Escuela Técnica Superior de Ingeniería
Departamento de Organización Industrial

Optimización lineal entera mixta

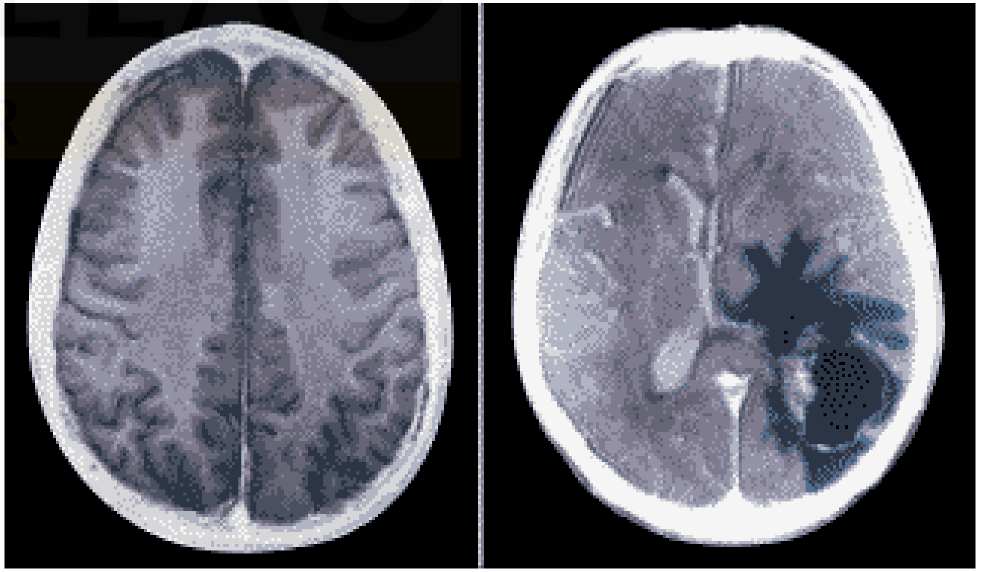
Andrés Ramos
Universidad Pontificia Comillas
http://www.iit.comillas.edu/aramos/
Andres.Ramos@comillas.edu

CONTENIDO

- ➢ INTRODUCCIÓN
- ❑ MÉTODOS DE SOLUCIÓN
- ❑ MÉTODO DE RAMIFICACIÓN Y ACOTAMIENTO
- ❑ DUALITY (master)
- ❑ PREPROCESSING (master)
- ❑ BRANCH AND CUT METHOD (master)

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?



Introducción

- Un problema de programación lineal entero mixto (MIP) es un problema lineal (LP) con algunas variables enteras
  - ✓ Programación lineal entera mixta (MILP)
    - $x \in R^+$, $y \in Z^+$
  - ✓ Programación entera pura (PIP)
    - $x \in Z^+$
  - ✓ Programación binaria (0-1 MIP, 0-1 IP, BIP)
    - $x \in \{0,1\}$: variables de asignación, lógicas
- Son más difíciles de resolver que los problemas LP
- Primer algoritmo de resolución se formuló por Ralph Gomory en 1958

CONTENIDO

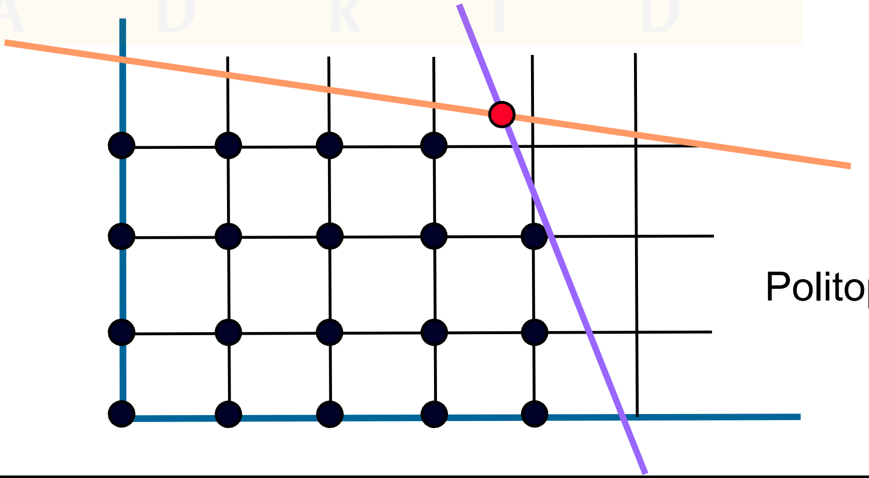
- ❑ INTRODUCCIÓN
- ➢ MÉTODOS DE SOLUCIÓN
- ❑ MÉTODO DE RAMIFICACIÓN Y ACOTAMIENTO
- ❑ DUALITY (master)
- ❑ PREPROCESSING (master)
- ❑ BRANCH AND CUT METHOD (master)

Métodos de solución

- Relajación lineal y discretización
- Enumeración exhaustiva
- Ramificación y acotamiento (*branch and bound*)
- Método de los planos de corte
- Ramificación y corte (*branch and cut*)

Relajación lineal y discretización (i)

- Problema relajado: aquél donde a las variables enteras se les permite tomar valores reales
- Si la solución cumple las condiciones de integralidad entonces es el óptimo del problema entero
  - ✓ Politopo entero: todos los puntos extremos son enteros.
  - ✓ Coincide con la envoltura convexa de las soluciones.
  - ✓ Es entero si la matriz A es totalmente unimodular (toda submatriz cuadrada tiene determinante 1, 0 ó -1).
  - ✓ Problema de transporte, asignación, flujo de coste mínimo.



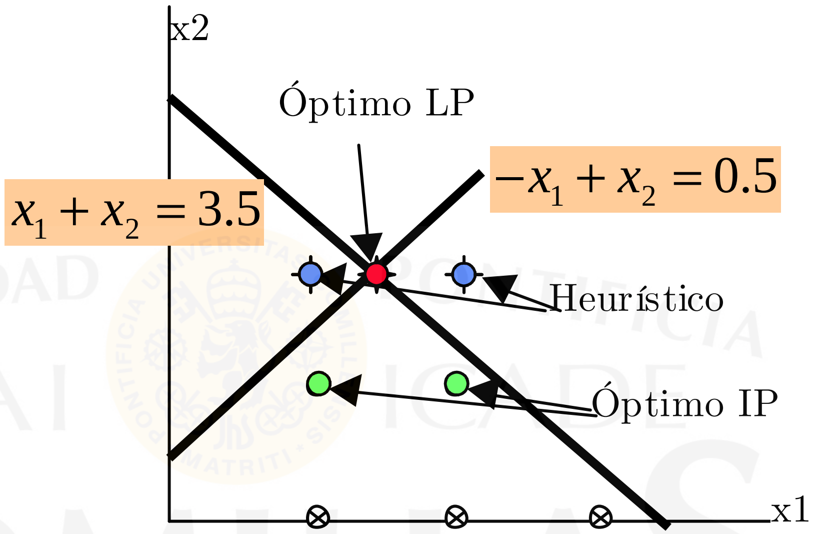
Politopo NO entero

Relajación lineal y discretización (ii)

- La solución de un problema entero NO es necesariamente la solución del problema relajado discretizada heurísticamente (redondeada a los valores enteros más próximos).
  - ✓ Solución aproximada si las variables enteras toman valores elevados
  - ✓ Posible pérdida de optimalidad
  - ✓ Posible pérdida de factibilidad
- Los métodos metaheurísticos son una alternativa a los de programación matemática (algoritmos genéticos, búsqueda heurística, etc.)

Discretización: pérdida de factibilidad

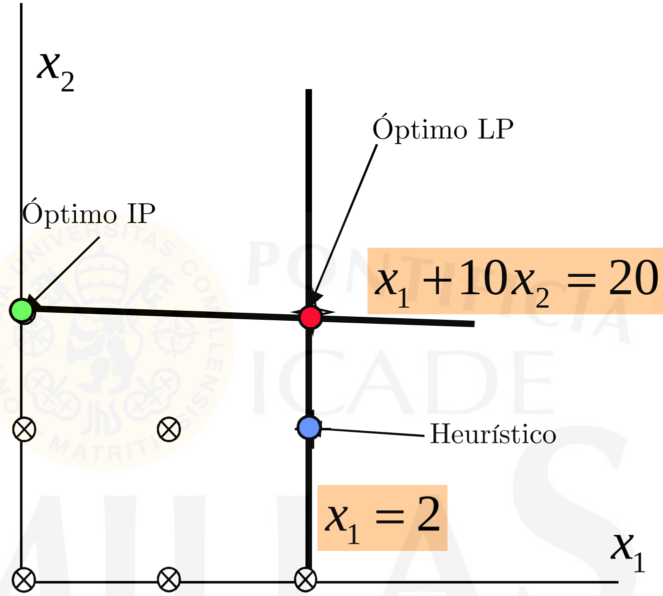
$$\max \quad x_2$$
$$-x_1 + x_2 \leq 0.5$$
$$x_1 + x_2 \leq 3.5$$
$$x_1, x_2 \geq 0 \quad x_i \in \mathbb{Z}$$



- Solución LP: (1.5,2)
- Soluciones discretizadas: (1,2) o bien (2,2), resultan infactibles
- Soluciones enteras óptimas: (1,1) o bien (2,1)

Discretización: pérdida de optimalidad

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



- Solución LP: (2,9/5)
- Solución discretizada: (2,1)
- Solución entera óptima: (0,2)

Enumeración exhaustiva

- No es viable, debido a que el número de soluciones crece exponencialmente

- En un problema BIP de *n* variables hay 2^n posibles soluciones

Métodos de solución

- Pérdida de convexidad de la región factible. Los puntos de su interior no se pueden poner como combinación lineal convexa de sus puntos extremos.
- Pérdida de la potencia matemática asociada a variables continuas (derivadas, condiciones de optimalidad, sensibilidades, etc.).
- La solución de un problema MIP es más difícil que la de un problema LP. Requiere más tiempo de cálculo y más requisitos de memoria.

CONTENIDO

- INTRODUCCIÓN
- MÉTODOS DE SOLUCIÓN
- MÉTODO DE RAMIFICACIÓN Y ACOTAMIENTO
- DUALITY (master)
- PREPROCESSING (master)
- BRANCH AND CUT METHOD (master)

Método de ramificación y acotamiento (*branch and bound*)

- ❑ Enumeración implícita de las soluciones enteras factibles.
- ❑ Utiliza el principio de divide y vencerás.
  - ✓ Divide (ramifica) el conjunto de soluciones enteras en subconjuntos disjuntos cada vez menores.
  - ✓ Determina (acota) el valor de la mejor solución del subconjunto.
    - En problema de maximización una cota inferior de la solución óptima de un problema MIP es la mayor solución entera factible encontrada hasta el momento.
    - En problema de maximización una cota superior de la solución óptima de un problema MIP es la solución óptima del problema lineal relajado RMIP o LP.
  - ✓ Poda (elimina) la rama del árbol si la cota indica que no puede contener la solución óptima.

Procedimiento (i)

1. Inicialización

- ✓ Inicializa la cota superior de la f.o. $z^* = -\infty$, en problemas de maximización
- ✓ Resolver una relajación del problema (habitualmente la lineal, aunque pueden usarse otras). Éste es el nodo raíz.
- ✓ Aplicar acotamiento o poda y criterio de optimalidad al problema completo.
- ✓ Si no se puede etiquetar el problema como no podado, comienza una iteración completa.

2. Iteración

- ✓ Ramificación
  - ❑ Seleccionar uno de los nodos entre los no explorados (nodos restantes). Ver criterios de selección
  - ❑ Seleccionar una variable entera que tenga valor continuo en la solución óptima del nodo relajado. Ver criterios de selección

Procedimiento (ii)

- Sea x_j^* el valor óptimo en el problema relajado
- Se ramifica en dos ramas incorporando las restricciones $\begin{cases} x_j \le \lfloor x_j^* \rfloor \\ x_j \ge \lfloor x_j^* \rfloor + 1 \end{cases}$

  siendo $\lfloor x_j^* \rfloor$ la parte entera de x_j^*
- Para variables binarias es fijar el valor de la variable a 0 ó a 1. No se puede repetir la ramificación.
- Cada vez que se ramifica se añade una restricción (análisis de sensibilidad mediante el método simplex dual). El problema primal resulta infactible. El problema dual resulta factible pero no óptimo.
- Cada rama elimina la solución óptima del problema anterior.

✓ Acotamiento

- Para cada nodo se obtiene su f.o. z.

Procedimiento (ii)

- ✓ Poda
  - Se intentan eliminar nodos (ramas) del árbol. Aplicar los siguientes criterios de poda para un problema de maximización:
    1. Solución (entera o no) peor que la solución entera actual $z \leq z^*$, siendo z^* el valor de la función objetivo para la solución entera actual. Se poda la rama.
    2. Solución entera mejor que la actual $z > z^*$. $z^* = z$ nueva solución entera actual

       Se aplica el criterio 1 a todos los nodos no podados con la nueva solución entera actual.
    3. Infactible. Se poda la rama.

3. Criterio de optimalidad

- ✓ Parar cuando no existan nodos sin analizar. La solución entera actual es la óptima.
- ✓ Si no, realizar otra iteración.

Ejemplo

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

Resolución (i)

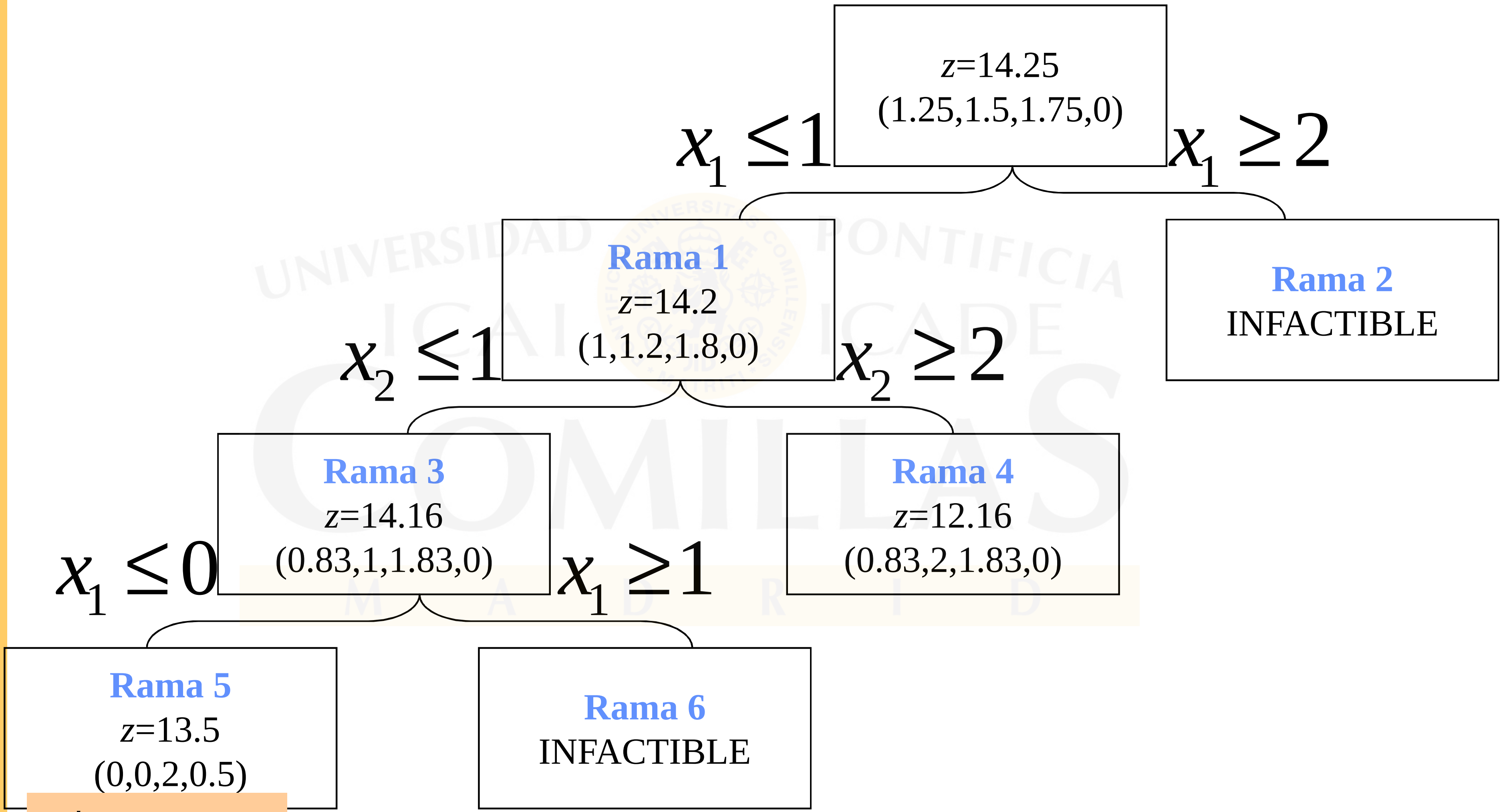
1. Inicialización
   - ✓ Se resuelve el problema LP relajado $z = 14.25$ $(x_1, x_2, x_3, x_4) = (1.25, 1.5, 1.75, 0)$
2. Iteración 1
   - ✓ Se ramifica con la primera variable que debiera ser entera y no lo es, x_1.
   - ✓ Rama 1: $x_1 \leq 1$ $z = 14.2$ $(x_1, x_2, x_3, x_4) = (1, 1.2, 1.8, 0)$
     Cualquier solución descendiente tendrá $z \leq 14.2$
   - ✓ Rama 2: $x_1 \geq 2$ infactible. Se poda la rama
3. Iteración 2
   - ✓ Se ramifica con la primera variable que debiera ser entera y no lo es, x_2.
   - ✓ Rama 3: $x_1 \leq 1$, $x_2 \leq 1$ $z = 14.1\widehat{6}$ $(x_1, x_2, x_3, x_4) = (0.8\widehat{3}, 1, 1.8\widehat{3}, 0)$
     Cualquier solución descendiente tendrá $z \leq 14.1\widehat{6}$
   - ✓ Rama 4: $x_1 \leq 1$, $x_2 \geq 2$ $z = 12.1\widehat{6}$ $(x_1, x_2, x_3, x_4) = (0.8\widehat{3}, 2, 1.8\widehat{3}, 0)$
     Cualquier solución descendiente tendrá $z \leq 12.1\widehat{6}$

Resolución (ii)

4. Iteración 3
   - ✓ Se selecciona la rama 3 por tener la mayor función objetivo.
   - ✓ Se ramifica con la variable x_1 .
   - ✓ Rama 5: $\begin{array}{l} x_1 \le 1 \\ x_2 \le 1 \\ x_1 \le 0 \end{array}$ $z = 13.5$ $(x_1, x_2, x_3, x_4) = (0, 0, 2, 0.5)$

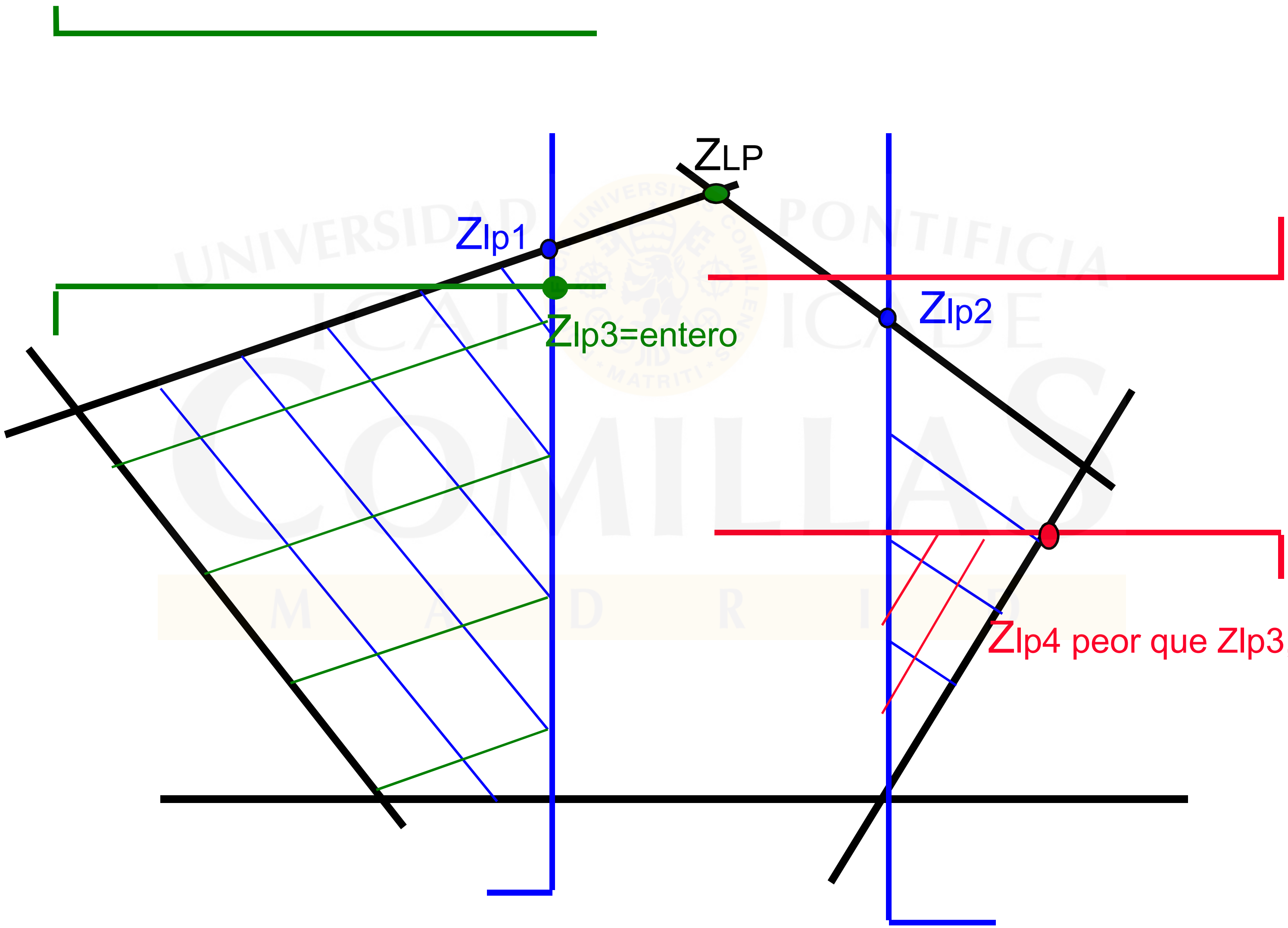
     Primera solución entera del problema MIP $z^* = 13.5$
   - ✓ Rama 6: $\begin{array}{l} x_1 \le 1 \\ x_2 \le 1 \\ x_1 \ge 1 \end{array}$ infactible
   - ✓ La rama 4 se puede podar porque su función objetivo es menor (en maximización) que la solución entera actual.
5. Criterio de optimalidad
   - ✓ Solución óptima alcanzada por no existir ramas sin explorar.

Árbol



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

Interpretación geométrica del método de ramificación y acotamiento



Linear problem (LP). Example 1

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

$$x + y \leq 11$$

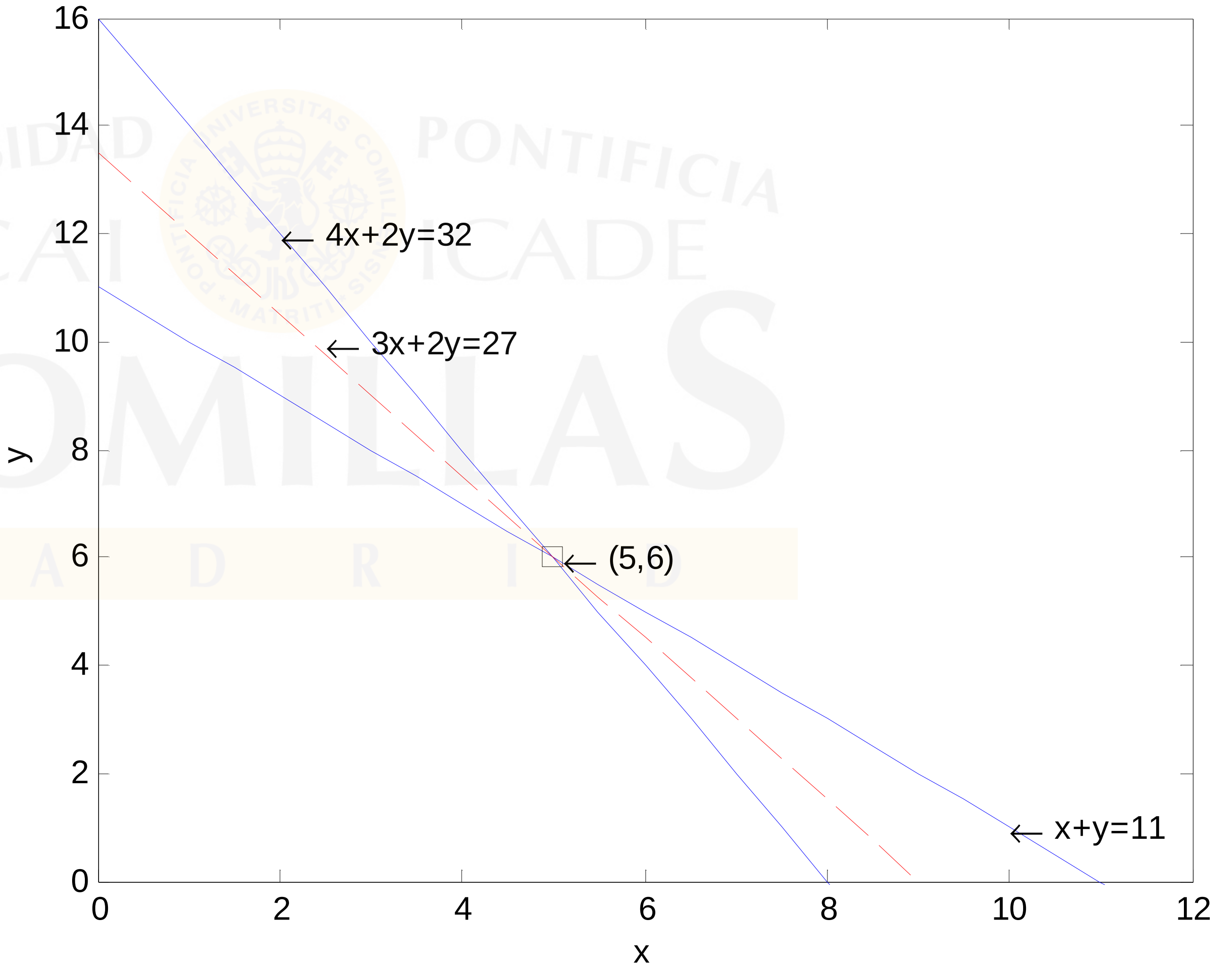
$$4x + 2y \leq 32$$

$$x, y \geq 0$$

LP solution

$$z = 27$$

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



Pure integer problem (PIP). Example 1

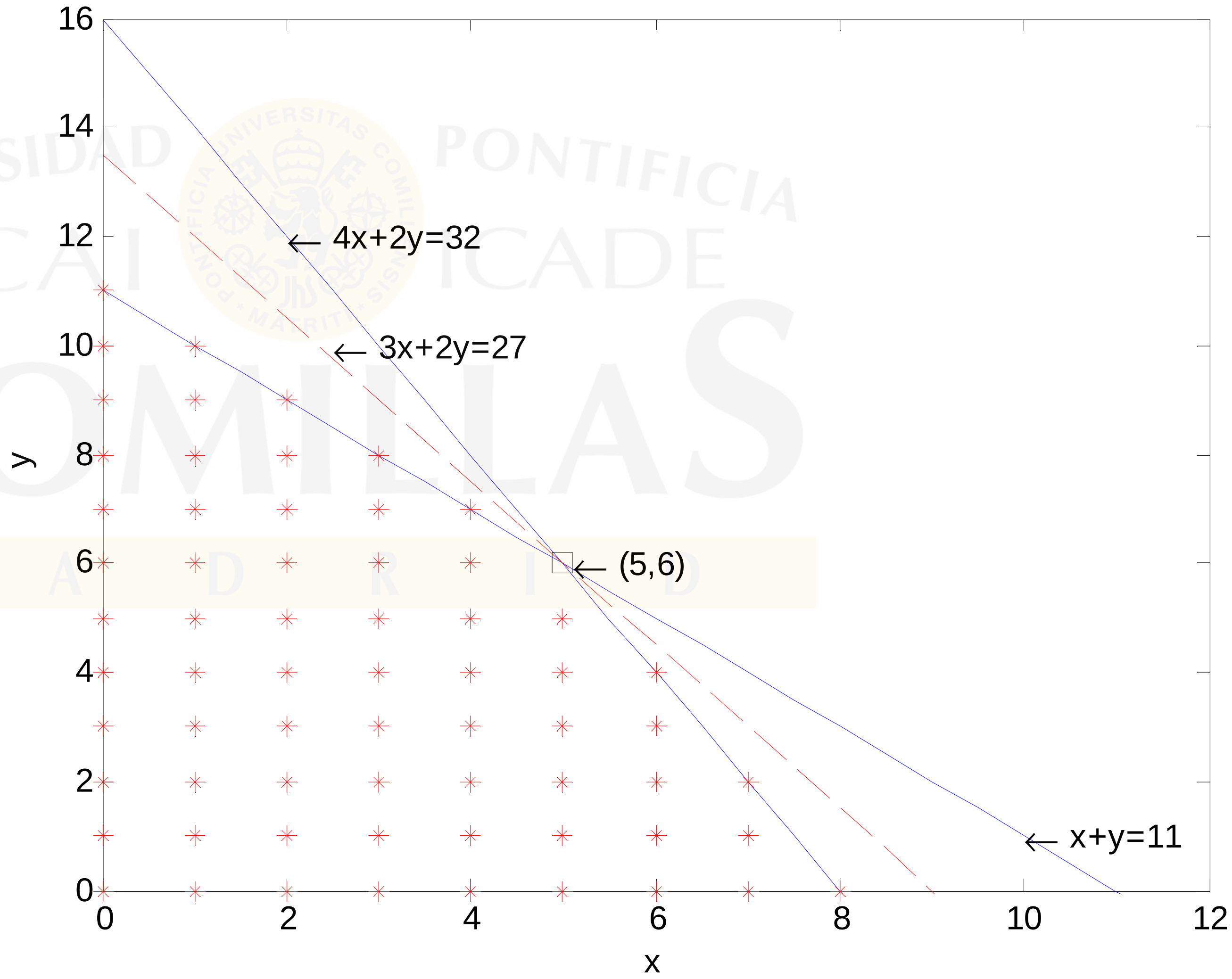
$$\max_{x,y} 3x + 2y$$

$$x + y \leq 11$$

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

$$x, y \geq 0$$

$$x, y \in \mathbb{Z}^+$$



PIP solution

$$z = 27$$

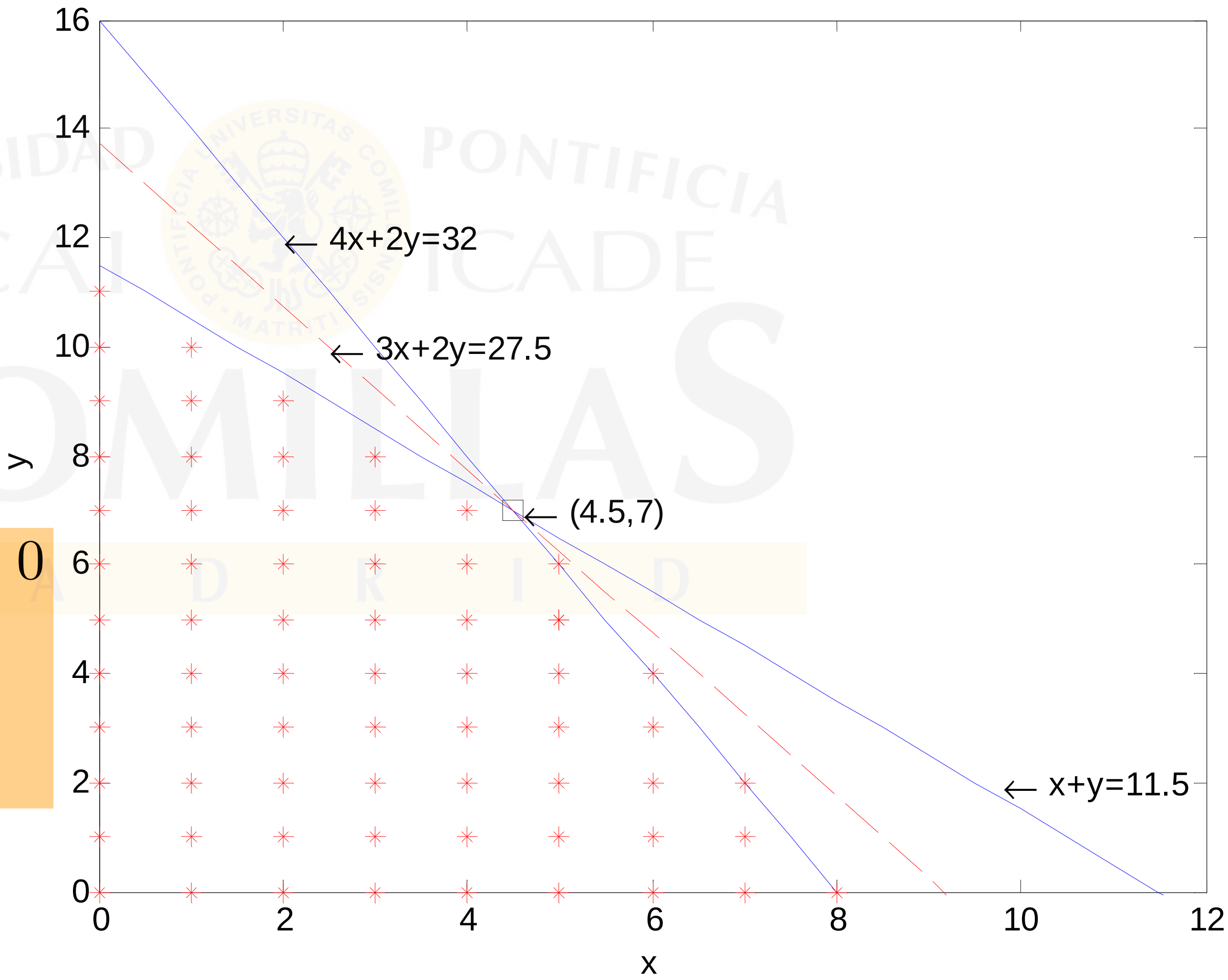
$$(x^*, y^*) = (5,6)$$

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

$$\max_{x,y} 3x+2y$$
$$x+y\le 11.5$$
$$4x+2y\le 32$$
$$x\le 4$$
$$x,y\ge 0$$

LP Problem 1

$z=27$

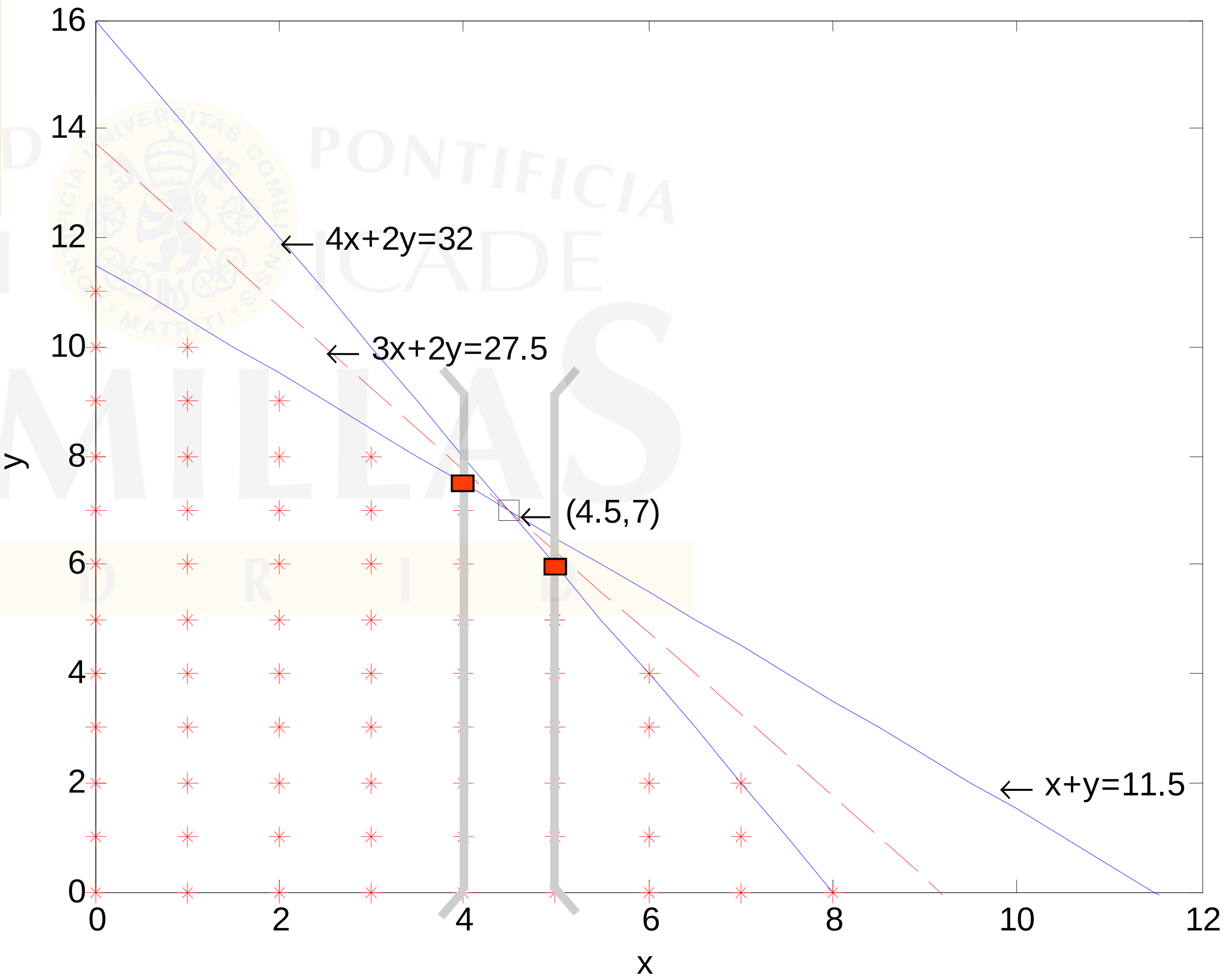
$(x^*,y^*)=(4,7.5)$

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

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

$z = 26.5$

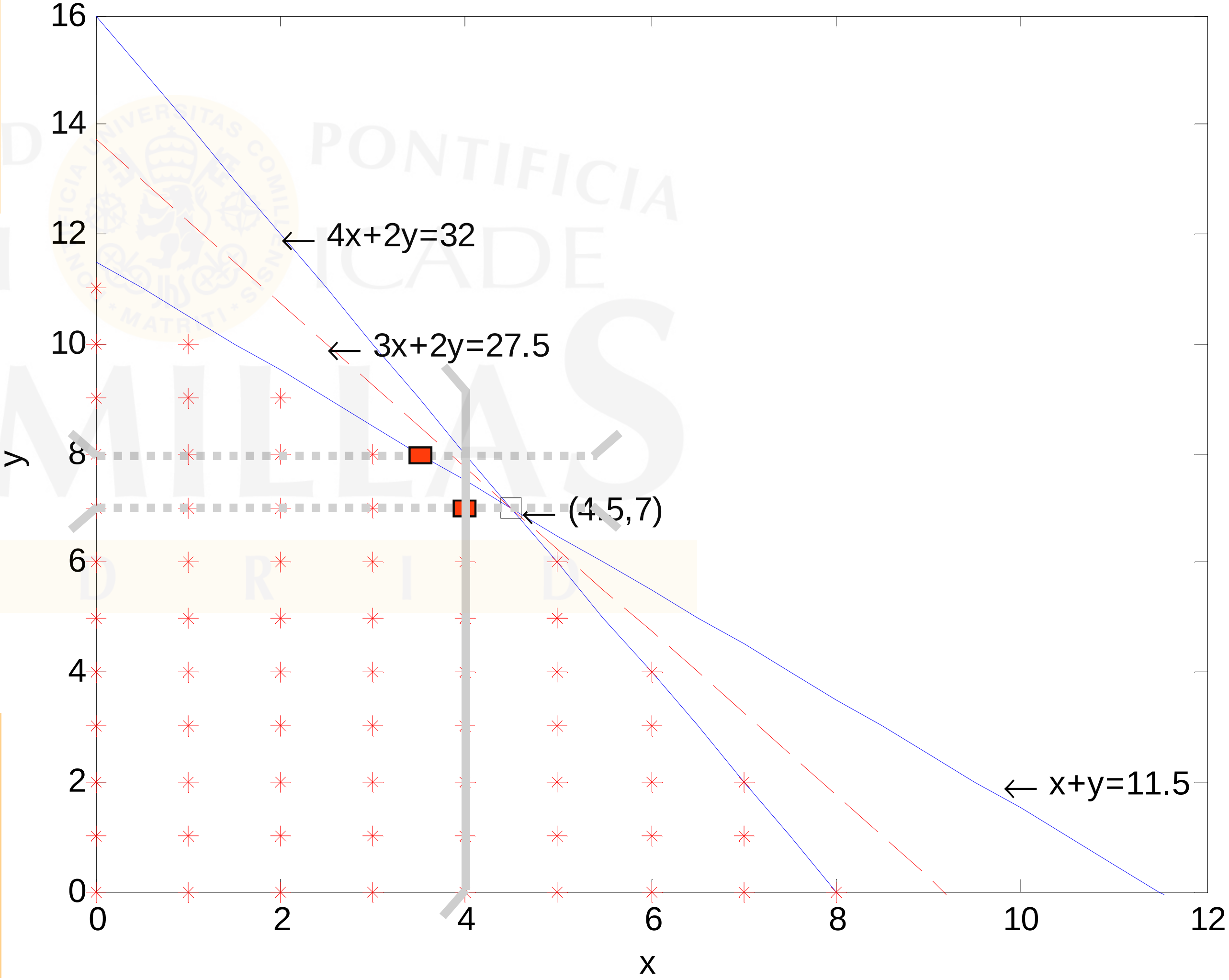
$(x^*, y^*) = (3.5, 8)$

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

LP Problem 4

$z = 26$

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



Pure integer problem (PIP). Example 2

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

$$x + y \leq 11.5$$

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

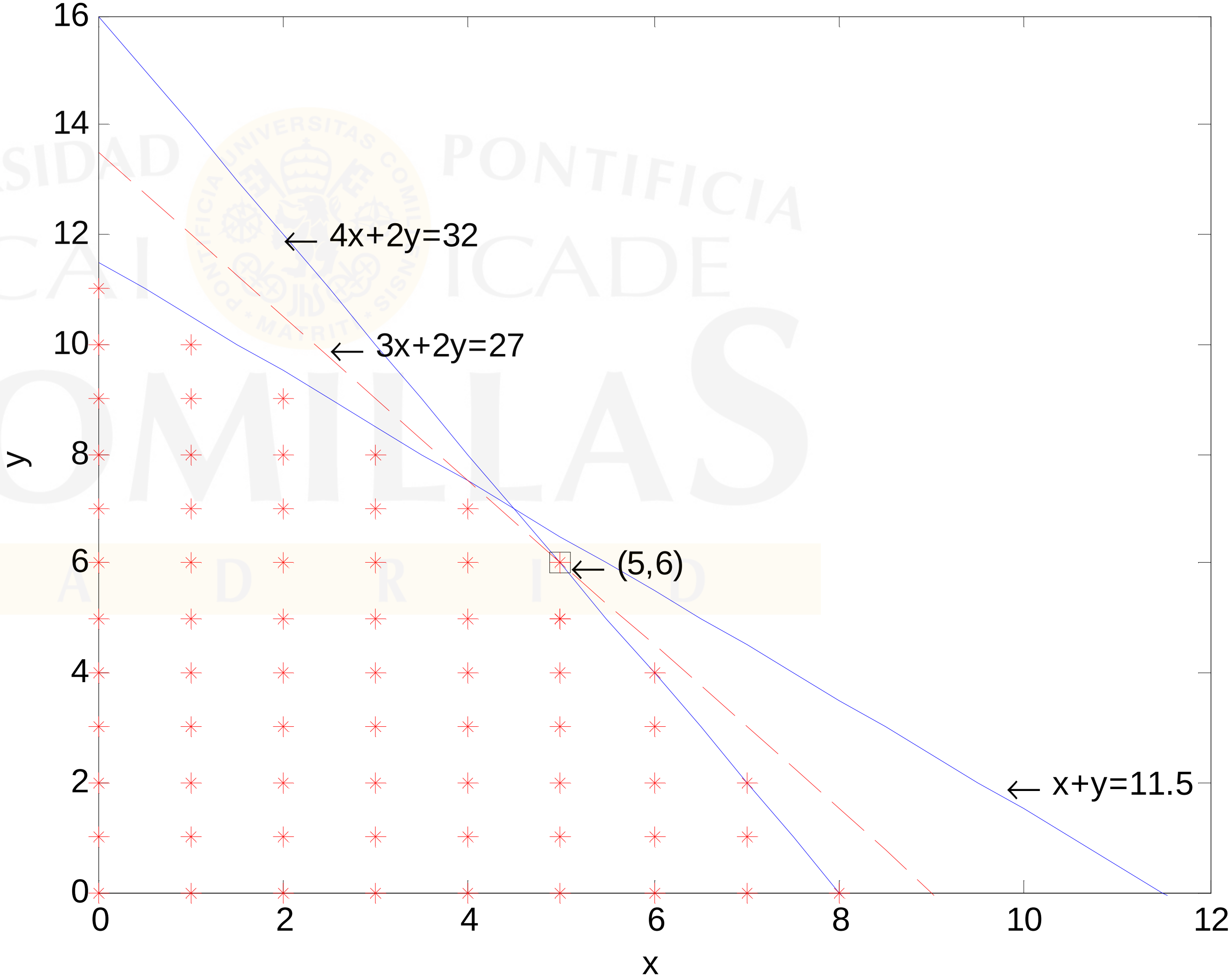
$$x, y \geq 0$$

$$x, y \in \mathbb{Z}^+$$

PIP solution

$$z = 27$$

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



Estrategias básicas

1. Buscar *factibilidad*
   - ✓ Ramificación y exploración en profundidad (avariciosa *greedy*) en el árbol fijando recursivamente las variables fraccionarias más próximas a su valor entero en el nodo seleccionado.

2. Demostrar *optimalidad*
   - ✓ Supuesto que se disponga de una solución entera se desea probar que ésta es óptima. Se seleccionan las variables que tienen un gran impacto en la función objetivo para descartar ramas del árbol lo antes posible.

Implantación de estrategias

1. Selección de la variable entera a ramificar
   - ✓ La encontrada en primer lugar
   - ✓ La de mayor o menor infactibilidad entera

2. Selección de la rama a resolver
   - ✓ La más reciente. Bueno para la reoptimización por el método simplex.
   - ✓ Aquélla con valor de la función objetivo más cercano o alejado al óptimo (mejor o peor cota)

Relajación del criterio de poda

- Marca la diferencia entre acabar un problema con una solución cuasióptima dentro de una cierta tolerancia conocida y NO solucionar el problema. Fundamental en la solución de problemas reales.
- Criterio de parada sin explorar exhaustivamente el árbol.

  Se añade un criterio de poda (para maximización) con una cierta tolerancia para una solución no entera mejor (pero no significativamente) que la solución entera actual

  - ✓ Relativa $z^* \le z \le z^*(1+\alpha)$
  - ✓ Absoluta $z^* \le z \le z^* + \beta$

  α (error de tolerancia relativo, por ejemplo 10^{-3}) **OPTCR**

  β (error de tolerancia absoluto), ambas constantes conocidas **OPTCA**

Parámetros de control del B&B (i)

- Prioridad en la selección de variables
  - ✓ Habitualmente se debe ramificar antes en las que más impacto tienen en la f.o. (por ejemplo, variables de inversión frente a las de operación)
- GUB (*generalized upper bound*) o SOS (*special ordered sets*)

$$\sum_{j=1}^{k} x_j = 1$$

  - ✓ En la ramificación normal cuando una $x_j = 1$ el resto quedan fijadas a 0. En la rama con $x_j = 0$ hay a su vez $k-1$ posibilidades
  - ✓ La ramificación GUB se hace ordenando las variables pertenecientes al conjunto GUB en dos subconjuntos más equilibrados, hasta que la suma de las variables de un conjunto sobrepasa 0.5

$$\begin{aligned} x_{j_i} &= 0 \quad i = 1,\ldots,r \\ x_{j_i} &= 0 \quad i = r+1,\ldots,k \end{aligned}$$

$$r = \min\left\{t : \sum_{i=1}^{t} x_{j_i}^{*} \geq 0.5\right\}$$

Parámetros de control del B&B (ii)

- Cota inicial (*cutoff, incumbent*)
  - ✓ Se trata de una cota inicial válida de la f.o. estimada por el usuario
- Método de solución de los problemas LP
  - ✓ Primera iteración (punto interior o simplex)
  - ✓ Iteraciones sucesivas (simplex primal o dual con diferentes estrategias de selección de VBE)

CONTENIDO

- INTRODUCCIÓN
- MÉTODOS DE SOLUCIÓN
- MÉTODO DE RAMIFICACIÓN Y ACOTAMIENTO
- DUALITY (master)
- PREPROCESSING (master)
- BRANCH AND CUT METHOD (master)

Pure integer problem (PIP). Dual variables

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

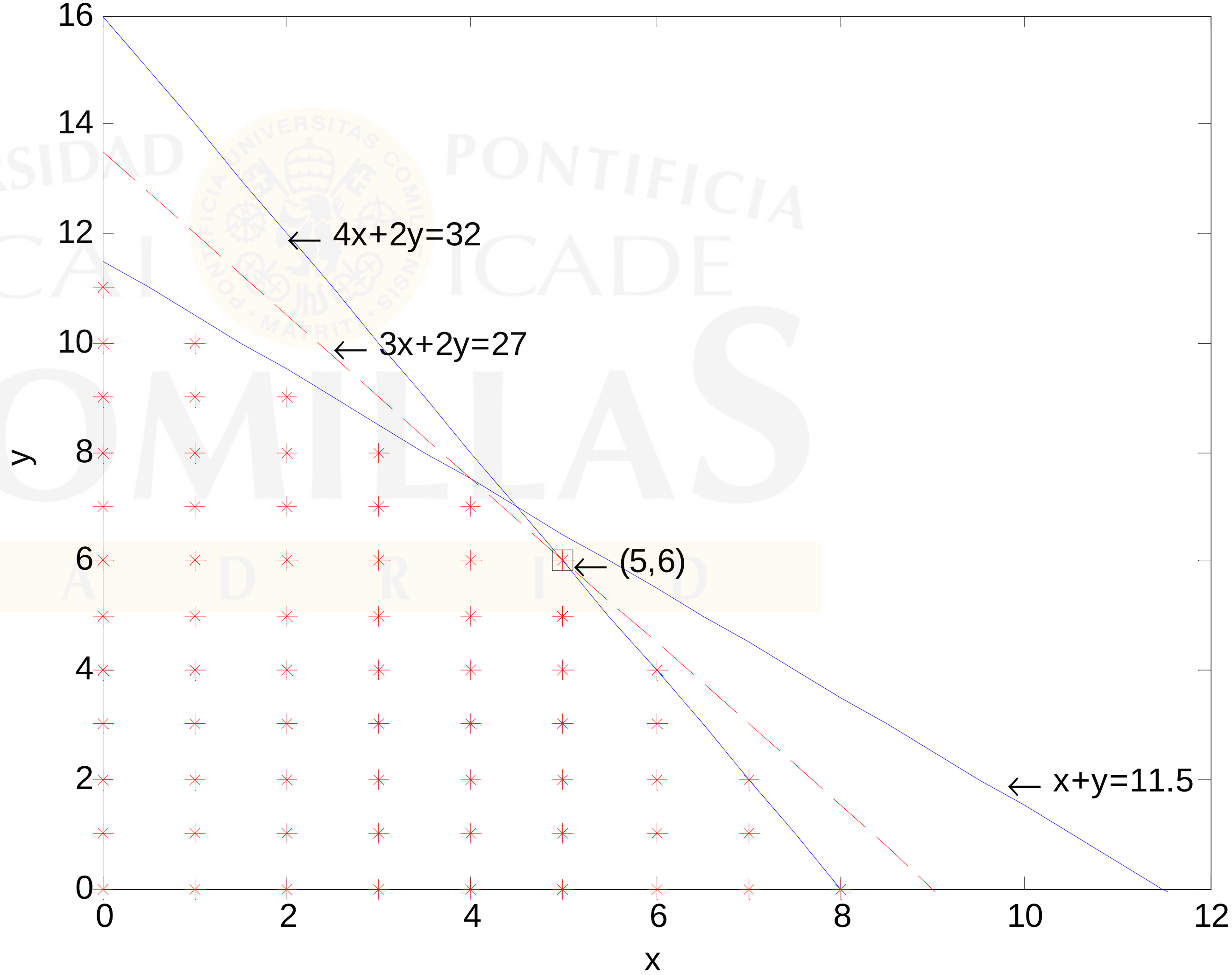
Pure integer problem (PIP). Dual variables. Example 2

$$\max_{x,y} 3x + 2y$$
$$x + y \le 11.5$$
$$4x + 2y \le 32$$
$$x, y \ge 0$$
$$x, y \in \mathbb{Z}^+$$

PIP solution

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

Obtain dual variables manually (resolving the model)



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

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

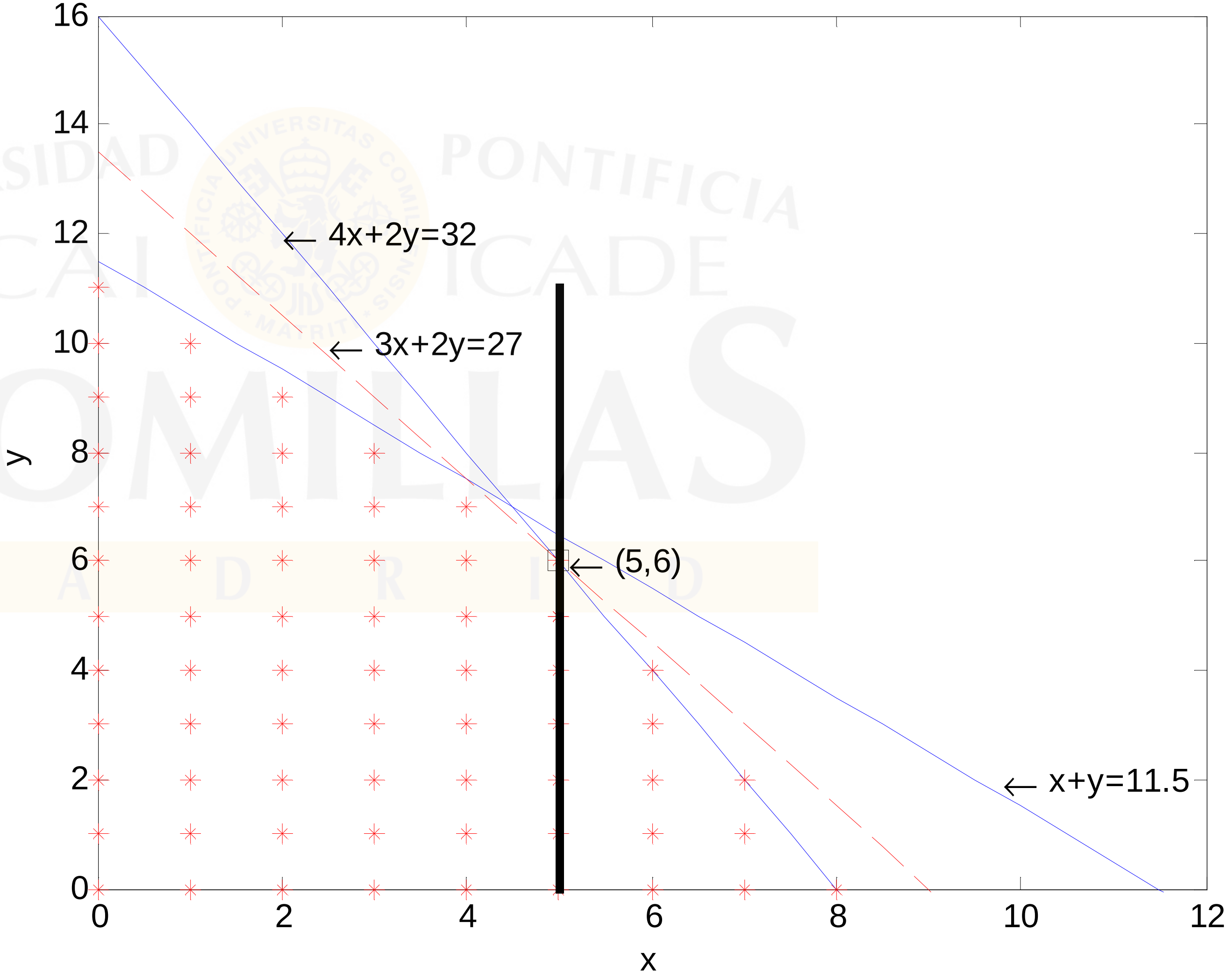
$$x + y \leq 11.5$$

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

$$\boxed{x = 5}$$

$$x, y \geq 0$$

Let's try fixing one variable



LP solution

$$z = 27$$

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

$$\boxed{(\pi_1^*, \pi_2^*) = (0,1)}$$

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

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

$$x + y \leq 11.5$$

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

$$\boxed{y = 6}$$

$$x, y \geq 0$$

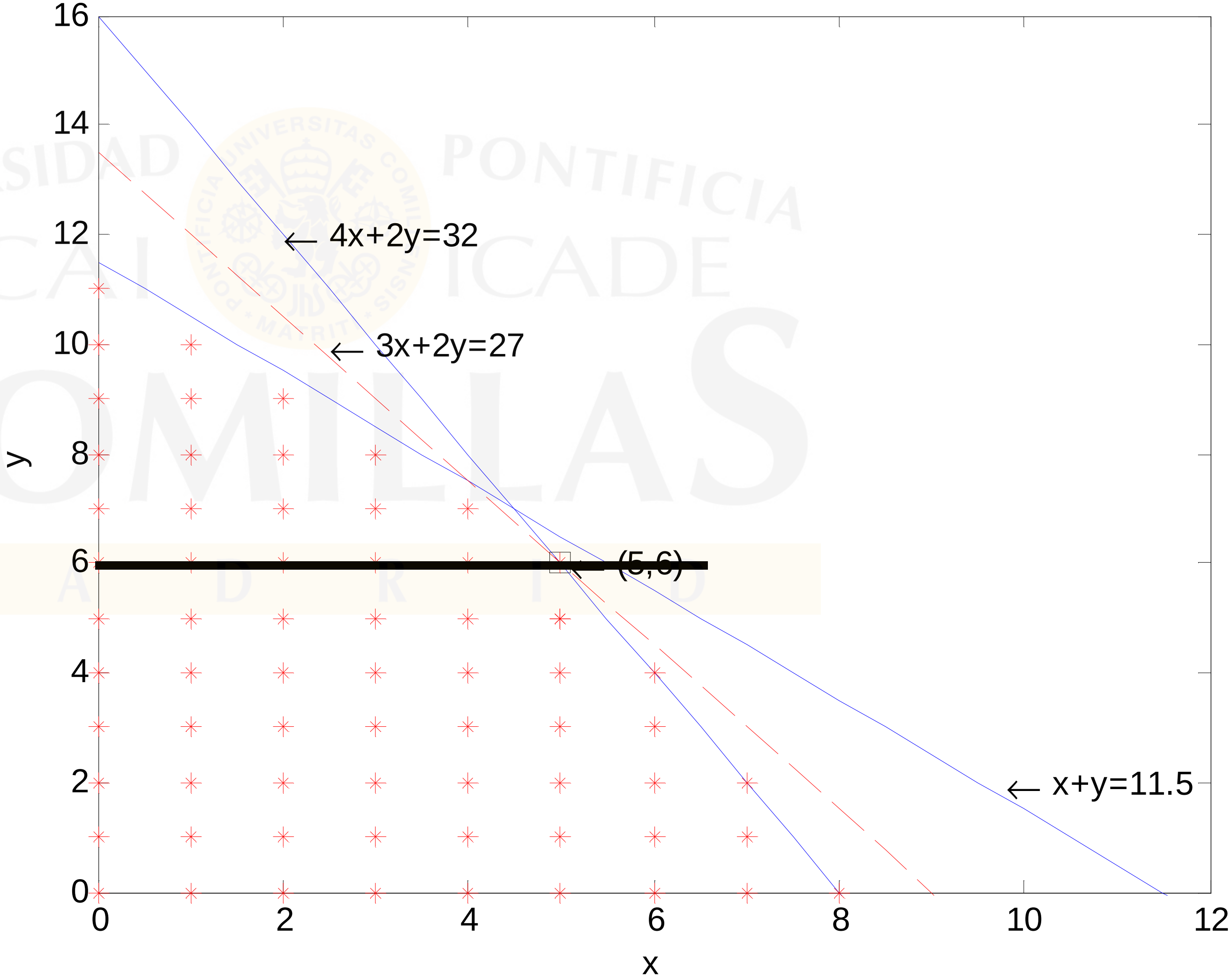
LP solution

$$z = 27$$

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

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

Let's try fixing the other variable



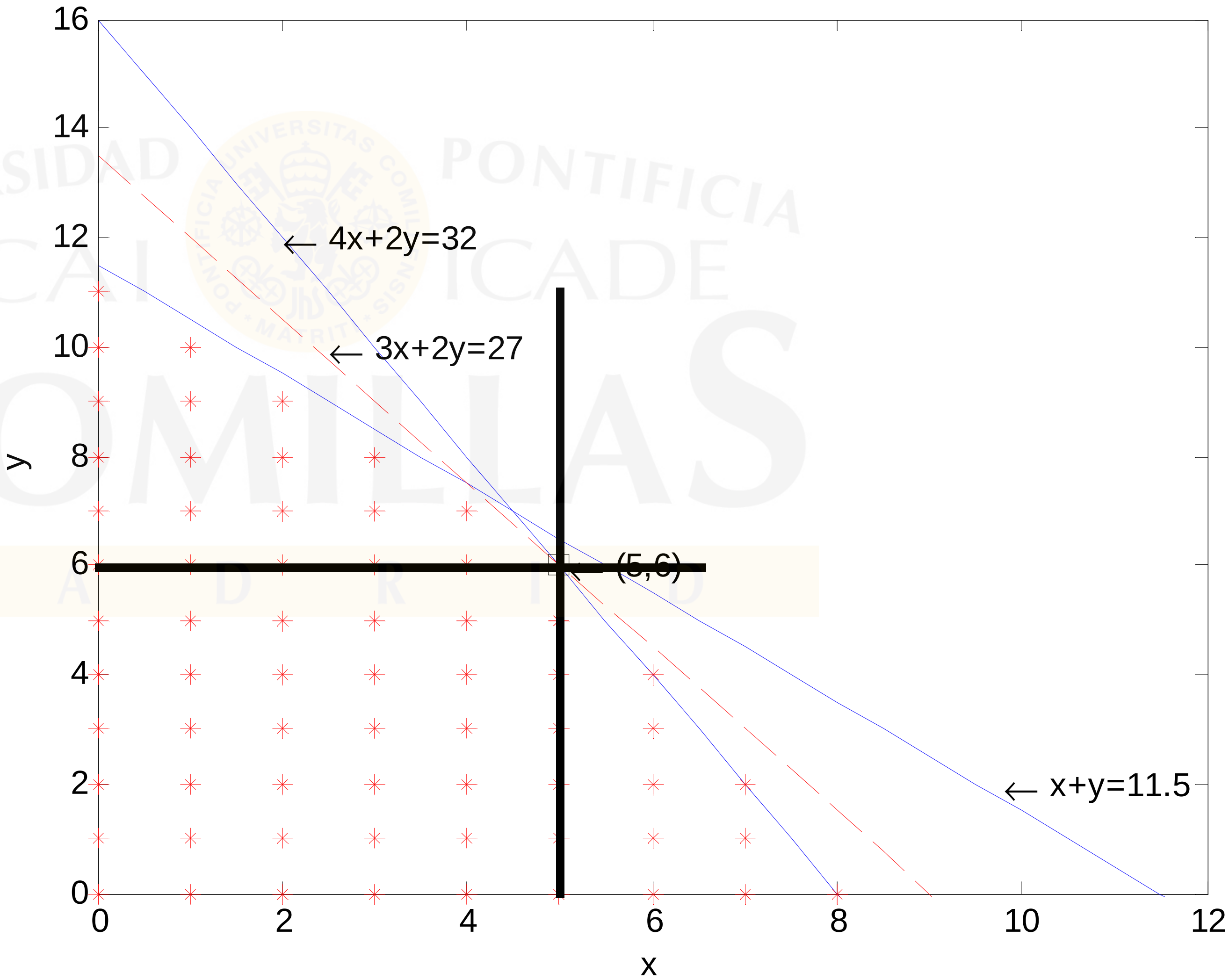
Pure integer problem (PIP). Dual variables. Example 2

$$\max_{x,y} 3x + 2y$$
$$x + y \le 11.5$$
$$4x + 2y \le 32$$
$$x = 5$$
$$y = 6$$
$$x, y \ge 0$$

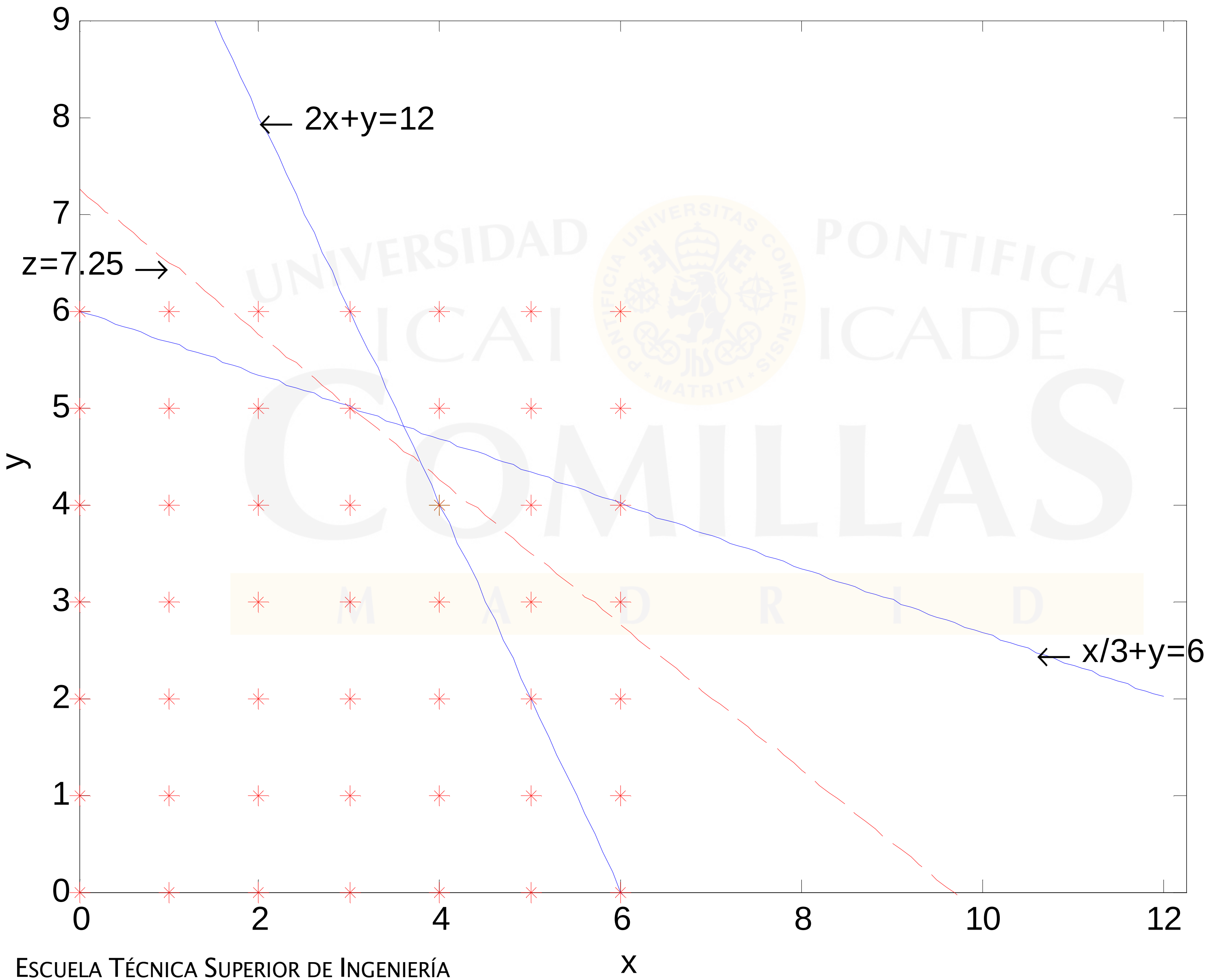
LP solution

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

Let's try fixing both variables



Pure integer problem (PIP). Example 3



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

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

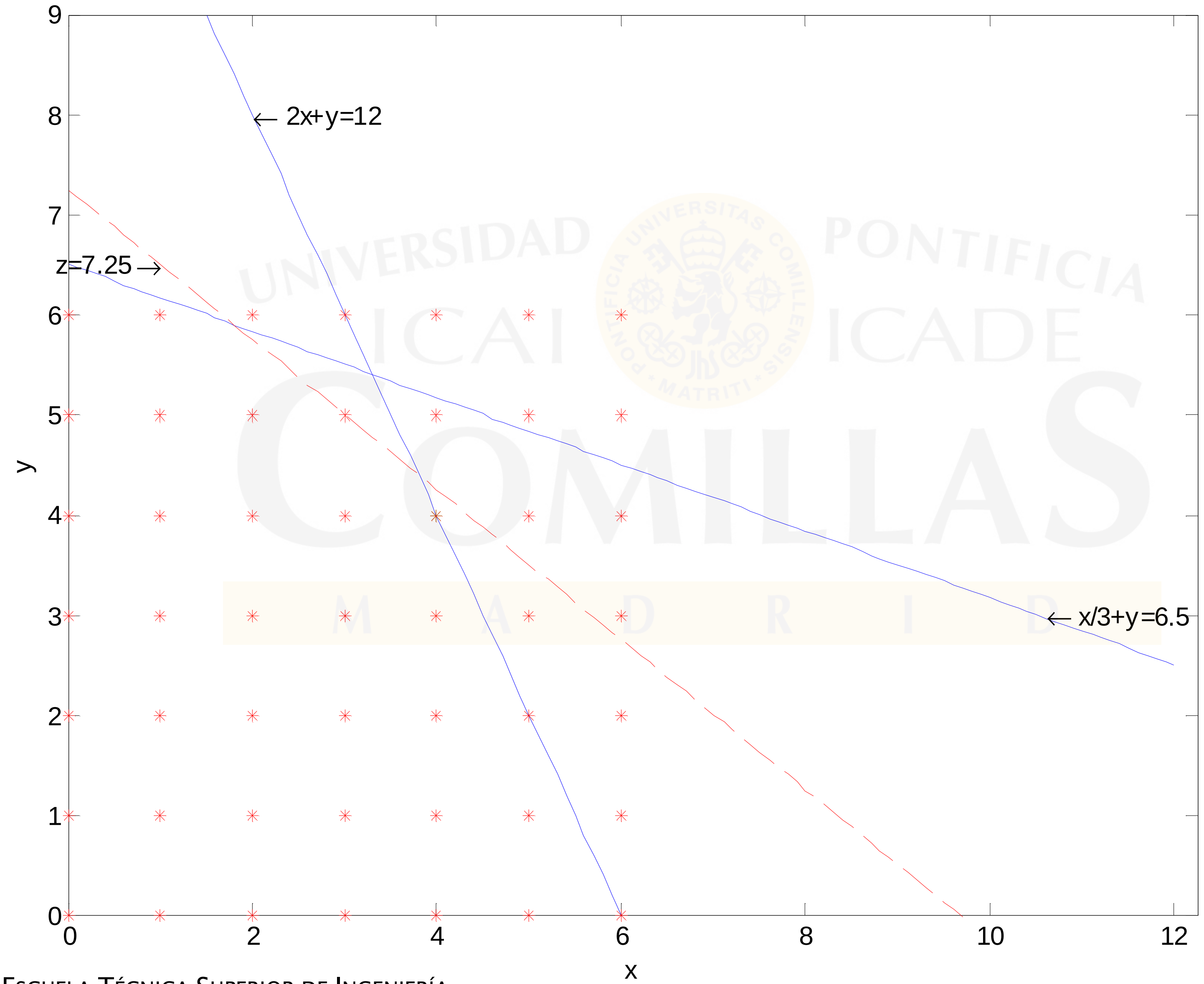
$$2x + y \le 12 \qquad : \pi_2$$

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

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

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

Pure integer problem (PIP). Example 4



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

$$\frac{x}{3} + y \leq 6.5 \qquad : \pi_1$$

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

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

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

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

Dual variables in a MIP problem

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

CONTENIDO

- ❑ INTRODUCCIÓN
- ❑ MÉTODOS DE SOLUCIÓN
- ❑ MÉTODO DE RAMIFICACIÓN Y ACOTAMIENTO
- ❑ DUALITY (master)
- ➢ PREPROCESSING (master)
- ❑ BRANCH AND CUT METHOD (master)

- 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, J.M. Latorre, and A. Ramos *Tight and Compact MILP Formulation for the Thermal Unit Commitment Problem* IEEE Transactions on Power Systems (accepted)
- 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 (accepted) 10.1109/TPWRS.2012.2222938

Preproceso

- Técnicas enfocadas a reducir sustancialmente las dimensiones o fortalecer la formulación del problema. Especialmente relevante para resolver problemas MIP
  - ✓ Dos formulaciones de un problema entero se dicen 0-1 *equivalentes* si tienen las mismas soluciones enteras.
  - ✓ Dadas dos formulaciones equivalentes de un problema entero, se dice que una es *más fuerte* que la otra, si la región factible de su relajación lineal está estrictamente contenida en la región factible de la otra. Se encuentran antes soluciones enteras factibles y se pueden podar más ramas del árbol. Las formulaciones más fuertes se realizan, a veces, a costa de incrementar el tamaño del problema (*menos compacta*).
  - ✓ Medida: intervalo de integralidad (*integrality gap*) diferencia entre f.o. de problema MIP y LP.

- 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 (accepted)

Técnicas de preproceso

- Preproceso general
  - ✓ Reforzamiento de cotas
  - ✓ Eliminación de restricciones redundantes
  - ✓ Asignación de variables
- Preproceso mixto 0-1
  - ✓ Reducción de coeficientes

Reforzamiento de cotas

- Aumentar cotas inferiores y disminuir cotas superiores mediante inspección de restricciones
- Especialmente para variables enteras
- Puede asignar variables o detectar infactibilidades

Reforzamiento de cotas (i)

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

- Para la primera restricción

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

- Para la segunda restricción

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

- Para la primera restricción

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

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

Reforzamiento de cotas (ii)

- Sea la restricción

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

$$l_j \le x_j \le u_j$$

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

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

Reforzamiento de cotas (iii)

- Si una variable debe ser entera y sus cotas no lo son, éstas se pueden ajustar

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

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

Eliminación de restricciones redundantes (i)

- Si una restricción se satisface incluso en la situación más "difícil" la restricción puede ser eliminada.
- Detecta redundancia por medio de cotas de variables
- Para restricciones $\leq$ se igualan a cota superior las variables con coeficientes > 0 y a cota inferior las demás.
- Var binarias $\begin{array}{l} 0 \leq x_1 \leq 1 \\ 0 \leq x_2 \leq 1 \end{array}$ $\begin{array}{l} 3x_1 + 2x_2 \leq 6 \\ 3x_1 - 2x_2 \leq 3 \\ 3x_1 - 2x_2 \geq -3 \end{array}$ son redundantes
- Las restricciones suelen ser redundantes a consecuencia del proceso de asignación de variables o reforzamiento de cotas.

Eliminación de restricciones redundantes (ii)

- Sea la restricción

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

- Es redundante si

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

- Es infactible si

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

Asignación de variables (i)

- Identificar variables que pueden fijarse a uno de sus valores (0/1) ya que el otro no puede ser solución factible y óptima.
- Si un valor de una variable no puede satisfacer una restricción, aun cuando las demás variables tomen sus valores más favorables para intentar cumplirla, la variable debe fijarse al valor opuesto.

siendo
$$\begin{array}{l} 0 \le x_1 \le 1 \\ 0 \le x_2 \le 1 \\ 0 \le x_3 \le 1 \end{array} \qquad \left.\begin{array}{llll} 3x_1 & & & \le 2 \\ 3x_1 & +x_2 & & \le 2 \\ 5x_1 & +x_2 & -2x_3 & \le 2 \end{array}\right\} \Rightarrow x_1 = 0$$

Asignación de variables (ii)

- Procedimiento para restricciones $\leq$
  - ✓ Identificar la variable con el mayor coeficiente positivo. Si la suma de dicho coeficiente y cualquier coeficiente negativo excede la cota de la restricción, la variable debe fijarse a 0.

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

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

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

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

- Reacción en cadena: el procedimiento se repetirá para las siguientes variables con el mayor coeficiente.

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

Reducción de coeficientes (i)

- Reduce la región factible del problema LP sin eliminar soluciones factibles del problema BIP modificando los coeficientes de las restricciones.
- Procedimiento para restricciones $\leq$ $a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq b$

1. Calcular S = suma de valores a_j positivos
2. Elegir cualquier $a_j \neq 0$ tal que $S < b + |a_j|$
   - ✓ No existe $\Rightarrow$ no se puede ajustar más la restricción
   - ✓ $a_j > 0 \quad a_j' = S - b \quad b' = S - a_j \quad a_j = a_j' \quad b = b'$
   - ✓ $a_j < 0 \quad a_j' = S - b \quad a_j = a_j'$
3. Ir a 1.

Reducción de coeficientes (ii)

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

$S = 2 + 3 = 5$

$a_1 \ne 0 \qquad 5 < 4 + 2$

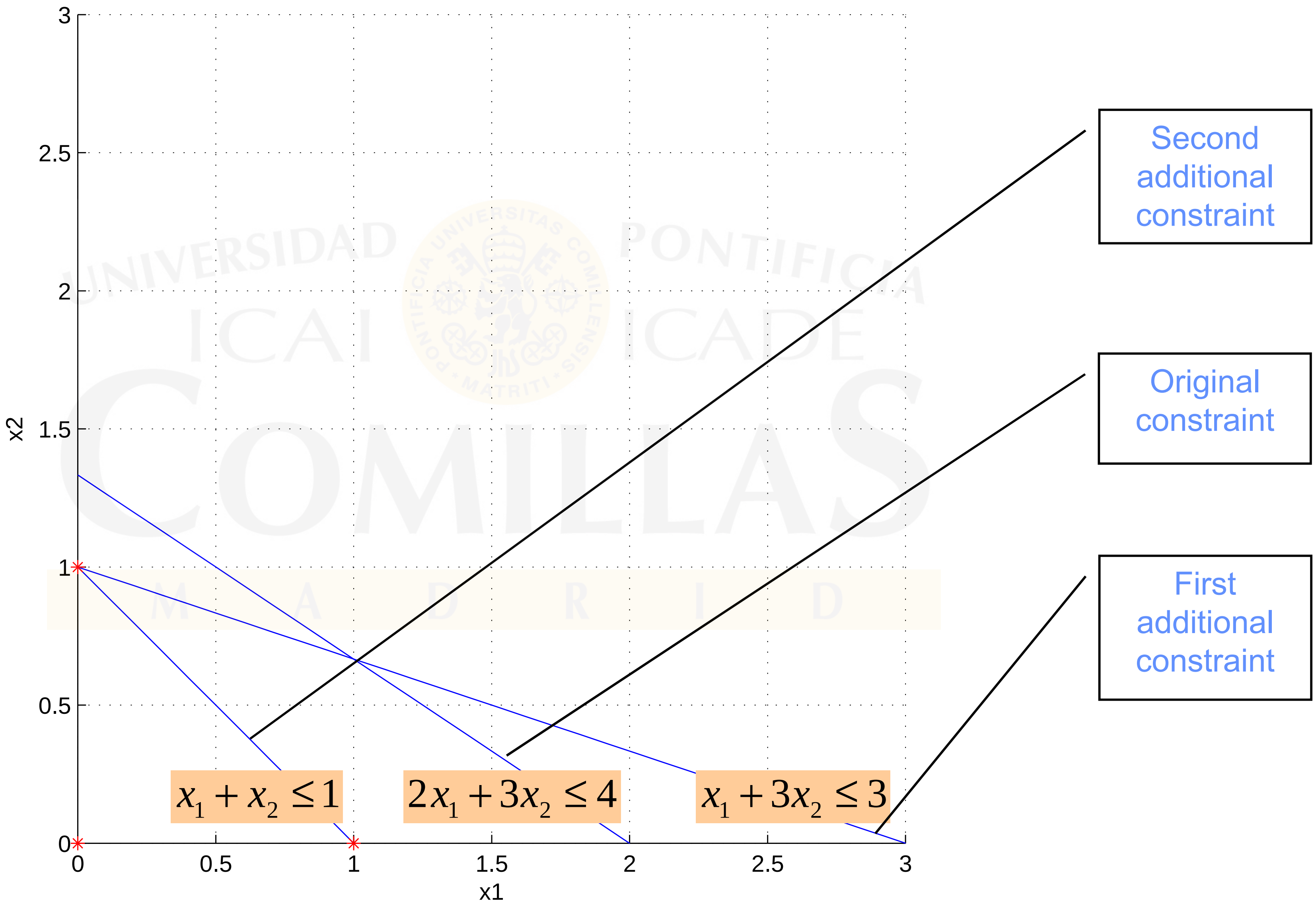
$a_1' = 5 - 4 = 1 \quad b' = 5 - 2 = 3 \quad a_1 = 1 \qquad b = 3 \qquad \boxed{x_1 + 3x_2 \le 3}$

$S = 1 + 3 = 4$

$a_2 \ne 0 \qquad 4 < 3 + 3$

$a_2' = 4 - 3 = 1 \quad b' = 4 - 3 = 1 \quad a_2 = 1 \qquad b = 1 \qquad \boxed{x_1 + x_2 \le 1}$

Reducción de coeficientes (iii)



CONTENIDO

- ❑ INTRODUCCIÓN
- ❑ MÉTODOS DE SOLUCIÓN
- ❑ MÉTODO DE RAMIFICACIÓN Y ACOTAMIENTO
- ❑ DUALITY (master)
- ❑ PREPROCESSING (master)
- ➢ BRANCH AND CUT METHOD (master)

Generación de planos de corte

- Nueva restricción que reduce la región factible del problema LP sin eliminar soluciones factibles del problema IP.
- Deducida válidamente de restricciones del problema

- Procedimiento de método de planos de corte
  1. Inicializar resolviendo el problema relajado LP
  2. Si la solución óptima es entera finalizar. Si no, continuar a paso 3
  3. Obtener un plano de corte que viole la solución óptima actual
  4. Añadir el plano de corte a las restricciones y reoptimizar. Ir a paso 2.

Planos de corte de tipo cubrimiento

- Considera cualquier restricción de tipo ≤ con variables binarias con todos los coeficientes no negativos (restricción tipo mochila)
- Encontrar grupo de variables (cubrimiento minimal) tal que
  - ✓ Se viola la restricción si las variables del cubrimiento son 1 y el resto son 0
  - ✓ Se satisface la restricción si una de las variables del cubrimiento se hace 0
- Formación del plano de corte $\sum \text{variables del cubrimiento} \leq N - 1$

siendo N el número de variables del cubrimiento

$$
\begin{array}{l}
6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 9 \\
x_i \in \{0,1\}
\end{array}
\qquad
\begin{cases}
x_1 + x_2 + x_4 \leq 2 \\
x_2 + x_3 + x_4 \leq 2 \\
x_1 + x_3 \leq 1
\end{cases}
$$

Planos de corte de Gomory (i)

- Sea el problema

$$\min_x c^T x$$
$$Ax = b$$
$$x \in \mathbb{Z}^+$$

- Sea una variable x_i no entera. La fila en la tabla del simplex es:

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

- El plano de corte de Gomory es:

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

o bien $$\sum_{t \in x_N} f_{it} x_t \ge f_i$$

siendo f_i y f_{it} son las partes fraccionales de $\hat{b}_i$ e y_{it}

- La variable de exceso del corte es entera si las var son enteras
- Este corte elimina la solución óptima no entera anterior

Planos de corte de Gomory (ii)

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

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

	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

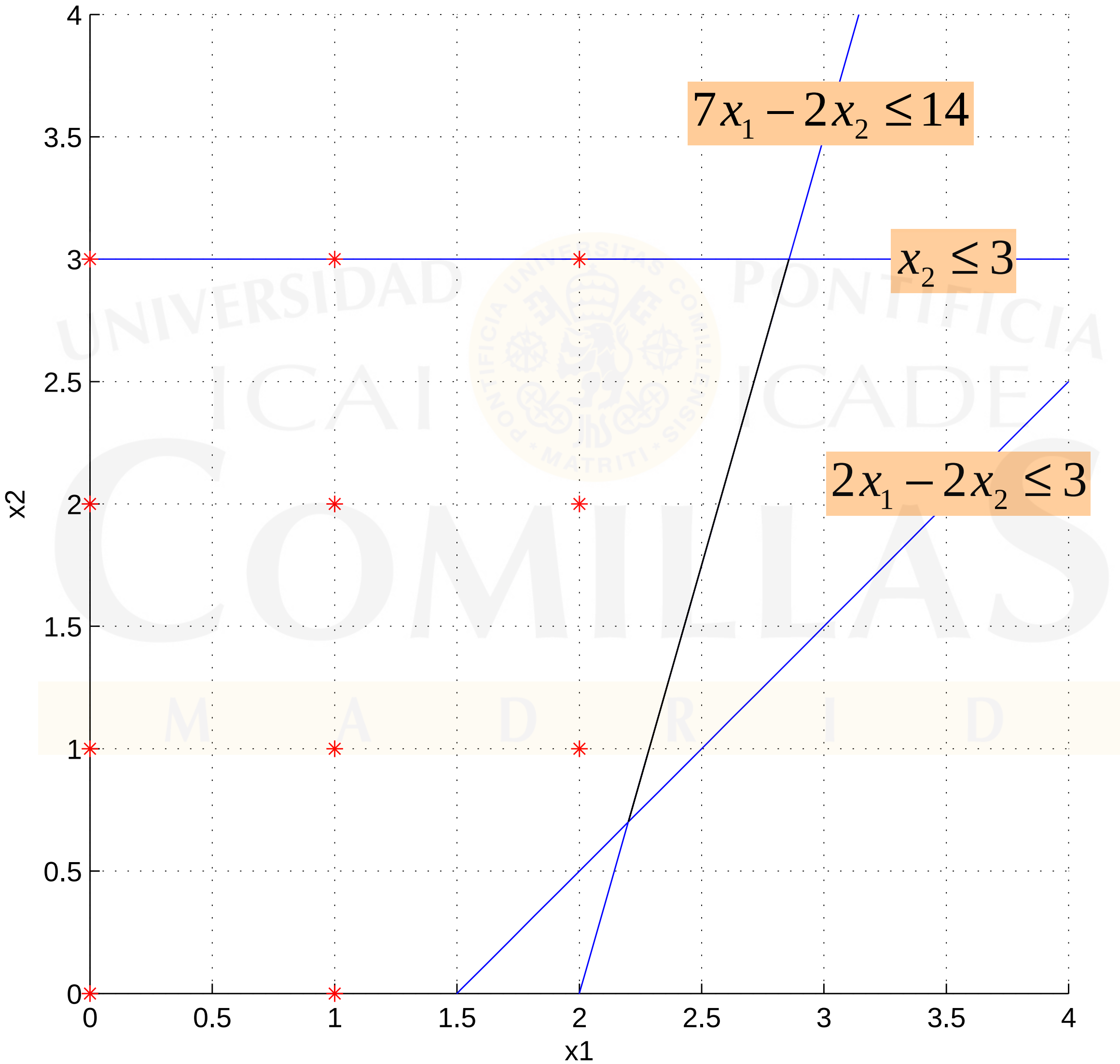
se introduce el corte $0.143h_1 + 0.286h_2 \ge 0.857$

	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

y luego el corte $0.5h_3 \ge 0.5$

que ya da lugar a la solución óptima entera

Planos de corte de Gomory (iii)



Planos de corte de Gomory (iv)

$$\boxed{0.143h_1 + 0.286h_2 \geq 0.857}$$

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

$$-7x_1 \geq -14$$

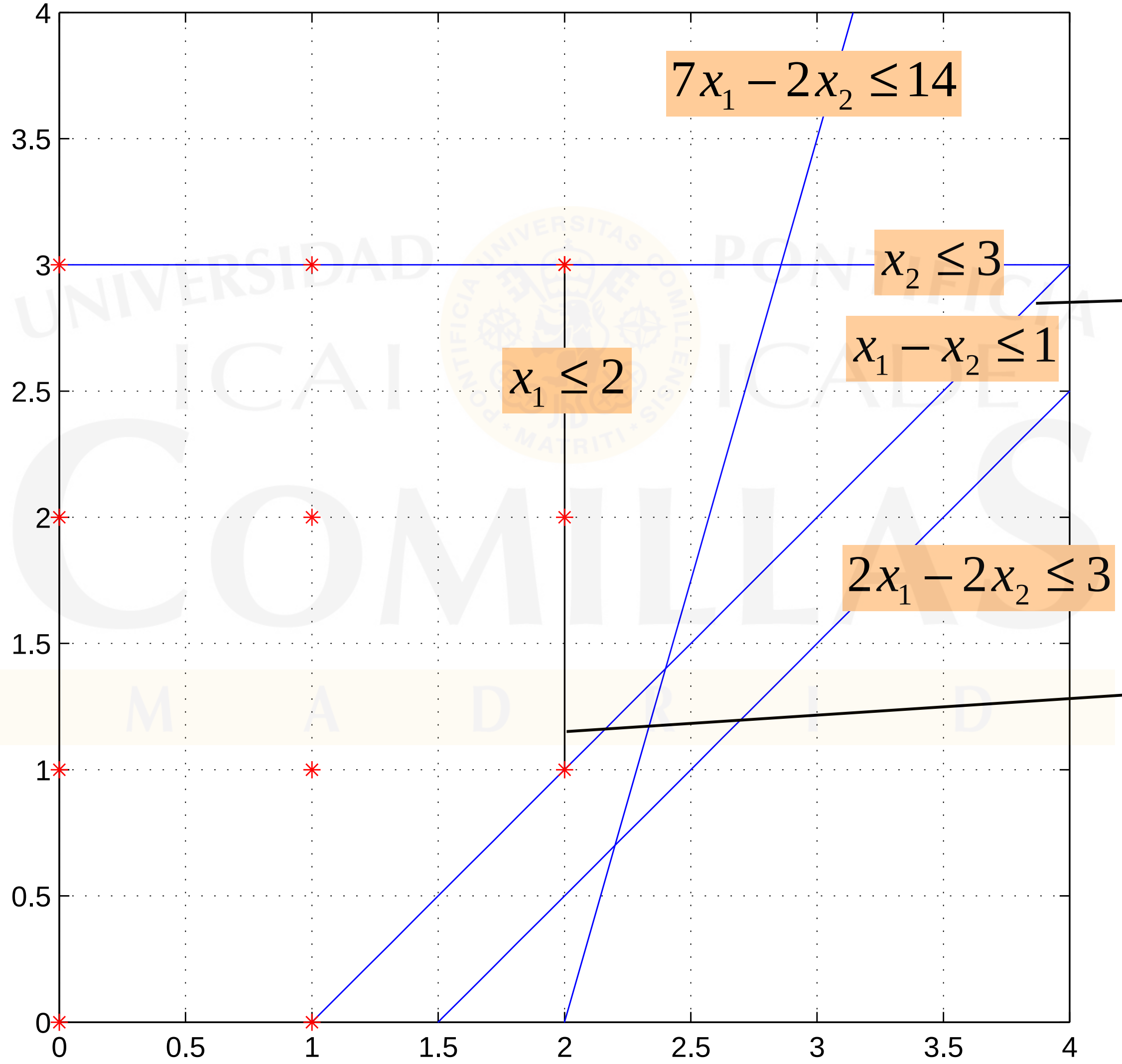
$$\boxed{x_1 \leq 2}$$

$$\boxed{0.5h_3 \geq 0.5}$$

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

$$\boxed{x_1 - x_2 \leq 1}$$

Planos de corte de Gomory (v)



Second Gomory cut

First Gomory cut

Método de ramificación y corte (*branch and cut*)

- Método de ramificación y acotamiento + Método de planos de corte en los nodos
- Disminuye mucho el tiempo de resolución
- Procedimiento
  - ✓ Elección de un nodo para evaluar (inicialmente el nodo raíz es el problema original relajado) y resolución
  - ✓ Decisión sobre generar o no planos de corte. Si se obtienen se añaden al problema y se resuelve éste.
  - ✓ Podar y ramificar con los criterios del método de ramificación y acotamiento.

CPLEX Performance Tuning for MIP

(http://www-01.ibm.com/support/docview.wss?uid=swg21400023)

- `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 0`
- `FpHeur 2`

Pure branch and bround

- `Cuts -1`
- `HeurFreq -1`

- 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 (http://www.gurobi.com/documentation/5.6/reference-manual/mip_models)

- ❑ 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`
- ❑ Heuristics
  - ✓ `Heuristics`, `SubMIPNodes`, `MinRelNodes`, `PumpPasses`, `ZeroObjNodes`
- ❑ Cutting Planes
  - ✓ `Cuts`, `FlowCoverCuts`, `MIRCuts`
- ❑ Presolve
  - ✓ `Presolve`, `PrePasses`, `AggFill`, `PreSparsify`

Andrés Ramos
Universidad Pontificia Comillas
http://www.iit.comillas.edu/aramos/
Andres.Ramos@comillas.edu