Facultad de Ingeniería
Departamento de Ciencias de la Ingeniería
Magíster en Logística y Gestión de Operaciones

Métodos de Optimización para la toma de decisiones

MLG-521

Programación Entera

1º Semestre 2017

Profesores:
Pamela Alvarez M.
Cristóbal Rojas.

Programación Lineal Entera

- En esta unidad veremos:
  - Generalidades.
  - Branch and Bound.
  - Robustecimiento.

Programación Lineal Entera

- Nuestro problema original es:

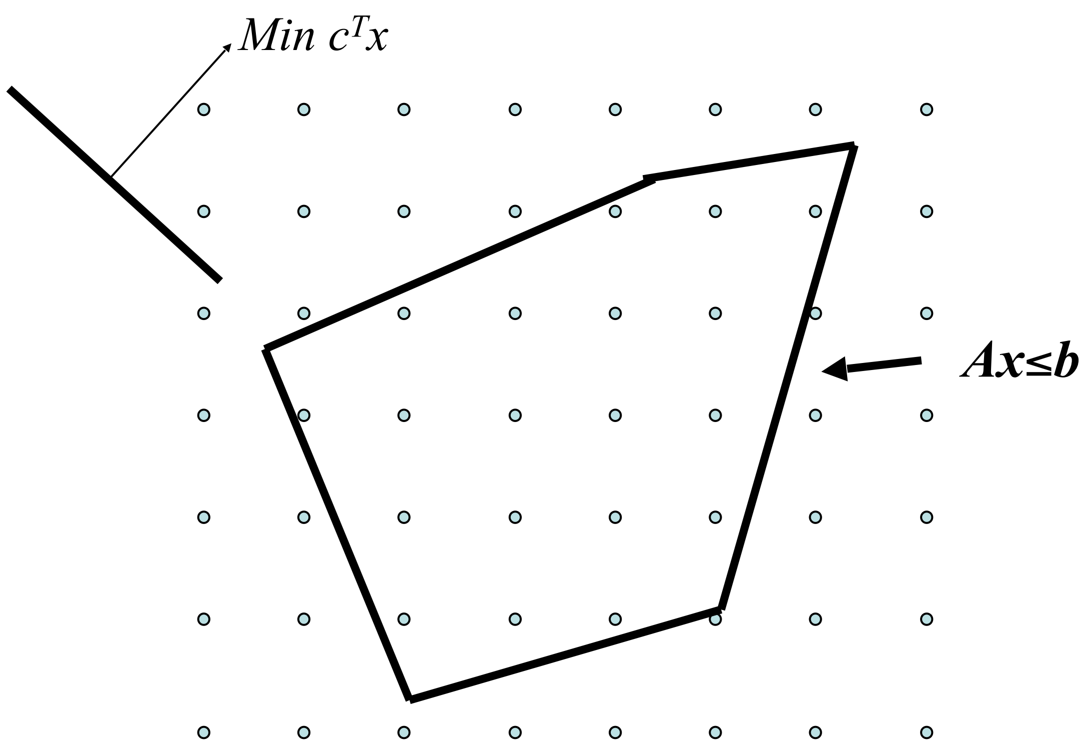
$$
PE) \qquad \begin{array}{ll} \min & c^T x \\ \text{s.a.} & Ax \leq b \\ & x \text{ entero} \end{array}
$$

- Sea P = { x: $Ax \leq b$ } el poliedro definido por las restricciones lineales.

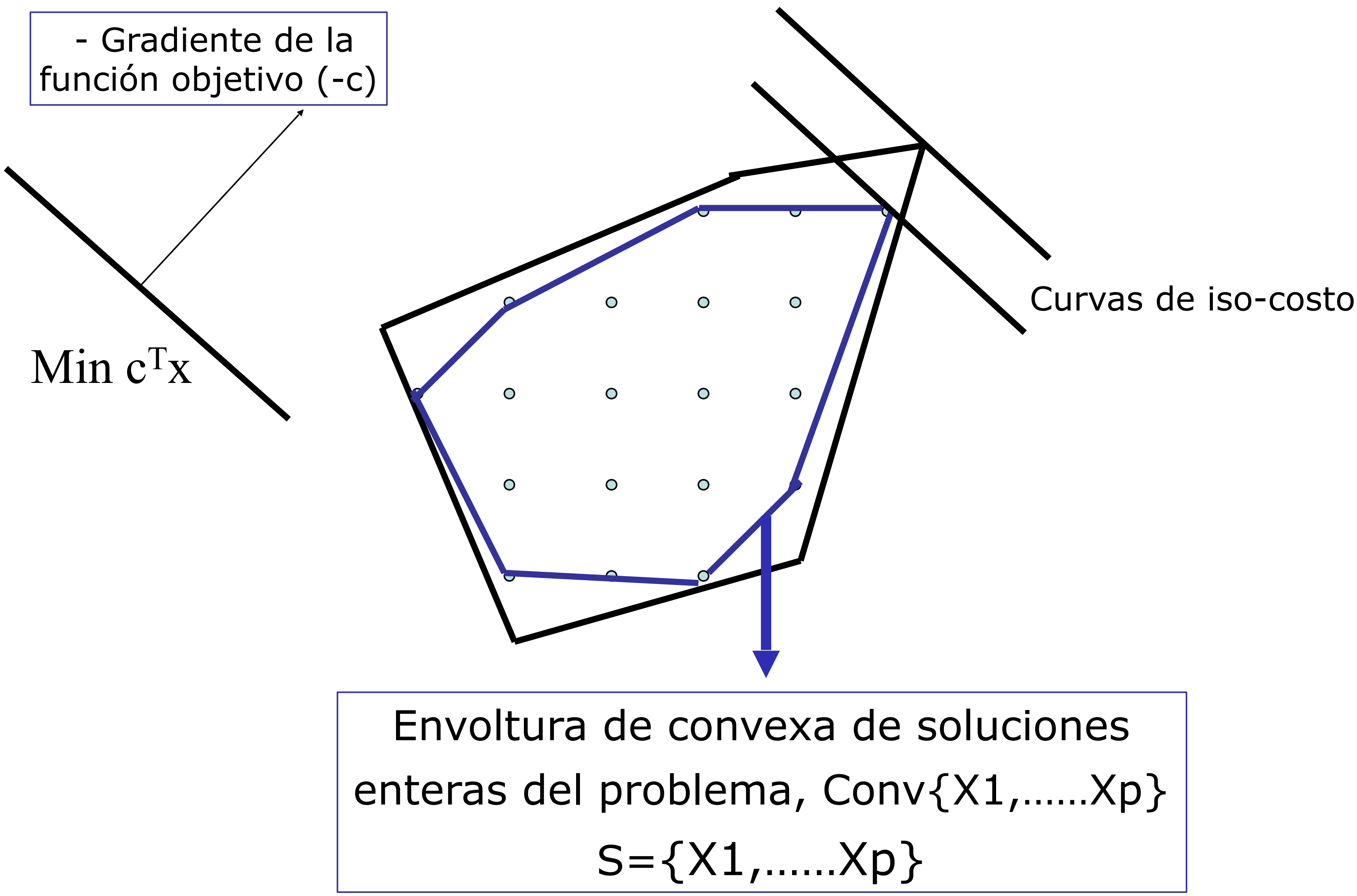
- Sea S = {x: $Ax \leq b$, x *entero* } el conjunto de soluciones factibles del problema entero.

Programación Lineal Entera

- Veamos P y S en la figura...



Programación Lineal Entera



Programación Lineal Entera

- Encontrar la formulación ideal es, en la gran mayoría de los casos, muy difícil.

- Pero la noción es importante, pues permite definir métodos prácticos con resultados satisfactorios:

  - Es posible construir **algunas** de las desigualdades que definen **facetas**

  - Construir desigualdades que, si bien no definen facetas, están muy “próximas” a la formulación ideal

  - Modificar iterativamente el conjunto de desigualdades que definen el conjunto P (aproximándose a la F. ideal)

- Esto es la base para la idea algorítmica de “planos cortantes” y métodos que consideran en el uso de desigualdades válidas y cortes.

Programación Lineal Entera

- Sea el problema:

$$
\begin{aligned}
PE)\quad z^* = \quad & Min \quad c^T x \\
& s.a. \quad Ax = b \\
& \qquad\quad x \geq 0 \\
& \qquad\quad x_j \in \{0,1\} \quad \forall j \in J \subset \{1,\ldots,n\}
\end{aligned}
$$

- **<u>Def.</u>**: El problema

$$
\begin{aligned}
PR)\quad z^0 = \quad & Min \quad c^T x \\
& s.a. \quad Ax = b \\
& \qquad\quad x \geq 0 \\
& \qquad\quad 0 \leq x_j \leq 1 \quad \forall j \in J \subset \{1,\ldots,n\}
\end{aligned}
$$

Se llama RELAJACION LINEAL del problema entero mixto binario original.

Programación Lineal Entera

- Veamos los siguientes ejemplos:

$$
\begin{array}{lll}
PE) & Max & z = 21x_1 + 11x_2 \\
 & s.a. & 7x_1 + 4x_2 \leq 13 \\
 & & x_1, x_2 \geq 0 \\
 & & x_1, x_2 \text{ enteros}
\end{array}
$$

$$
\begin{array}{lll}
PE) & Max & z = 4x_1 + x_2 \\
 & s.a. & 2x_1 + 3x_2 \leq 5 \\
 & & 2x_1 + 3x_2 = 5 \\
 & & x_1, x_2 \geq 0 \\
 & & x_1, x_2 \text{ enteros}
\end{array}
$$

$$
\begin{array}{lll}
PE) & Max & z = 4x_1 + x_2 \\
 & s.a. & x_1 \leq 5 \\
 & & x_2 \leq 3 \\
 & & x_1, x_2 \geq 0 \\
 & & x_1, x_2 \text{ enteros}
\end{array}
$$

Programación Lineal Entera

- Relación entre ambos problemas....

$$\left.\begin{array}{llll} PE) & Min & z_E = c^T x \\ & s.a. & Ax = b \\ & & x \geq 0, enteros \\ \\ PR) & Min & z_R = c^T x \\ & s.a. & Ax = b \\ & & x \geq 0 \end{array}\right\} \quad \boxed{z_E^* \geq z_R^*}$$

Programación Lineal Entera

- Entonces …. ¿Cómo resolver los problemas enteros?
- Una primera idea intuitiva es resolver la relajación lineal y luego aproximar o redondear.
- Eventualmente se pueden construir métodos un poco más elaborados de redondear inteligentemente.
- Agregar “cortes” o restricciones que eliminen las soluciones fraccionales obtenidas.
- Construir iterativamente restricciones que eliminen las soluciones fraccionales, de modo de ir acercándose a un óptimo entero …… exploremos más esta idea.

Programación Lineal Entera

- Sea el problema:

$$PE)\quad z = \min\ c^T x$$
$$s.a:\ Ax \le b$$
$$x \ge 0 \text{ entero}$$

- Y su relajación lineal

$$PR)\quad z = \min\ c^T x$$
$$s.a:\ Ax \le b,\ x \ge 0$$

- Sea $R^0 = \{x: Ax \le b, x \ge 0\}$

- Sea x^0 una solución óptima de PR, es decir, $z^0 = c^T x^0$.

Programación Lineal Entera

- ¿Qué pasa si x^0 es entero…?

- En caso contrario, sea k un índice tal que x^0_k es fraccionario (no siempre todas las variables resultan fraccionarias).

- Idea Básica de B&B: Dada una solución optima fraccional de PR, x^0, dividir el problema en dos sub-problemas que NO contienen dicha solución fraccional:

$$R_0^+ = R_0 \cap \left\{ x : x_k \geq \lfloor x_k \rfloor + 1 \right\}$$
$$R_0^- = R_0 \cap \left\{ x : x_k \leq \lfloor x_k \rfloor \right\}$$

Programación Lineal Entera

- La idea es repetir este proceso, obteniendo sucesivamente subproblemas para regiones más y más acotadas.
- De esta forma se construye un “árbol”, donde cada nodo es un subproblema que contiene las restricciones que se van incorporando.

- ¿Qué puede ocurrir?
  - Alguna rama es un problema infactible, en cuyo caso se termina o se “corta” dicha rama.
  - Alguna rama es un problema cuya solución es automáticamente entera. Esto entrega una cota superior para el valor óptimo del problema (caso min).
    - No es necesario seguir explorando dicho árbol.
    - ¿Se dispone alguna cota inferior?
  - Alguna rama entrega una nueva solución fraccional…

Programación Lineal Entera

- Debemos ir guardando los valores óptimos de cada problema ya que nos sirven como cotas
- **DEFINICION:** El valor de la MEJOR solución entera disponible en cada iteración se llama **INCUMBENTE.**
- Este es el algoritmo de Ramificación y Acotamiento
- Si el problema original es de tipo BINARIO, entonces algunas cosas se simplifican un poco…

- **En resumen: Método: Ramificación y Acotamiento o Branch & Bound**

1. Inicialización:
   - Establecer cota superior (∞) e inferior (-∞)
   - Resolver el PR
     - Casos
   - Actualizar cota inferior
2. Ramificación:
   - Variable x_k no entera (y debiera serlo)
   - Si x_k=a.b se generan 2 subproblemas a los que se le agregan las siguientes restricciones respectivamente:
     - $x_k \leq$ a
     - $x_k \geq$ a + 1

Programación Lineal Entera

- **Método: Ramificación y Acotamiento o Branch & Bound**

  3. Solución subproblema correspondiente.
  4. Actualización de cotas:
     - Casos
  5. Poda:
     - Por integralidad
     - Por cotas
     - Por infactibilidad
  6. Optimalidad:
     - ¿Quedan subproblemas por resolver?

Programación Lineal Entera

- Se tiene el siguiente problema:
  - Una empresa procesa uranio (x) y plutonio (y) para vender a centrales nucleares como combustible, debe decidir cuánto producir de cada uno durante el próximo mes, de tal forma de obtener la mayor ganancia
  - El precio de venta del uranio es de US$ 1.000.000 por kilo, mientras que el del plutonio es US$ 2.000.000
  - La producción de un kilo de cada uno de estos combustibles le toma a la empresa 1 semana
  - Además, debido a la materia prima disponible, calculan que no podrán procesar más de 2,8 Kg. de plutonio durante dicho mes
  - Debido al sistema de turnos con el que trabaja la planta, únicamente se puede cambiar el producto que están procesando los días domingo (es decir, deben estar una semana completa procesando el mismo producto antes de poder cambiarlo)

Programación Lineal Entera

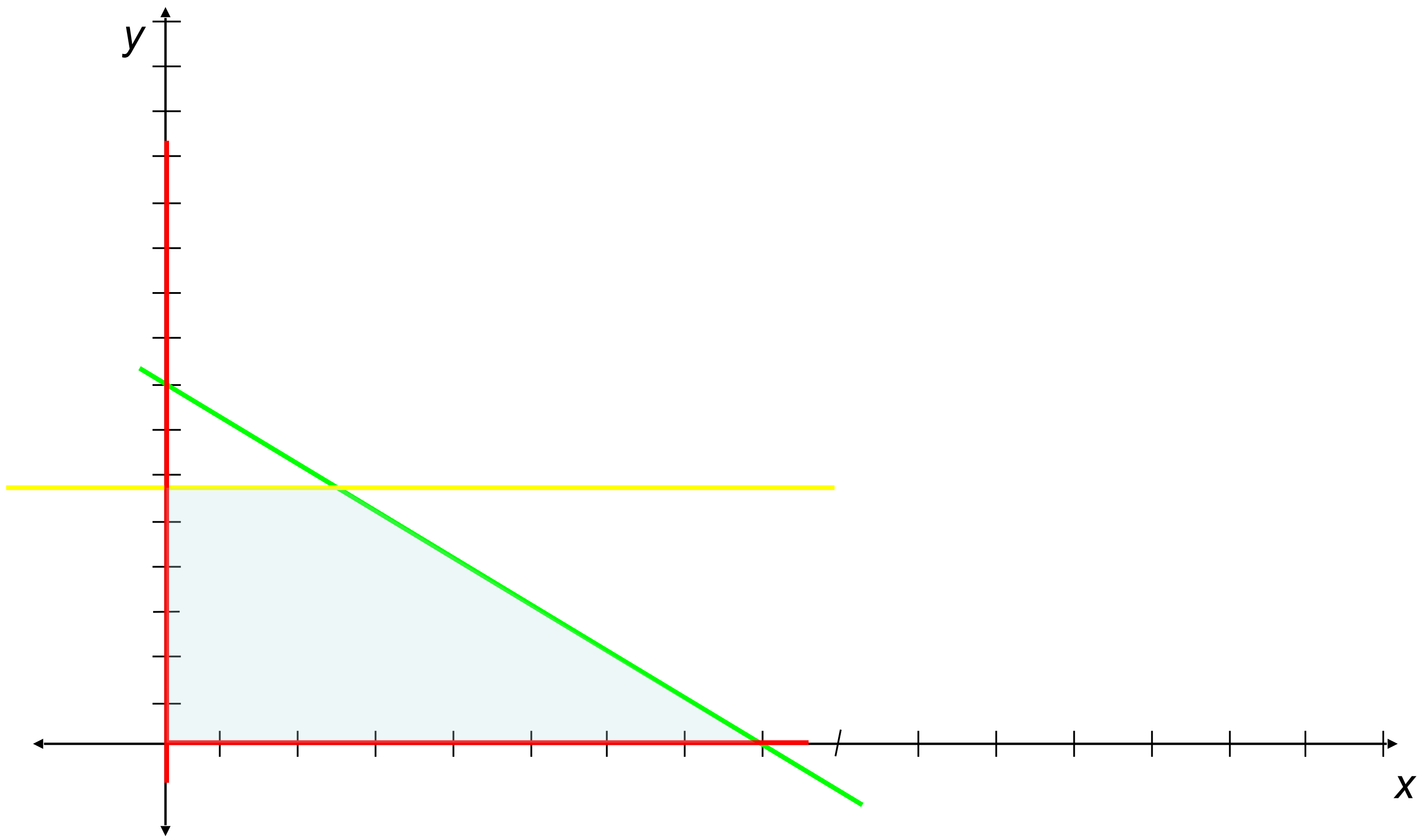
UNIVERSIDAD ANDRES BELLO

x = Semanas produciendo uranio a producir durante un mes

y = Semanas produciendo plutonio a producir durante un mes

$$
\begin{array}{ll}
Max & x + 2y \\
s.a & x + y \leq 4 \\
 & y \leq 2,8 \\
 & x, y \in \mathbb{Z}
\end{array}
$$

Programación Lineal Entera



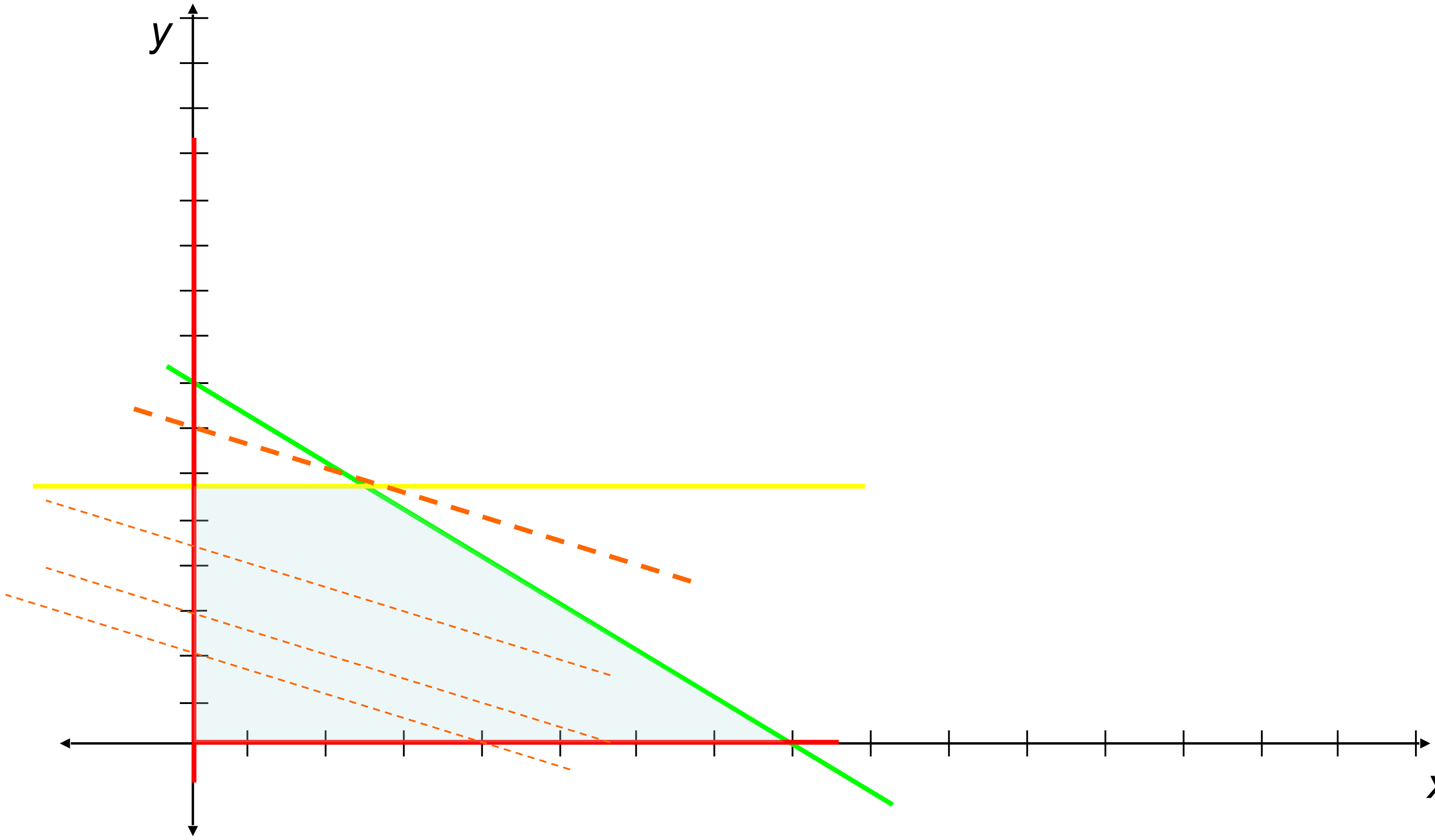
Programación Lineal Entera

- Problema Relajado Lineal

$$
\begin{array}{lll}
P1) & Max & x + 2y \\
 & s.a & x + y \leq 4 \\
 & & y \leq 2,8 \\
 & & x, y \geq 0
\end{array}
$$

Programación Lineal Entera

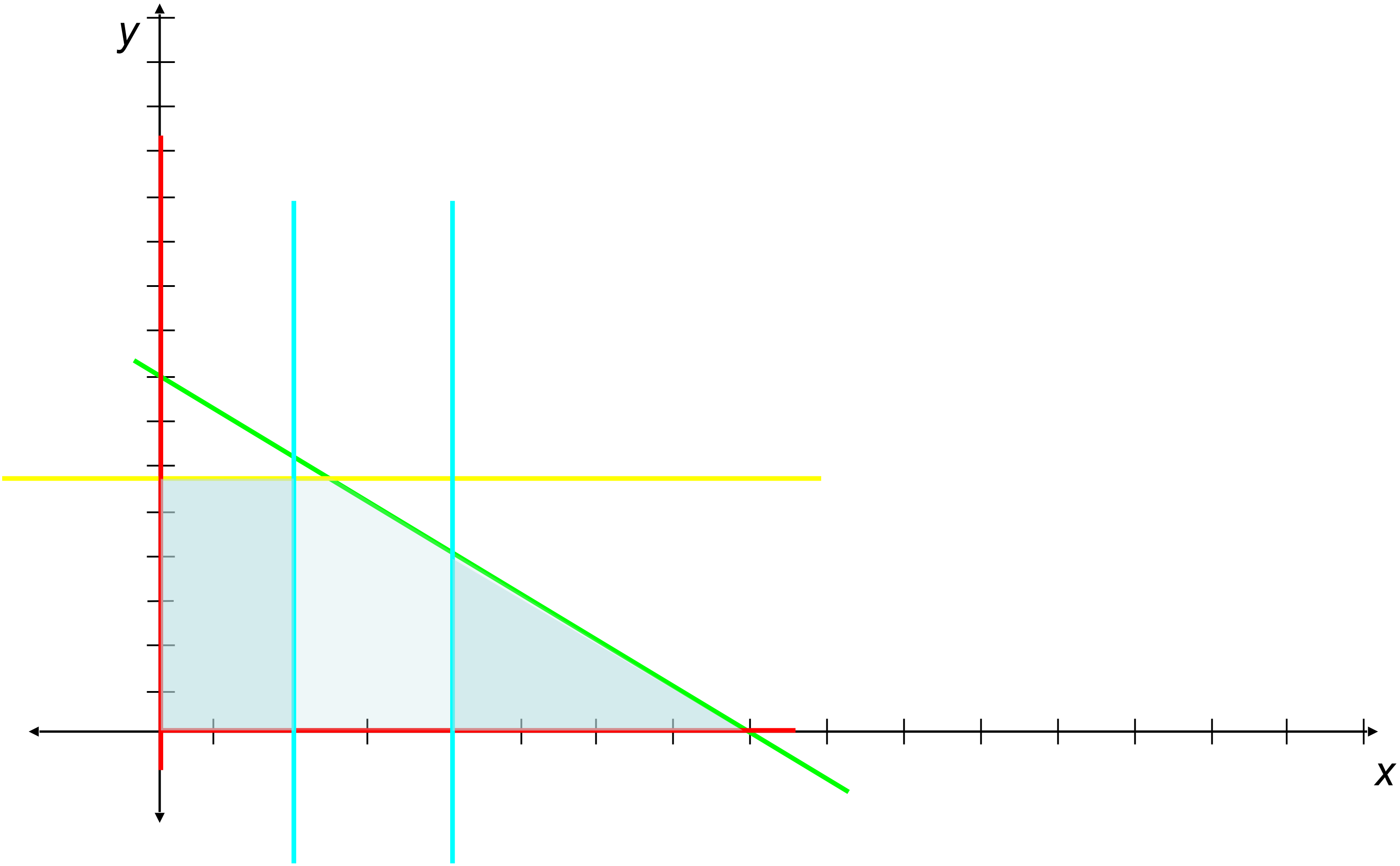
- Problema Relajado Lineal



Programación Lineal Entera

- La solución óptima del problema relajado lineal es x=1,2, y=2,8
- El valor de la función objetivo en este caso es 6,8
- Claramente el problema no es entero
- Para encontrar una solución entera, se procede a “aislar” la solución obtenida

- Problema Relajado Lineal



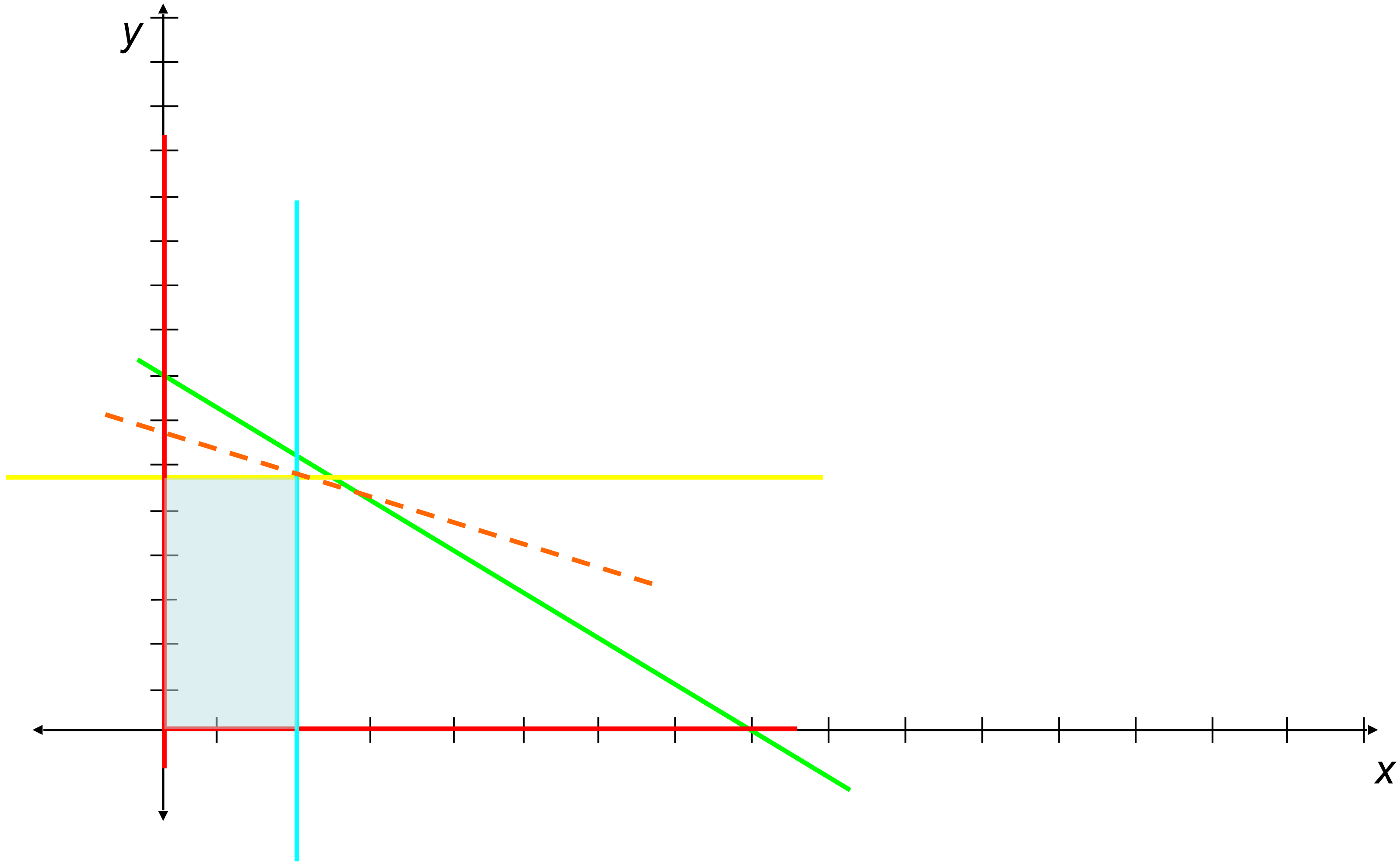
Programación Lineal Entera

- Es decir, ahora se tienen dos problemas

$$
\begin{array}{llll}
P2) & Max & x + 2y \\
 & s.a & x + y \leq 4 \\
 & & y \leq 2,8 \\
 & & x \leq 1 \\
 & & x, y \geq 0
\end{array}
\qquad
\begin{array}{llll}
P3) & Max & x + 2y \\
 & s.a & x + y \leq 4 \\
 & & y \leq 2,8 \\
 & & x \geq 2 \\
 & & x, y \geq 0
\end{array}
$$

- Nuevamente, se deben resolver ambos problemas

- Problema 2

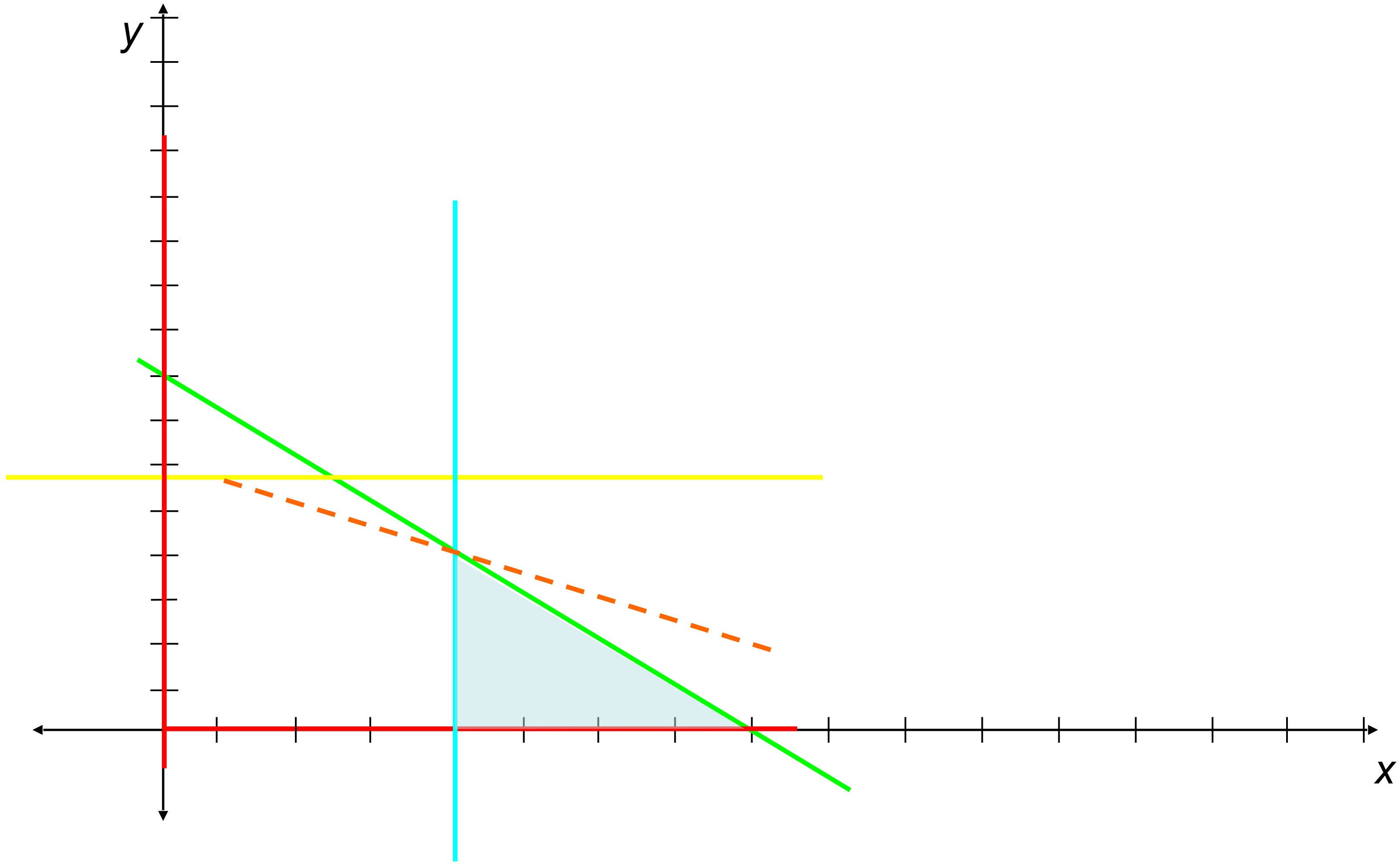


Programación Lineal Entera

- La solución óptima del P2 es x=1, y=2,8. el valor de la función objetivo es 6,6
- ¿Cómo varió la función objetivo respecto a P1? ¿Por qué?

Programación Lineal Entera

- Problema 3



Programación Lineal Entera

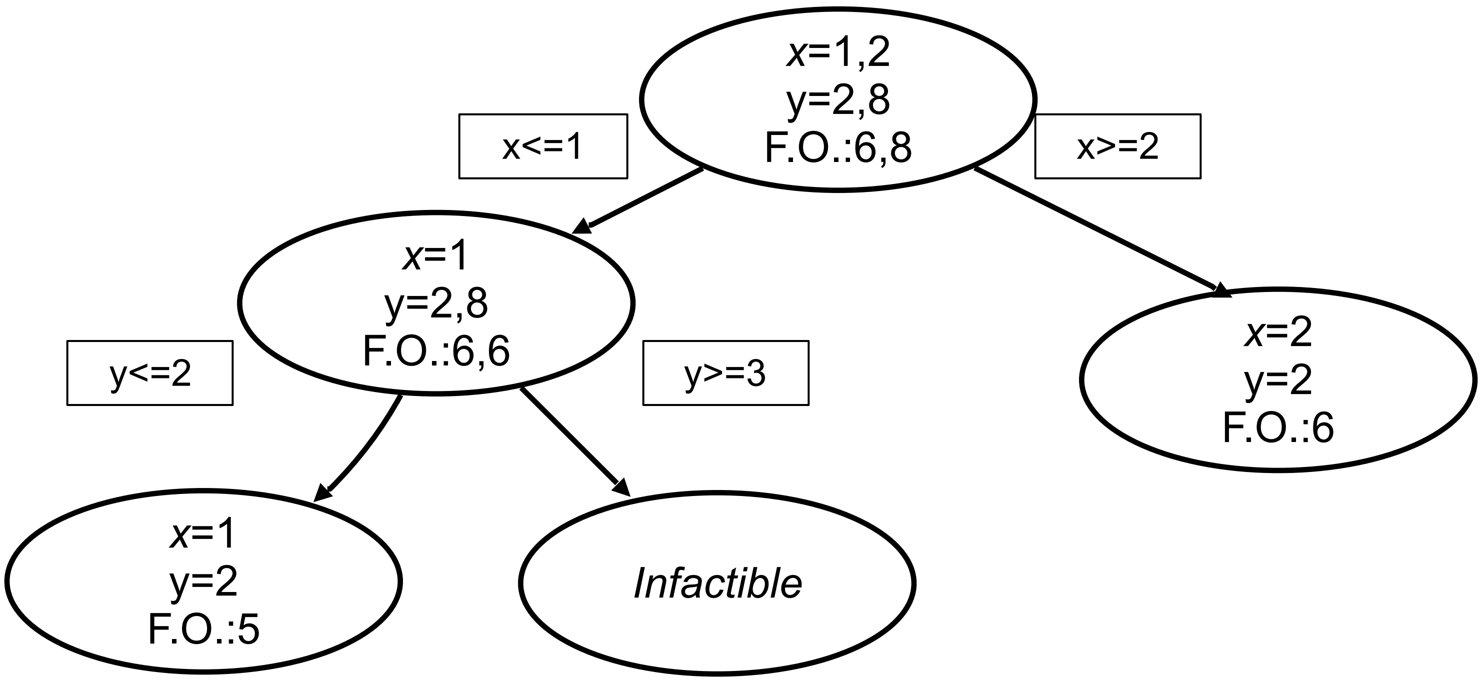
- La solución óptima del P3 es x=2, y=2 el valor de la función objetivo es 6
- ¿Cómo varió la función objetivo respecto a P1? ¿Por qué?
- ¿Hasta cuándo se debe acotar?

Programación Lineal Entera

- El algoritmo B&B tiene 3 criterios de parada en una "rama":
  - Si la solución encontrada en la rama es entera
  - Si el valor objetivo de la solución de una rama es menor o igual al valor de otra rama donde se ha encontrado una solución entera
  - Si el problema a resolver en una rama no tiene solución

Programación Lineal Entera

- El problema analizado quedaría de la siguiente forma:



Programación Lineal Entera

- Resolver por el método B&B el siguiente problema:

$$
\begin{array}{lll}
P) & Max & 7x+9y \\
 & s.a & x+y \le 8 \\
 & & 6x+11y \le 66 \\
 & & x,y \ge 0 \\
 & & x,y \quad \text{enteros}
\end{array}
$$

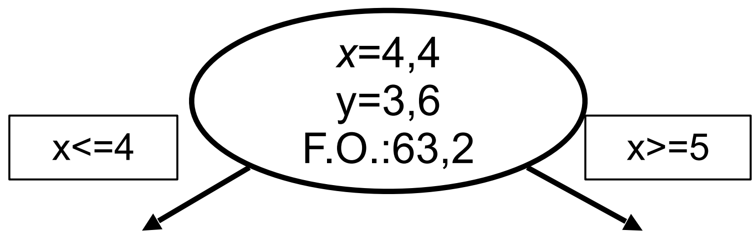
Programación Lineal Entera

- En primer lugar se resuelve la relajación lineal de este problema

$$
\begin{array}{lll}
PRL) & Max & 7x + 9y \\
 & s.a & x + y \leq 8 \\
 & & 6x + 11y \leq 66 \\
 & & x, y \geq 0
\end{array}
$$

Programación Lineal Entera

- El problema analizado quedaría de la siguiente forma:



Programación Lineal Entera

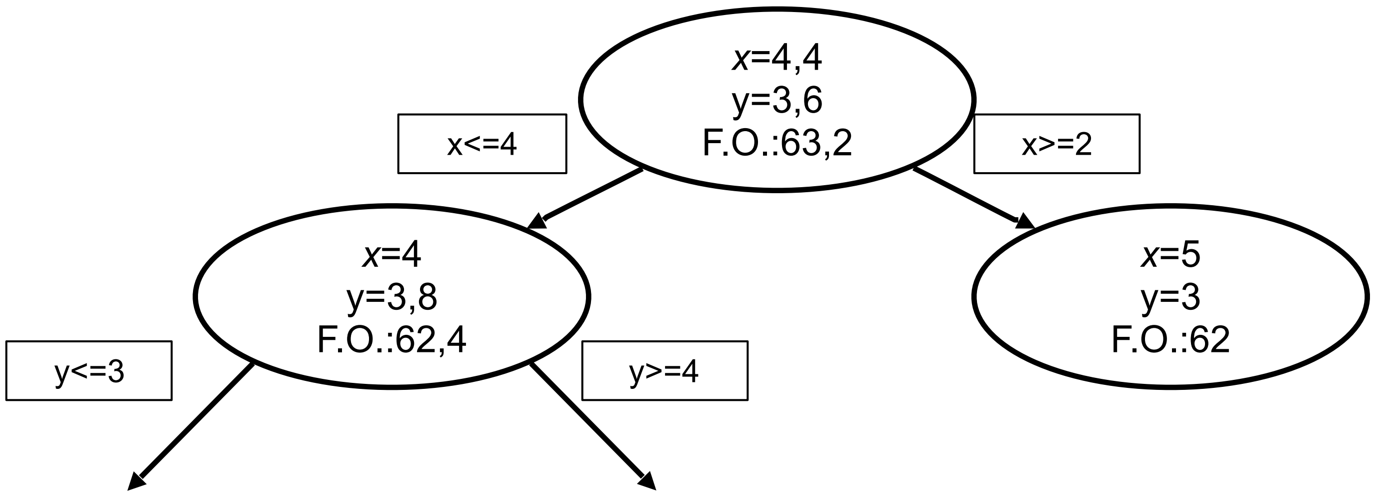
- Al "aislar" la variable *x*, se obtienen dos "ramas"

$$
\begin{aligned}
P1) \quad & Max \quad && 7x + 9y \\
& s.a && x + y \leq 8 \\
& && 6x + 11y \leq 66 \\
& && x \leq 4 \\
& && x, y \geq 0
\end{aligned}
$$

$$
\begin{aligned}
P2) \quad & Max \quad && 7x + 9y \\
& s.a && x + y \leq 8 \\
& && 6x + 11y \leq 66 \\
& && x \geq 5 \\
& && x, y \geq 0
\end{aligned}
$$

Programación Lineal Entera

- El problema analizado quedaría de la siguiente forma:



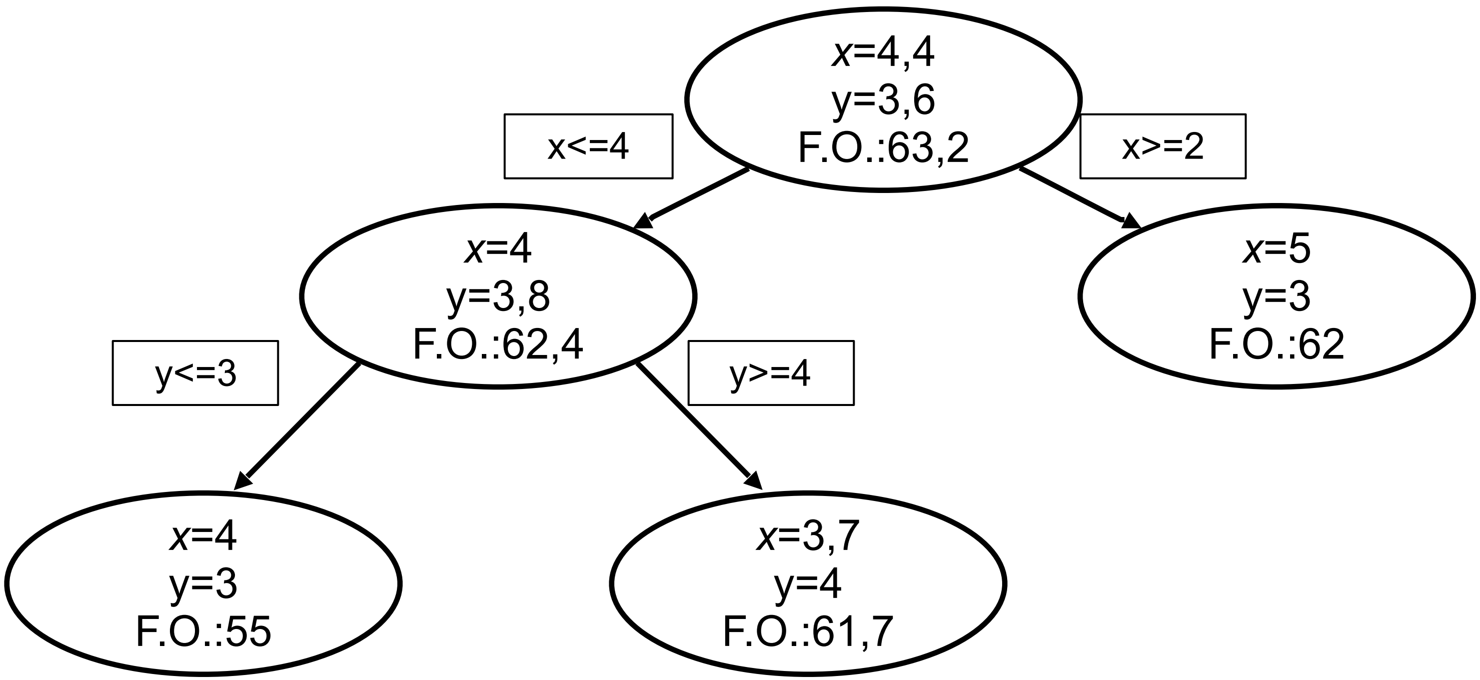
- Al ramificar el *P*2) se obtiene:

$$
\begin{array}{lll}
P21) & Max & 7x+9y \\
 & s.a & x+y \leq 8 \\
 & & 6x+11y \leq 66 \\
 & & x \leq 4 \\
 & & y \leq 3 \\
 & & x, y \geq 0
\end{array}
$$

$$
\begin{array}{lll}
P22) & Max & 7x+9y \\
 & s.a & x+y \leq 8 \\
 & & 6x+11y \leq 66 \\
 & & x \leq 4 \\
 & & y \geq 4 \\
 & & x, y \geq 0
\end{array}
$$

Programación Lineal Entera

- El problema analizado quedaría de la siguiente forma:



Programación Lineal Entera

- Por lo tanto, la solución óptima al problema será: $(x,y)=(5,3)$
- El valor óptimo es 62

Robustecimiento

- Robustecer la formulación de manera "iterativa"

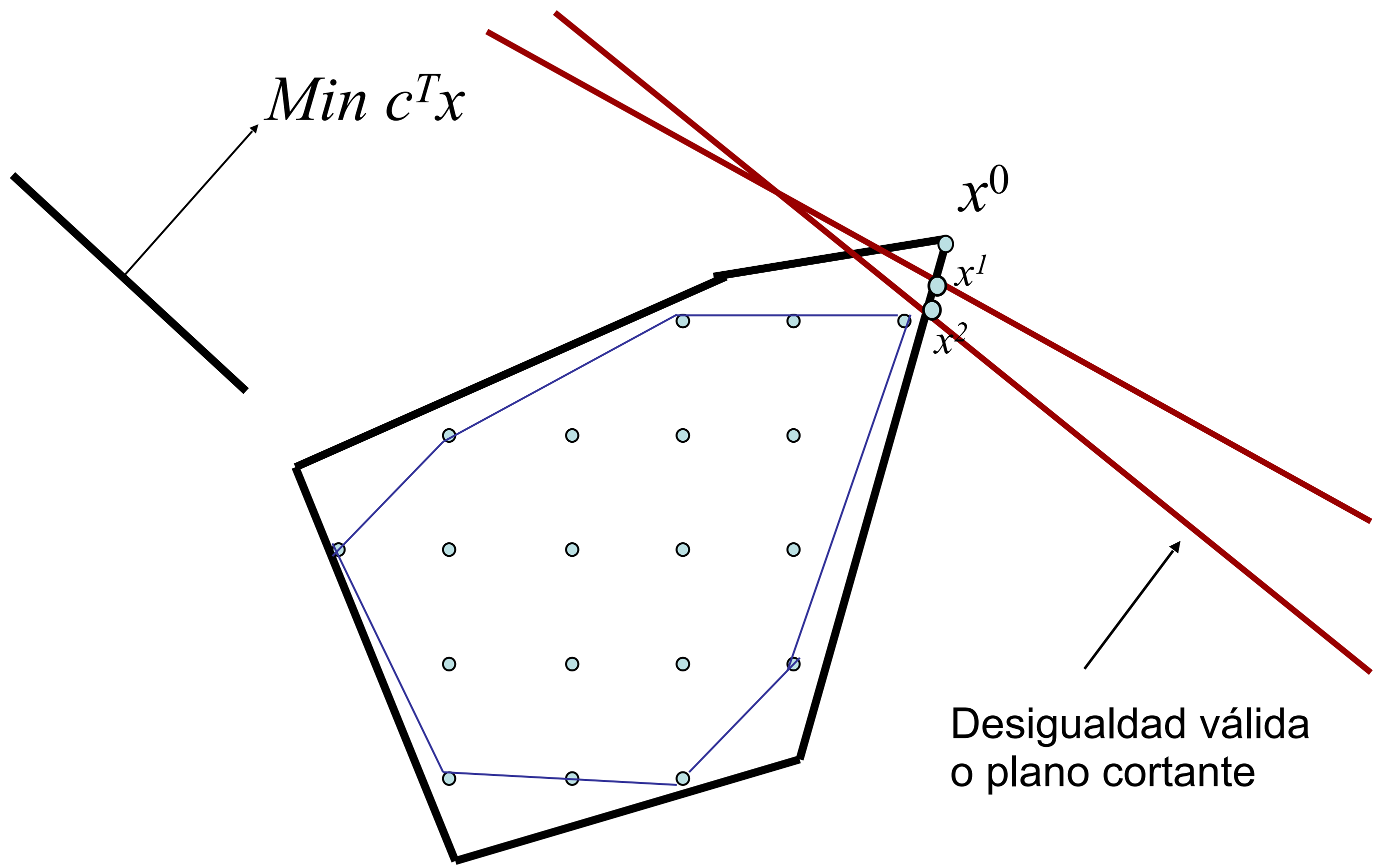
- Consideramos el problema entero:

$$\begin{array}{ll} Min & c^T x \\ s.a. & Ax \leq b \\ & x \ \text{entero} \end{array}$$

- Como siempre, S = { $x : A\,x \leq b,\ x$ entero } .

Robustecimiento

- Recordemos....



Robustecimiento

- El algoritmo de planos cortantes sería:

- Sea $S^0 = \{ x : Ax \leq b \}$, k = 0.

- Etapa general k:
  - Resolver el prob. lineal $Min\ \{c^T x : x \in S^k \}$.
  - Sea x^k sol. óptima.
  - Si x^k es entero, **PARAR**, tenemos una solución óptima.

- Si no, **buscar una desigualdad válida** $\boldsymbol{\alpha_k^T x \leq \beta_k}$ **tal que** $\boldsymbol{\alpha_k^T x^k > \beta_k}$.

- $S^{k+1} = S^k \cup \{ x : \alpha_k^T x \leq \beta_k \}$.

- $k \leftarrow k + 1$, ir a 1.

Robustecimiento

- ¿Cómo construir **desigualdades válidas?**

- **Restricciones del tipo “Knapsack”**

$$3x_1 + 5x_2 \leq 10 \qquad\qquad x_i \in \{0,1\}$$

- ¿Cuál sería una desigualdad válida?

- ¿Se pueden construir otro tipo de desigualdades para este tipo de restricciones?
  - COVER

Robustecimiento

- **COVER**

- Consideremos un conjunto tipo "knapsack" de la forma:

$$X = \{x \in \{0,1\}^n : \sum_{j=1}^{n} a_j x_j \leq b\}$$

- Donde que a_j > 0, y que b >0.
- Sea N = {1,2,…,n}
- **Definición:** C $\subseteq$ N se llama COVER si:

$$\sum_{j \in C} a_j > b$$

Robustecimiento

- **COVER**

- Si C ⊆ N es un COVER, entonces se puede construir la siguiente desigualdad válida:

$$\sum_{j \in C} x_j \leq |C| - 1$$

- Ejemplo:
  - Dada la siguiente restricción:

$$11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19, \quad x_i \in \{0,1\}, \forall i$$

- ¿Cuáles son cover y cuáles son sus desigualdades válidas asociadas?

- ¿Se ve algo más? ......

Robustecimiento

- Veamos por ejemplo el cover C={3,4,5,6}

- ¿Se puede extender el cover a otras variables?

- Si fijamos $x_2 = x_7 = 0$, entonces ¿para qué valores de un parámetro α_1 la siguiente desigualdad sigue siendo válida?

$$\alpha_1 x_1 + x_3 + x_4 + x_5 + x_6 \leq 3$$

Robustecimiento

- Se repite para las otras variables
- En general, sea I $\subseteq$ N-C y supongamos que se ha obtenido la desigualdad válida

$$\sum_{j\in I} \alpha_j x_j + \sum_{j\in C} x_j \leq |C| - 1$$

- Sea k $\in$ N - C , k $\notin$ I. Para calcular el mayor valor de α_k resolvemos un problema de mochila….

- Este procedimiento se llama “**levantamiento**” o “**lifting**”

Robustecimiento

- ¿Cómo identificar un COVER en un problema real?

- Supongamos que tenemos una solución actual del problema, x^* que NO es entera.
- Observemos primero que si C es un Cover, la desigualdad se puede rescribir:

- Queremos encontrar una desigualdad de Cover que NO sea satisfecha por el punto x^*
- Es decir, identificar un conjunto $C \subseteq N$ tal que :

- Escribámoslo como un problema de optimización y veamos qué pasa

Robustecimiento

- **Cortes de Gomory**
  - Consideremos el problema en forma estándar:

$$\begin{aligned} Max \quad & c^T x \\ s.a \quad & Ax = b \\ & x \geq 0 \text{ entero} \end{aligned}$$

  - Tenemos una solución óptima de la relajación lineal, x^0.
  - A se particiona en: $A = [\, B \mid N \,]$,
  - x^0 se particiona en $x_B^0 \geq 0$ y $x_N^0 = 0$.

Robustecimiento

- **Cortes de Gomory**
  - Sea NB el conjunto de índices de las variables **no básicas.**
  - Entonces en el óptimo se tiene:

$$x_{Bi}^{0} + \sum_{j \in NB} \overline{a}_{ij} x_j^{0} = \overline{b}_i \quad , i = 1, \ldots, m$$

  - Si la solución óptima **no es entera**, entonces existe un índice k tal que $\overline{b}_k$ es **fraccionario.**
  - **Obtengamos una desigualdad válida a partir de la ecuación k**

Robustecimiento

- **Cortes de Gomory**

- ¿Satisface x^0 esta restricción…?

$$x_{B_k} + \sum_{j \in NB} \lfloor \bar{a}_{kj} \rfloor \cdot x_j \leq \lfloor \bar{b}_k \rfloor$$

Reemplazando X_{Bi}
de la ecuación original

$$\sum_{j \in k} \left(a_{kj} - \lfloor \bar{a}_{kj} \rfloor \right) \cdot x_j \geq b_k - \lfloor \bar{b}_k \rfloor$$

- Gomory demostró que este procedimiento termina, después de generar un número FINITO de cortes, con una solución óptima entera o un problema infactible (caso en el cual el problema original es infactible).

- Ejemplo…

Robustecimiento

- **Branch and Cut**

- Se puede eficientar el algoritmo de B&B agregando cortes

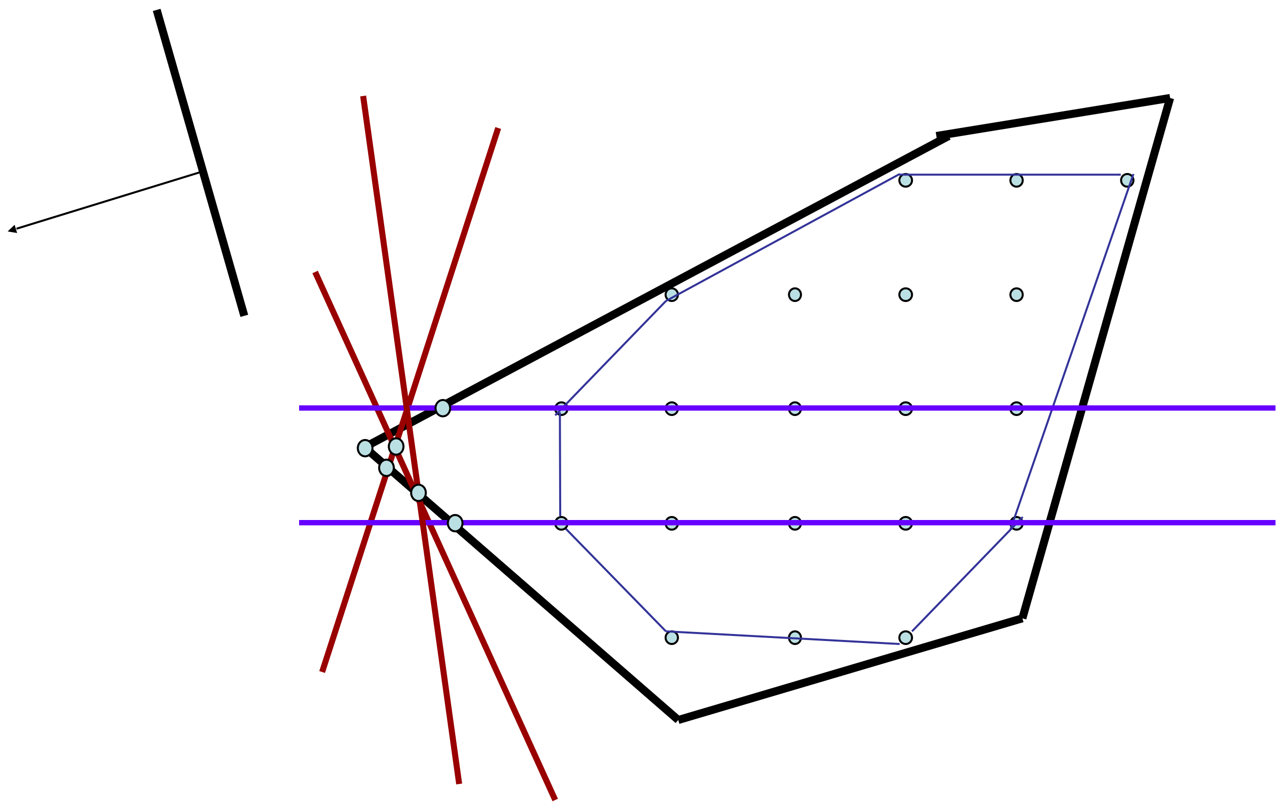
$$
PE) \qquad \begin{array}{rl} z = Min & c^T x \\ s.a. & Ax \leq b \\ & x \geq 0 \text{ entero} \end{array}
$$

- Luego de resolver la relajación lineal se tiene una solución x_k fraccionaria
- Se tienen 2 subproblemas.
- En cada subproblema se pueden agregar cortes, ¿para qué?

Robustecimiento

- **Branch and Cut**

$$Min\ c^T x$$



Robustecimiento

- **Branch and Cut**