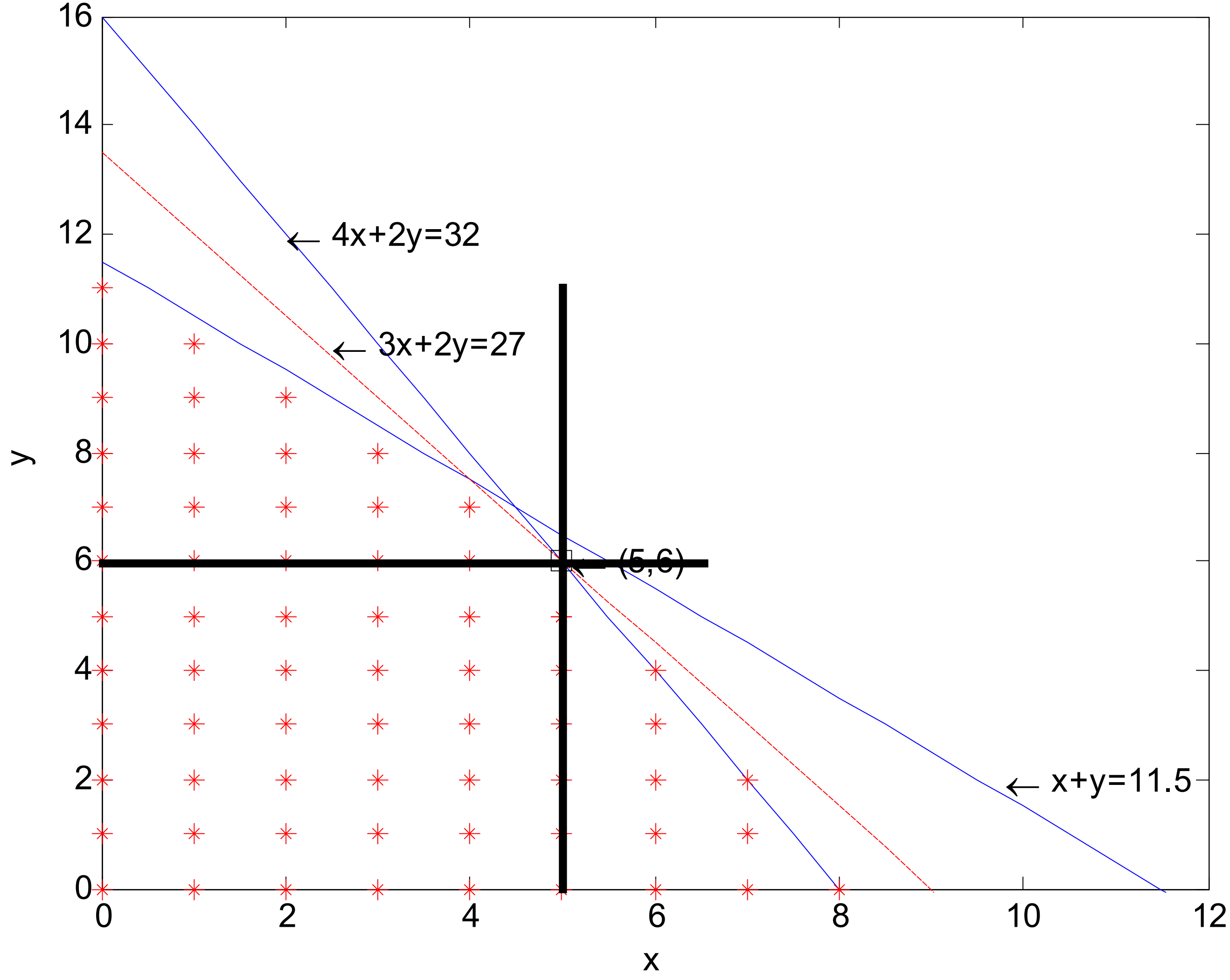
LP solution

$$z = 27$$

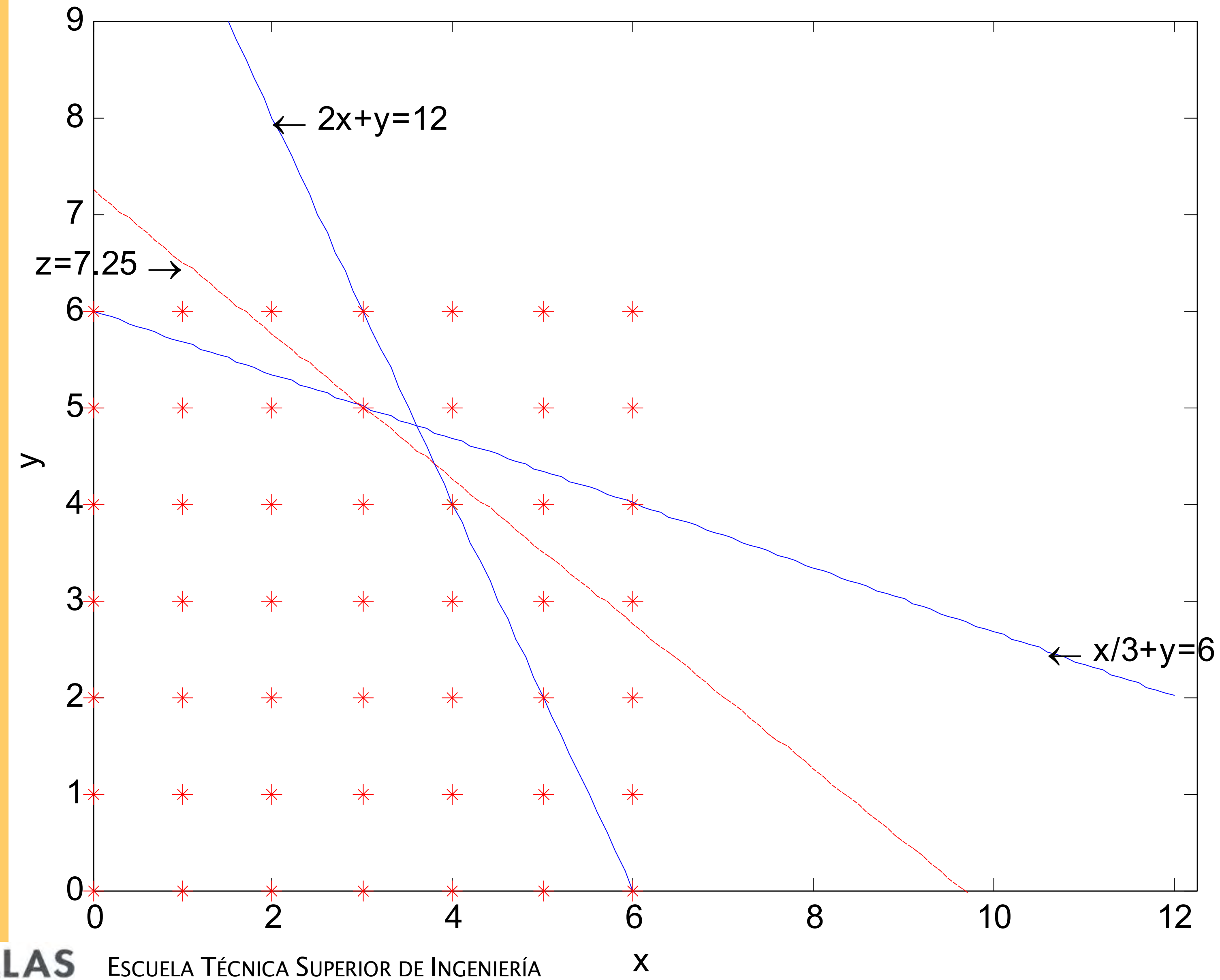
$$(x^*, y^*) = (5,6)$$

$$(\pi_1^*, \pi_2^*) = (0,0)$$

Let's try fixing both variables



Pure integer problem (PIP). Example 3



$$\max_{x,y} z = \frac{3}{4}x + y$$

$$\frac{x}{3} + y \le 6 \qquad : \pi_1$$

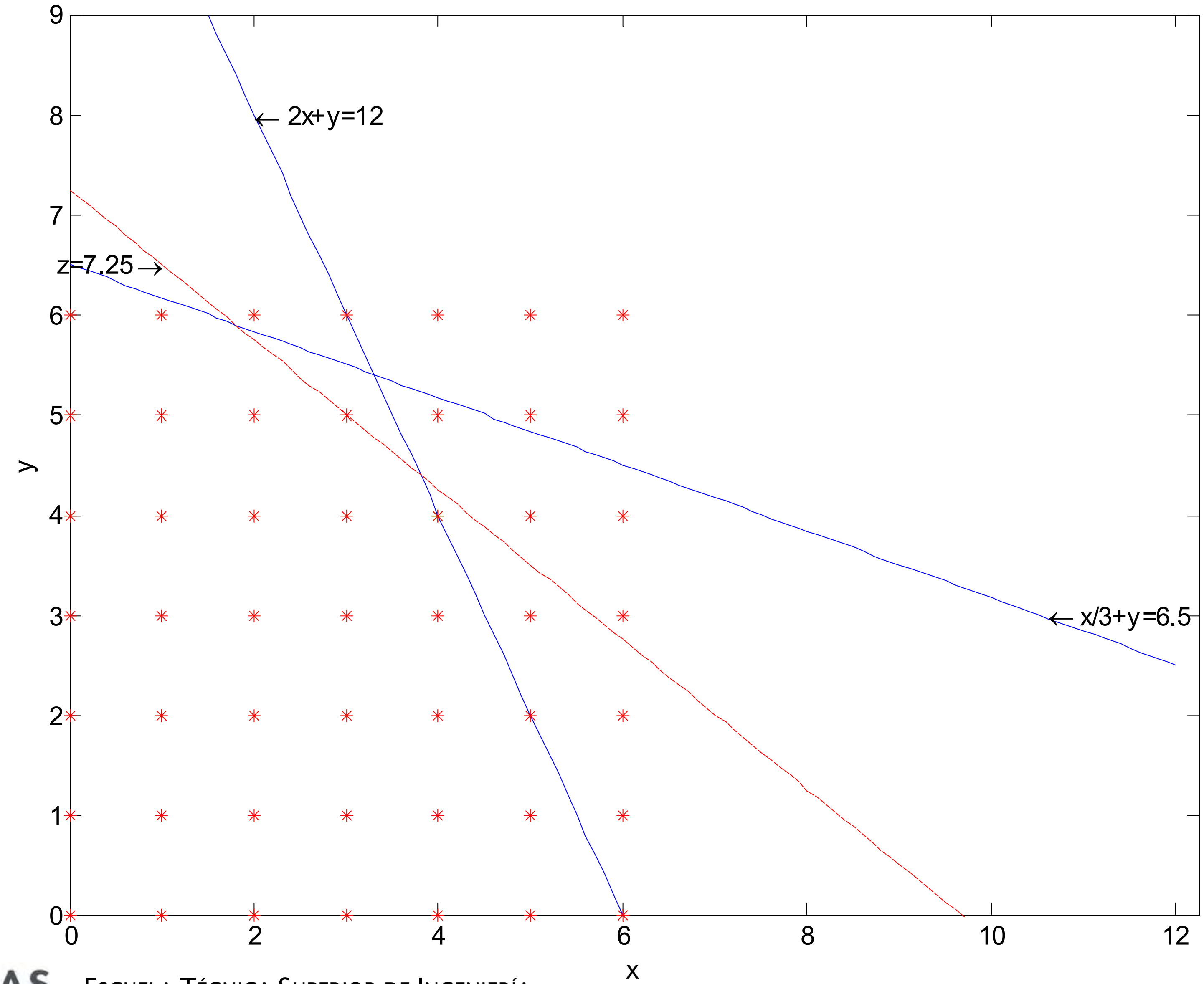
$$2x + y \le 12 \qquad : \pi_2$$

$$x, y \text{ integer}$$

$$(x, y) = (3,5)$$

$$(\pi_1, \pi_2) = (0,0)$$

Pure integer problem (PIP). Example 4



$$\max_{x,y} z = \frac{3}{4}x + y$$

$$\frac{x}{3} + y \le 6.5 \quad : \pi_1$$

$$2x + y \le 12 \quad : \pi_2$$

$$x, y \text{ integer}$$

$$(x, y) = (3,5)$$

$$(\pi_1, \pi_2) = (0,0)$$

In a PIP problem, dual variables = 0

Dual variables in an MIP problem

- In an LP problem, the complementarity slackness condition states:
  - ✓ If a constraint is binding dual variable may be different from 0
  - ✓ If a constraint is nonbinding dual variable is necessarily 0
- In an MIP problem
  - ✓ It is not clear how to obtain dual variables in an MIP problem
  - ✓ There must be those corresponding to the node that has provided the optimal solution in a B&B algorithm
  - ✓ In practice:
    - Fix all the integer variables to their optimal values and solve the corresponding LP problem to determine the dual variables

Duality gap

- Primal LP problem (*P*)

$$\min z = c^T x$$
$$Ax = b$$
$$x \geq 0$$

- Dual LP problem (*D*)

$$\max y_0 = b^T y$$
$$A^T y \leq c$$

- Strong duality property $z^* = c^T x^* = y_0^* = b^T y^*$
- Primal MIP problem (*P'*)

$$\min z' = c^T x$$
$$Ax = b$$
$$x \in \mathbb{Z}^{+}$$

$$z'^* \geq z^* = y_0^*$$

- Duality gap $z'^* - y_0^*$
- "Approximate" measure of the validity of dual variables
  - ✓ The larger the duality gap, the lower the validity range

CONTENT

- ❑ INTRODUCTION
- ❑ SOLUTION METHODS
- ❑ BRANCH AND BOUND
- ❑ DUALITY (master)
- ➢ PREPROCESSING (master)
- ❑ BRANCH AND CUT METHOD (master)

- T. Achterberg , R. Wunderling “*Mixed Integer Programming: Analyzing 12 Years of Progress*” in *Facets of Combinatorial Optimization. Festschrift for Martin Grötschel,* M. Jünger, G. Reinelt Eds. Springer 2013.
- R. Bixby, M. Fenelon, Z. Gu, E. Rothberg, and R. Wunderling, “MIP: theory and practice–closing the gap,” in *System Modelling and Optimization: Methods, Theory and Applications,* M. J. D. Powell and S. Scholtes, Eds. Boston: Kluwer Academic Publishers, 2000, vol. 174, p. 19–49.
- F. Vanderbeck and L. A. Wolsey, “Reformulation and decomposition of integer programs,” in *50 Years of Integer Programming 1958-2008,* M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 431–502.
- R. Bixby and E. Rothberg, “Progress in computational mixed integer programming—A look back from the other side of the tipping point,” *Annals of Operations Research, vol. 149, no. 1, pp. 37–41, Jan. 2007.*
- L. Wolsey, “Strong formulations for mixed integer programs: valid inequalities and extended formulations,” *Mathematical programming,* vol. 97, no. 1, p. 423–447, 2003.
- G. Morales-España, C.M. Correa-Posada, A. Ramos *Tight and Compact MIP Formulation of Configuration-Based Combined-Cycle Units* IEEE Transactions on Power Systems 31 (2), 1350-1359, March 2016 10.1109/TPWRS.2015.2425833
- G. Morales-España, J.M. Latorre, and A. Ramos *Tight and Compact MILP Formulation for the Thermal Unit Commitment Problem* IEEE Transactions on Power Systems 28 (4): 4897–4908, Nov 2013 10.1109/TPWRS.2012.2222938
- G. Morales-España, J.M. Latorre, and A. Ramos *Tight and Compact MILP Formulation of Start-Up and Shut-Down Ramping in Unit Commitment* IEEE Transactions on Power Systems 28 (2): 1288-1296, May 2013 10.1109/TPWRS.2012.2222938

Preprocessing

- Techniques focused on reducing substantially the dimensions or strengthening the problem formulation. Crucial for solving MIP problems
  - ✓ Two formulations of an integer problem are 0-1 *equivalent* if they have the same integer solutions.
  - ✓ Given two formulations of an integer problem, one is *stronger* (*tighter*) than the other if the feasible region of the linear relaxation is strictly included in the feasible region of the other. Consequently, feasible integer solutions are found quickly, and more branches can be pruned. Tighter formulations are sometimes based on increasing the problem size (*less compact*).
  - ✓ Measurement: integrality gap difference between the o.f. of the MIP and LP problems.

Preprocessing techniques

- General preprocessing
  - ✓ Bound strengthening
  - ✓ Elimination of redundant constraints
  - ✓ Variable assignment

- Mixed 0-1 preprocessing
  - ✓ Coefficient reduction

Bound strengthening

- Increase lower bounds and decrease upper bounds by inspection of the constraints
- Especially useful for integer variables
- May assign variables or detect infeasibilities

Bound strengthening (i)

$$\max 2x_1 + x_2 - x_3$$
$$5x_1 - 2x_2 + 8x_3 \le 15$$
$$8x_1 + 3x_2 - x_3 \ge 9$$
$$x_1 + x_2 + x_3 \le 6$$
$$0 \le x_1 \le 3$$
$$0 \le x_2 \le 1$$
$$1 \le x_3$$

← Bound strengthening of these constraints

- For the first constraint

$$5x_1 \le 2x_2 - 8x_3 + 15 \le 2\cdot 1 - 8\cdot 1 + 15 = 9 \Rightarrow \boxed{x_1 \le 9/5}$$
$$8x_3 \le -5x_1 + 2x_2 + 15 \le -5\cdot 0 + 2\cdot 1 + 15 = 17 \Rightarrow \boxed{x_3 \le 17/8}$$
$$-2x_2 \le -5x_1 - 8x_3 + 15 \le -5\cdot 0 - 8\cdot 1 + 15 = 7 \Rightarrow \boxed{x_2 \ge -7/2} \text{ superfluous bound}$$

- For the second constraint

$$8x_1 \ge -3x_2 + x_3 + 9 \ge -3\cdot 1 + 1 + 9 = 7 \Rightarrow \boxed{x_1 \ge 7/8}$$
$$3x_2 \ge -8x_1 + x_3 + 9 \ge -8\cdot 9/5 + 1 + 9 = -4.4 \Rightarrow \boxed{x_2 \ge -4.4} \text{ superfluous bound}$$
$$-x_3 \ge -8x_1 - 3x_2 + 9 \ge -8\cdot 9/5 - 3\cdot 1 + 9 = -8.4 \Rightarrow \boxed{x_3 \le 8.4} \text{ superfluous bound}$$

- For the first constraint

$$8x_3 \le -5x_1 + 2x_2 + 15 \le -5\cdot 7/8 + 2\cdot 1 + 15 = 101/8 \Rightarrow \boxed{x_3 \le 101/64}$$

➡

$$0.875 \le x_1 \le 1.8$$
$$0 \le x_2 \le 1$$
$$1 \le x_3 \le 1.578$$

Bound strengthening (ii)

- In the constraint

$$a_0 x_0 + \sum_{j=1}^{n} a_j x_j \le b$$
$$l_j \le x_j \le u_j$$

- If $\alpha_0 > 0$ then

$$x_0 \le \left(b - \sum_{j:a_j>0} a_j l_j - \sum_{j:a_j<0} a_j u_j \right) \Big/ a_0$$

- If $\alpha_0 < 0$ then

$$x_0 \ge \left(b - \sum_{j:a_j>0} a_j l_j - \sum_{j:a_j<0} a_j u_j \right) \Big/ a_0$$

Bound strengthening (iii)

- If a variable must be integer and their bounds aren't, then these bounds can be adjusted

$$x_j \in \mathbb{Z}^+$$

$$l_j \le x_j \le u_j \quad \Rightarrow \quad \lceil l_j \rceil \le x_j \le \lfloor u_j \rfloor$$

Elimination of redundant constraints (i)

- If a constraint is satisfied even in the "hardest" situation, then the constraint can be deleted
- Detect redundancies using variable bounds
- For $\leq$ constraints the variables with coefficients > 0 are fixed to their upper bound and the remaining ones to their lower bound
- Binary variables $\begin{matrix} 0 \leq x_1 \leq 1 \\ 0 \leq x_2 \leq 1 \end{matrix}$ $\begin{matrix} 3x_1 + 2x_2 \leq 6 \\ 3x_1 - 2x_2 \leq 3 \\ 3x_1 - 2x_2 \geq -3 \end{matrix}$ are redundant
- Constraints are usually redundant because of the variable assignment or of the bound strengthening

Elimination of redundant constraints (ii)

- The constraint

$$\sum_{j=1}^{n} a_j x_j \le b$$
$$l_j \le x_j \le u_j$$

Is redundant (always satisfied) if

$$\sum_{j:a_j>0} a_j u_j + \sum_{j:a_j<0} a_j l_j \le b$$

Is infeasible (never satisfied) if

$$\sum_{j:a_j>0} a_j l_j + \sum_{j:a_j<0} a_j u_j > b$$

In this case, the problem is infeasible without trying to solve it

Binary variable assignment (i)

- Identify binary variables that must be fixed to one of their values {0,1} given that for the other value there is no feasible solution
- If the value of a binary variable can't satisfy a constraint, even though the remaining variables take the most favorable values to achieve it, the binary variable must be fixed to its opposite value

being
$$\begin{aligned} 0 \le x_1 \le 1 \\ 0 \le x_2 \le 1 \\ 0 \le x_3 \le 1 \end{aligned}$$

$$\left.\begin{array}{llll} 3x_1 & & & \le 2 \\ 3x_1 & +x_2 & & \le 2 \\ 5x_1 & +x_2 & -2x_3 & \le 2 \end{array}\right\} \Rightarrow x_1 = 0$$

Binary variable assignment (ii)

- Procedure for $\leq$ constraints
  - ✓ Identify the binary variable with the largest positive coefficient. If the sum of this coefficient and any negative coefficient exceeds the constraint bound, the variable must be fixed to 0

$$\left.\begin{array}{llll} 3x_1 & & & \geq 2 \\ 3x_1 & +x_2 & & \geq 2 \\ 5x_1 & +x_2 & -2x_3 & \geq 2 \end{array}\right\} \Rightarrow x_1 = 1$$

$$x_1 + x_2 - 2x_3 \geq 1 \Rightarrow x_3 = 0$$

$$3x_1 + x_2 - 3x_3 \geq 2 \Rightarrow x_1 = 1, x_3 = 0$$

$$3x_1 - 2x_2 \leq -1 \Rightarrow x_1 = 0, x_2 = 1$$

- Chain reaction: the procedure will be repeated for the following variable with the largest coefficient

$$\begin{array}{lr} 3x_1 + x_2 - 2x_3 \geq 2 \Rightarrow x_1 = 1 & \\ x_1 + x_4 + x_5 \leq 1 & \Rightarrow x_4 = 0, x_5 = 0 \\ -x_5 + x_6 \leq 0 & \Rightarrow x_6 = 0 \end{array}$$

Coefficient reduction (i)

- Reduce the feasible solution of the LP problem without eliminating feasible solutions of the BIP problem by modifying the constraint coefficients
- Procedure for $\leq$ constraints $a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq b$
  1. Determine S = sum of positive a_j
  2. Choose any $a_j \neq 0$ such as $S < b + |a_j|$
     - It doesn't exist $\Rightarrow$ the constraint can't be further adjusted
     - $a_j > 0 \quad a'_j = S - b \quad b' = S - a_j \quad a_j = a'_j \quad b = b'$
     - $a_j < 0 \quad a'_j = S - b \quad a_j = a'_j$
  3. Go to 1

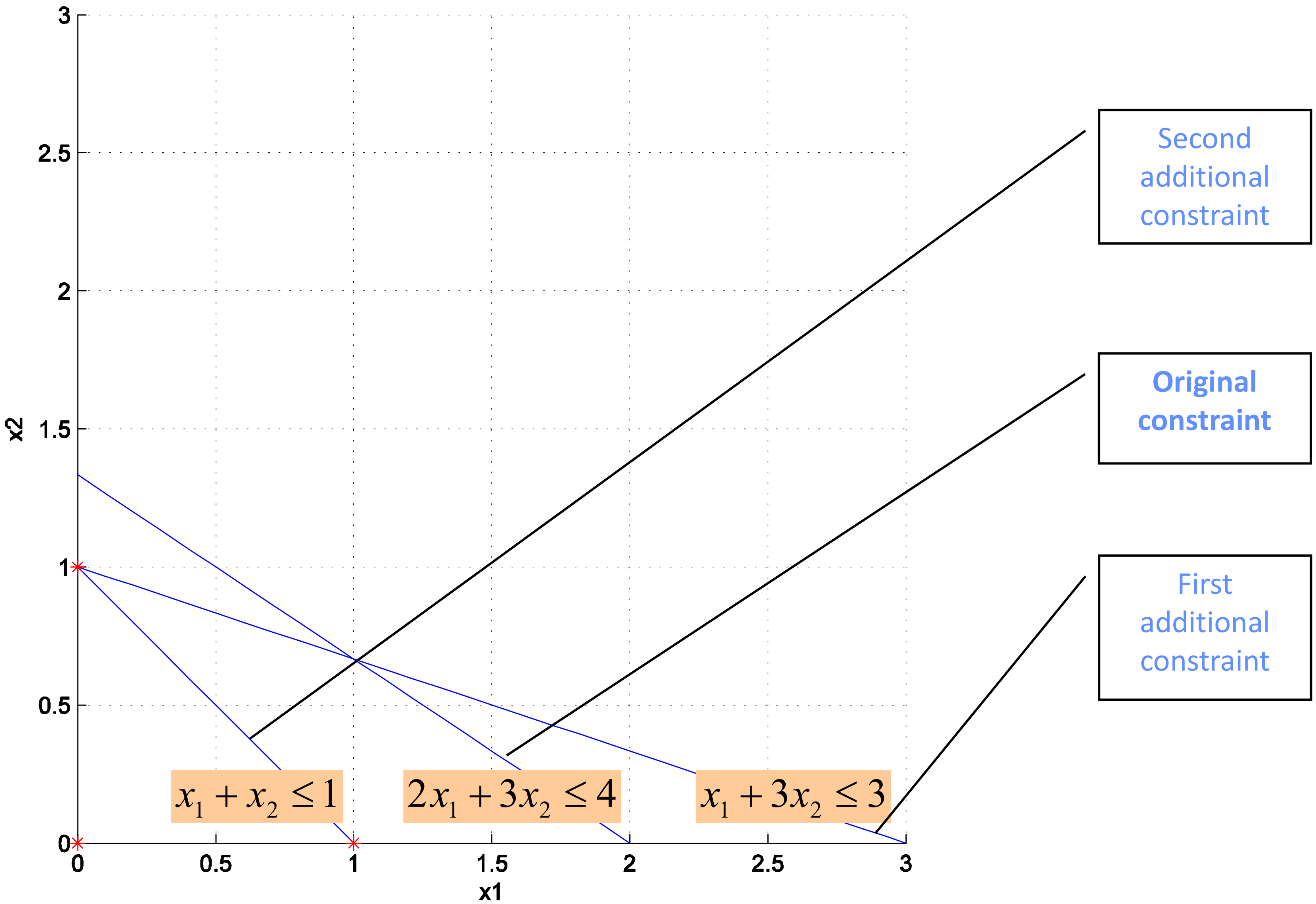
Coefficient reduction (ii)

$$2x_1 + 3x_2 \le 4$$
$$x_1, x_2 \in \{0,1\}$$

$$S = 2 + 3 = 5$$
$$a_1 \ne 0 \qquad 5 < 4 + 2$$
$$a_1' = 5 - 4 = 1 \quad b' = 5 - 2 = 3 \quad a_1 = 1 \quad b = 3 \quad x_1 + 3x_2 \le 3$$

$$S = 1 + 3 = 4$$
$$a_2 \ne 0 \qquad 4 < 3 + 3$$
$$a_2' = 4 - 3 = 1 \quad b' = 4 - 3 = 1 \quad a_2 = 1 \quad b = 1 \quad x_1 + x_2 \le 1$$

Coefficient reduction (iii)



Preprocessing by GUROBI

40 % reduction in rows, 38 % in columns and 30 % in nonzeros

```
--- StarNetLite_TEPM_Iceland.gms(15000) 986 Mb
---   1,167,736 rows  2,004,441 columns  8,140,314 non-zeroes
---   0 nl-code  0 nl-non-zeroes
---   7 discrete-columns
***   63,652 relaxed-columns WARNING
--- StarNetLite_TEPM_Iceland.gms(15000) 984 Mb
--- Executing GUROBI: elapsed 0:00:20.917
--- StarNetLite_TEPM_Iceland.gms(15000) 984 Mb  3 secs

Gurobi           24.6.1 r55820 Released Jan 18, 2016 WEI x86 64bit/MS Windows

Gurobi link license.
Gurobi library version 6.5.0
Reading parameter(s) from "C:\Users\aramos\Desktop\Aramos\TEPES\gurobi.opt"
>>  Method   2
>>  IntFeasTol 1e-9
>>  OptimalityTol 1e-9
>>  FeasibilityTol 1e-9
>>  IIS   1
>>  RINS     100
>>  DisplayInterval 1
>>  NumericFocus       1
>>  Kappa    1
>>  MarkowitzTol 0.999
>>  UseBasis 0
>>  CrossOver 0
>>  Names  0
>>  AggFill 0
>>  GomoryPasses 0
>>  Heuristics 0.001
>>  MipFocus 3
>>  *PreDual 0
>>  *PrePasses 3
>>  *PreSolve 2
>>  *PreSparsify 1
```

```
Finished reading from "C:\Users\aramos\Desktop\Aramos\TEPES\gurobi.opt"
Starting Gurobi...
Optimize a model with 1167735 rows, 2004440 columns and 8140308 nonzeros
Coefficient statistics:
  Matrix range     [7e-09, 1e+03]
  Objective range  [1e+00, 1e+00]
  Bounds range     [4e-07, 5e+00]
  RHS range        [1e-18, 1e+01]
Presolve removed 257457 rows and 497937 columns (presolve time = 2s) ...
Presolve removed 259520 rows and 500000 columns (presolve time = 2s) ...
Presolve removed 470416 rows and 710896 columns (presolve time = 3s) ...
Presolve removed 470616 rows and 711132 columns (presolve time = 4s) ...
Presolve removed 470666 rows and 753485 columns (presolve time = 5s) ...
Presolve removed 470705 rows and 753535 columns (presolve time = 6s) ...
Presolve removed 470705 rows and 753535 columns
Presolve time: 6.23s
Presolved: 697030 rows, 1250905 columns, 5645696 nonzeros
```

Preprocessing by GUROBI

- Before and after presolve can help you in detecting improvements in the formulation
- Allows to get the optimization problem after the presolve

```
ModelName = read("OriginalProblem.lp")
ModelNamePresolved = ModelName.presolve()
ModelNamePresolved.write("PresolvedProblem.lp")
```

CONTENT

- ❑ INTRODUCTION
- ❑ SOLUTION METHODS
- ❑ BRANCH AND BOUND
- ❑ DUALITY (master)
- ❑ PREPROCESSING (master)
- ➢ BRANCH AND CUT METHOD (master)

Cutting plane

- New constraint that reduces the feasible region of an LP problem without eliminating feasible solutions of an IP problem.
- Deducted validly from problem constraints

- Procedure of cutting plane method
  1. Initialize solving the relaxed LP problem
  2. If the optimal solution is integer, stop. If not, go to step 3
  3. Obtain a cutting plane that violates the current optimal solution
  4. Add the cutting plane to the constraint set and re-optimize. Go to step 2.

Cover-type cutting planes

- Consider any ≤ constraint with binary variables with all the coefficients nonnegative (knapsack-type constraint)
- Find a set of variables (minimal cover) such that
  - ✓ The constraint is violated if the cover variables are 1 and the remaining ones are 0
  - ✓ The constraint is satisfied if one of the cover variables is 0
- Structure of the cutting plane $\sum \text{cover variables} \leq N - 1$

  being N the number of cover variables

Original constraint → Cover-type cutting planes

$$6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 9$$
$$x_i \in \{0,1\}$$

$$\begin{cases} x_1 + x_2 + x_4 \leq 2 \\ x_2 + x_3 + x_4 \leq 2 \\ x_1 + x_3 \leq 1 \end{cases}$$

Gomory's cut (i)

- Let's see the problem
$$\min_x c^T x$$
$$Ax = b$$
$$x \in \mathbb{Z}^+$$

- Be x_i a non-integer variable. The rows of simplex tableau are:

$$x_B = B^{-1}(b - Nx_N) = B^{-1}b - B^{-1}Nx_N \qquad x_i = \hat{b}_i - \sum_{t \in x_N} y_{it} x_t$$

- The Gomory's cut is:

$$x_i \le \lfloor \hat{b}_i \rfloor - \sum_{t \in x_N} \lfloor y_{it} \rfloor x_t \qquad \sum_{t \in x_N} (y_{it} - \lfloor y_{it} \rfloor) x_t \ge \hat{b}_i - \lfloor \hat{b}_i \rfloor$$

  or also $$\sum_{t \in x_N} f_{it} x_t \ge f_i$$

  being f_i and f_{it} the fractional parts of $\hat{b}_i$ and y_{it}

- The excess variable of the cut is integer if variables are integer. This cut eliminates the previous non integer optimal solution

Gomory's cut (ii)

$$\max 4x_1 - x_2$$
$$7x_1 - 2x_2 \le 14$$
$$x_2 \le 3$$
$$2x_1 - 2x_2 \le 3$$
$$x_1, x_2 \ge 0 \text{ and integer}$$

$$\max 4x_1 - x_2$$
$$7x_1 - 2x_2 + h_1 = 14$$
$$x_2 + h_2 = 3$$
$$2x_1 - 2x_2 + h_3 = 3$$
$$x_1, x_2, h_1, h_2, h_3 \ge 0 \text{ and integer}$$

	X_1	X_2	HLG_1	HLG_2	HLG_3	
Z	0.000	0.000	9.143	17.286	0.000	179.857
X_2	0.000	1.000	0.000	1.000	0.000	3.000
HLG_3	0.000	0.000	-0.286	1.429	1.000	3.28
X_1	1.000	0.000	0.143	0.286	0.000	2.857

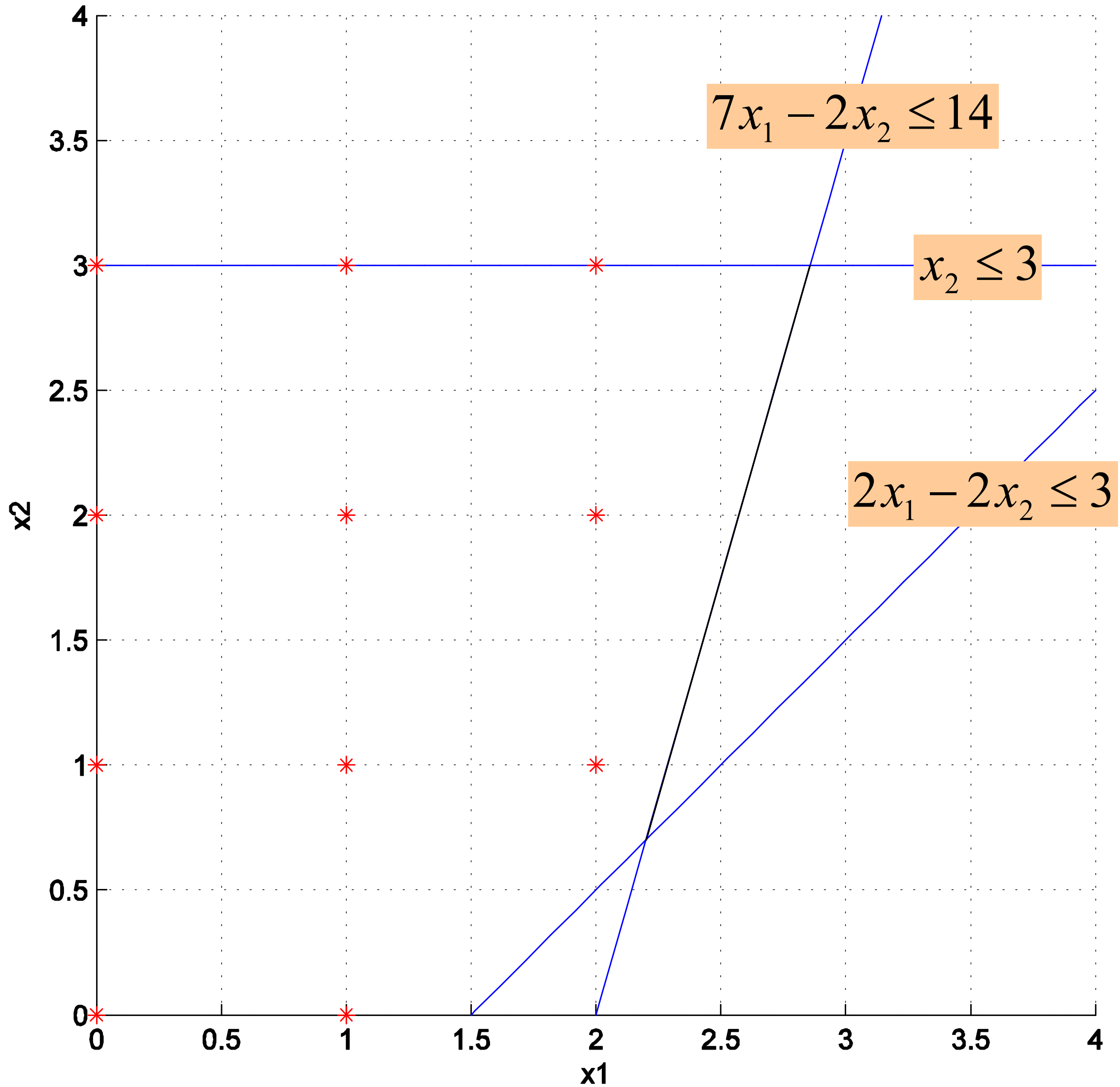
Introduce the cut $0.143h_1 + 0.286h_2 \ge 0.857$

	X_1	X_2	HLG_1	HLG_2	HLG_3	EXC_1	
Z	0.000	0.000	0.000	0.000	0.500	3.000	7.500
X_2	0.000	1.000	0.000	0.000	-0.500	1.000	0.500
HLG_2	0.000	0.000	0.000	1.000	0.500	-1.000	2.500
X_1	1.000	0.000	0.000	0.000	0.000	1.000	2.000
HLG_1	0.000	0.000	1.000	0.000	-1.000	-5.000	1.000

And then the cut $0.5h_3 \ge 0.5$

That produces the optimal solution

Gomory's cut (iii)



Gomory's cut (iv)

$$0.143h_1 + 0.286h_2 \geq 0.857$$

$$\frac{1}{7}(14 - 7x_1 + 2x_2) + \frac{2}{7}(3 - x_2) \geq \frac{6}{7}$$

$$-7x_1 \geq -14$$

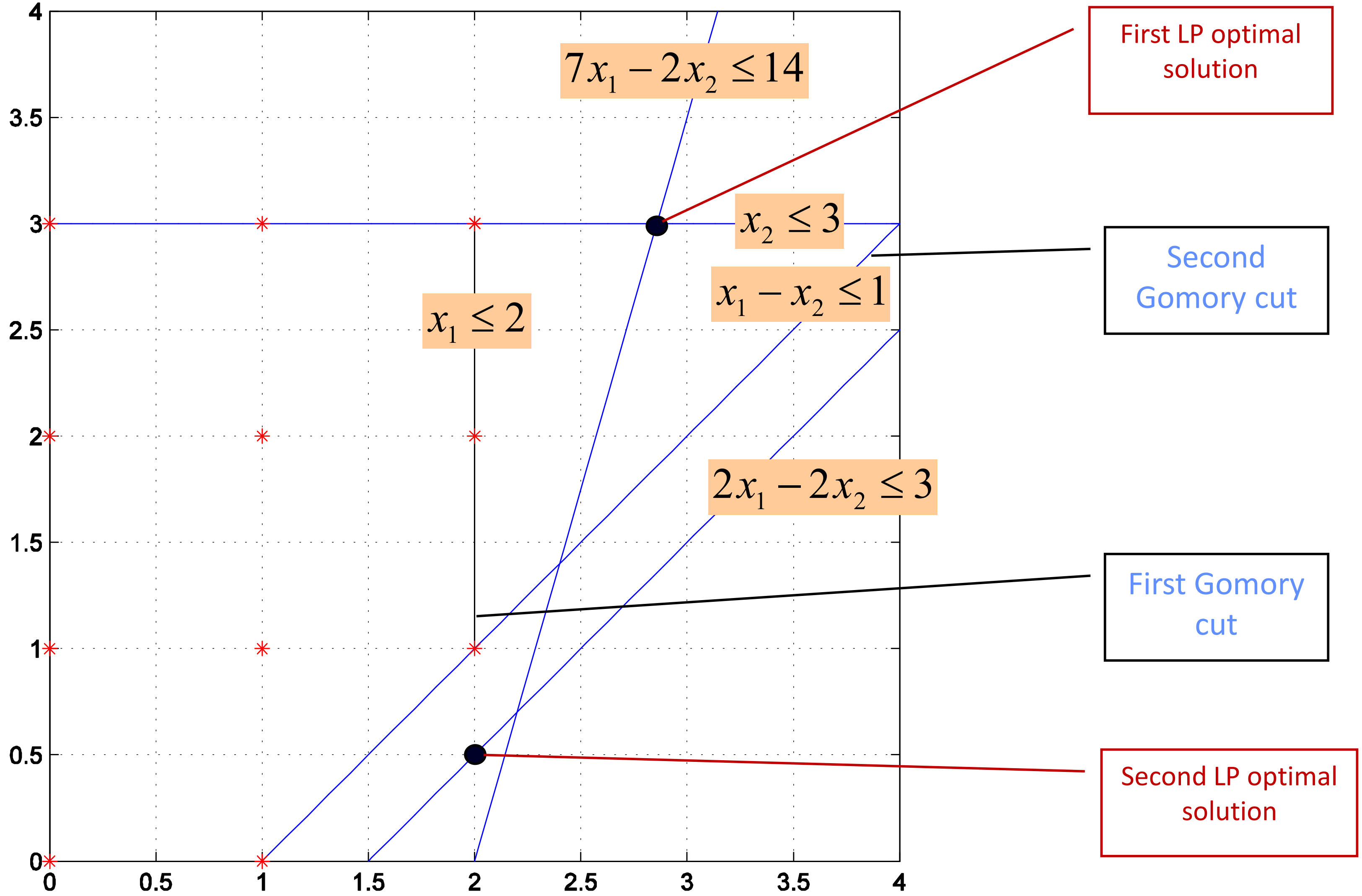
$$x_1 \leq 2$$

$$0.5h_3 \geq 0.5$$

$$3 - 2x_1 + 2x_2 \geq 1$$

$$x_1 - x_2 \leq 1$$

Gomory's cut (v)



Different types of cuts

Different types of cuts and their characteristics, where z is binary unless otherwise noted, and x is continuous.

Cut name	Mathematical description of cut	Structure of original MILP that generates the cut
Clique[b]	$\sum_i z_i \leq 1$	Packing constraints
Cover[b]	$\sum_i z_i \leq b$, b integer	Knapsack constraints
Disjunctive[a]	Constraint derived from an LP solution	$\sum_i a_i' x_i \geq b'$ or $\sum_i a_i'' x_i \geq b''$, x_i continuous or integer
Mixed Integer Rounding[a]	Use of floors and ceilings of coefficients and integrality of original variables	$a_C x_C + a_I x_I = b, x \geq 0$
Generalized Upper Bound[b]	$\sum_i x_i \leq b$, b integer	Knapsack constraints with precedence or packing
Implied Bound[b]	$x_i \leq \frac{b}{a_i}$	$\sum_i a_i x_i \leq bz, x \geq 0$
Gomory[a]	Mixed integer rounding applied to a simplex tableau row $\bar{a}$ associated with optimal node LP basis	$\bar{a}_C x_C + \bar{a}_{I/k} x_{I/k} + x_k = \bar{b}$, x_k integer, $x \geq 0$
Zero-half[a]	$\lambda^T Ax \leq \lfloor \lambda^T b \rfloor$, $\lambda_i \in \{0, 1/2\}$	Constraints containing integer variables and coefficients
Flow Cover[b]	Linear combination of flow and binary variables involving a single node	Fixed charge network
Flow Path[b]	Linear combination of flow and binary variables involving a path of nodes	Fixed charge network
Multicommodity flow[b]	Linear combination of flow and binary variables involving nodes in a network cut	Fixed charge network

[a] Based on general polyhedral theory.
[b] Based on specific, commonly occurring problem structure.

E. Klotz, A.M. Newman *Practical Guidelines for Solving Difficult Mixed Integer Linear Programs* Surveys in Operations Research and Management Science 18 (1-2), 18-32, Oct 2013 10.1016/j.sorms.2012.12.001

Branch and cut (B&C) method

- ❑ Branch and bound + cutting planes in the nodes
- ❑ Decreases solution time
- ❑ Procedure
  - ✓ Select a node to evaluate (initially, the root node is the original relaxed problem) and solve it
  - ✓ Generate or not cutting planes. Add to the problem and solve it.
  - ✓ Prune and branch with the criteria of the branch and bound (B&B) method.

CPLEX Performance Tuning for MIP

(https://www.ibm.com/support/pages/cplex-performance-tuning-mixed-integer-programs)

- Names no
- NodeFileInd 3
- **NodeSel** 0
- **VarSel** 3
- **StartAlg** 4
- MemoryEmphasis 1
- WorkMem 1000
- MIPEmphasis 2
- MIPSearch 2
- SolveFinal 0
- Solution Polishing
- Solution pool
- FlowCovers
- FeasOptMode 2
- FeasOpt 1

- **RINSHeur 100**
- FpHeur 2

Pure branch and bound

- Cuts -1
- HeurFreq -1

Presolve

- PreInd, PrePass

- Solution method of LP problem
  - ✓ First iteration (interior point or simplex method)
  - ✓ Successive iterations (primal or dual simplex)
- Priority for variable selection
  - ✓ Select variables that impact the most in the o.f. (e.g., investment vs. operation variables)
- Initial cutoff or incumbent
  - ✓ Initial valid bound of the o.f. estimated by the user

GUROBI (https://www.gurobi.com/documentation/9.5/refman/mip_models.html)

- Most important parameters
  - ✓ Threads, MIPFocus
- Solution Improvement
  - ✓ ImproveStartTime, ImproveStartGap
- Termination
  - ✓ TimeLimit
  - ✓ MIPGap, MIPGapAbs
  - ✓ NodeLimit, IterationLimit, SolutionLimit
  - ✓ Cutoff
- Speeding Up The Root Relaxation
  - ✓ Method
- Heuristics
  - ✓ Heuristics, SubMIPNodes, MinRelNodes, PumpPasses, ZeroObjNodes
  - ✓ **RINS 100**
- Cutting Planes
  - ✓ Cuts, FlowCoverCuts, MIRCuts
- **Presolve**
  - ✓ **Presolve, PrePasses, AggFill, PreSparsify**

Andrés Ramos

https://www.iit.comillas.edu/aramos/

Andres.Ramos@comillas.edu

Pedro Sánchez

Pedro.Sanchez@comillas.edu

Sonja Wogrin

https://www.iit.comillas.edu/swogrin/

Sonja.Wogrin@comillas.edu