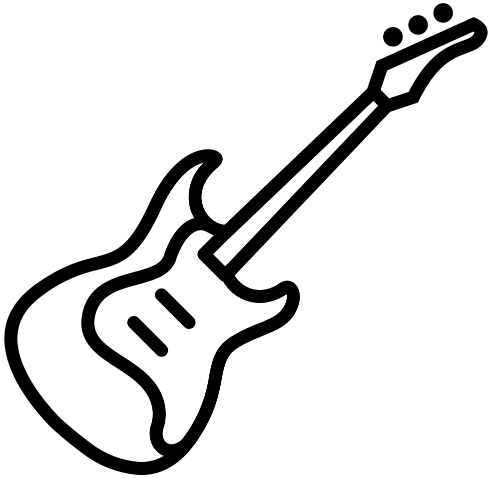
Optimización Aplicada a Sistemas de Potencia

1

Oscar Carreño
mauricio.carreno@udea.edu.co

Ingeniería Eléctrica
Universidad de Antioquia
2020

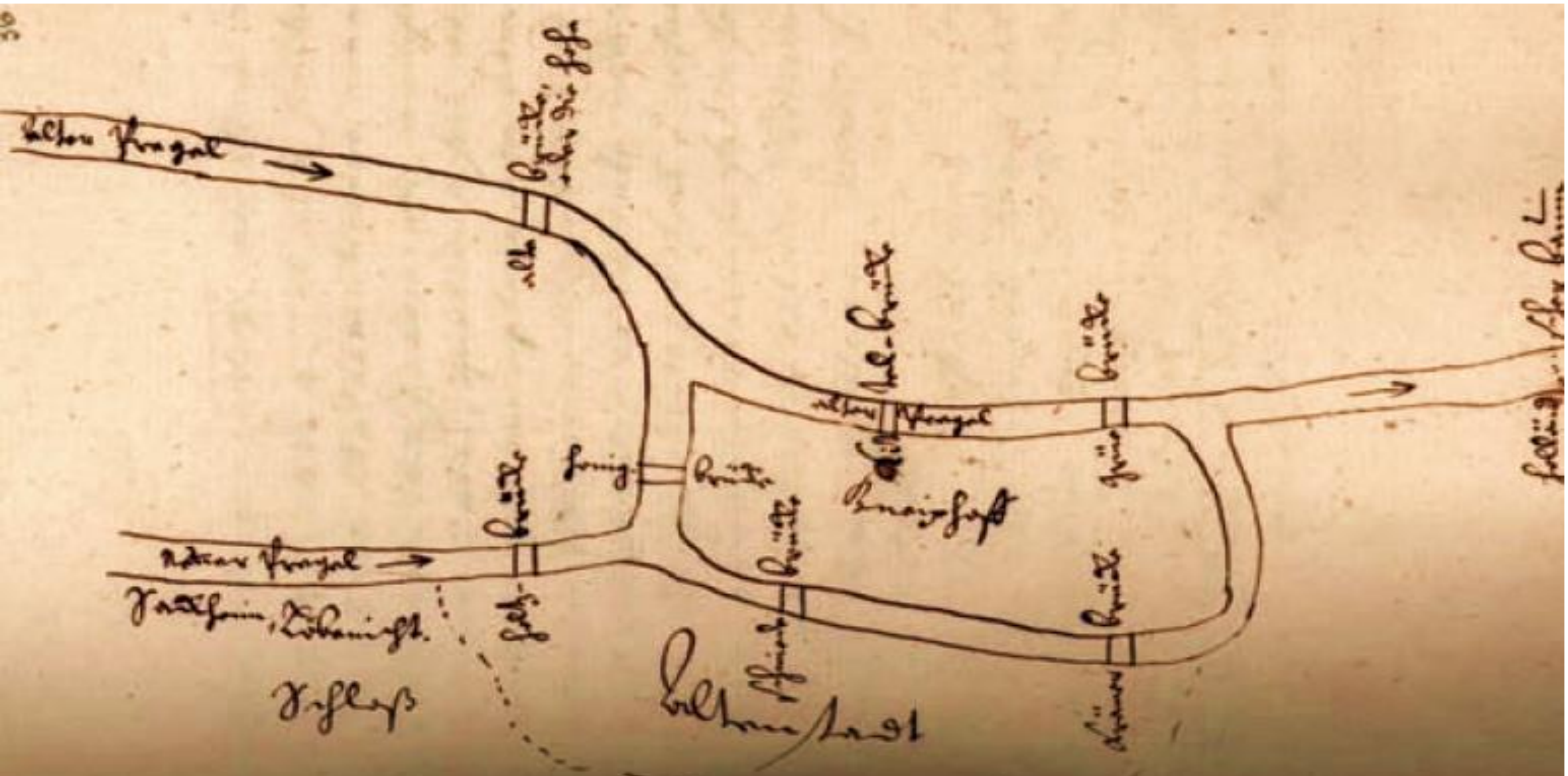


El profe

- Hijo de Emma y Ray (R.I.P)
- Ingeniero Electricista UdeA (1998)
- Músico de corazón (Punk Rocker).
- Esposo de Carolina
- Padre de Pocho (filósofo-los primos) y Juanita (doctora)
- Me encantan los deportes
- Emprendedor – Rightside

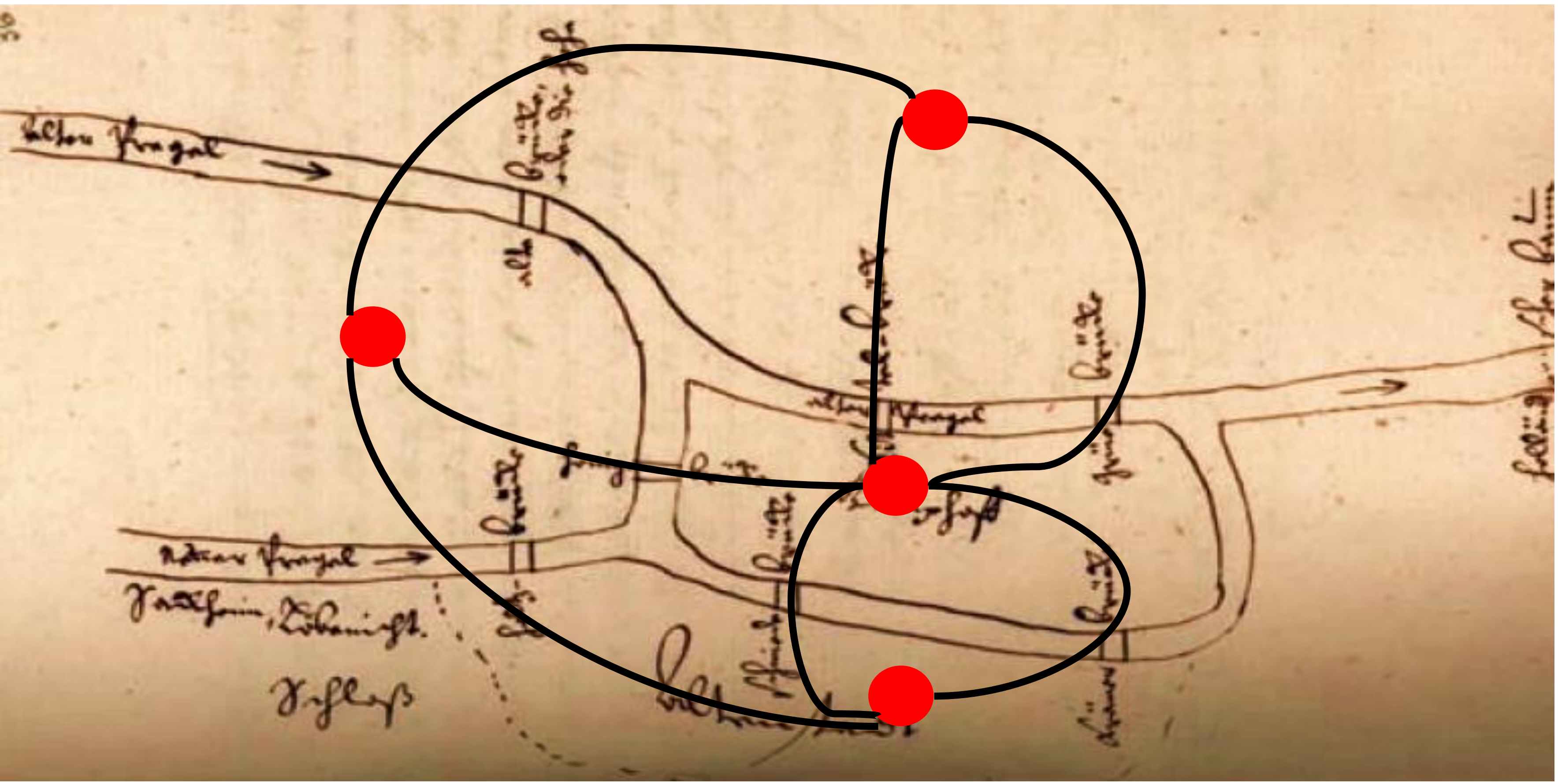
Leonhard Euler: 1736

The Königsberg Bridges Problem (*briefly* KBP):
Is it possible for a pedestrian to walk across all seven bridges in Königsberg without crossing any bridge twice?

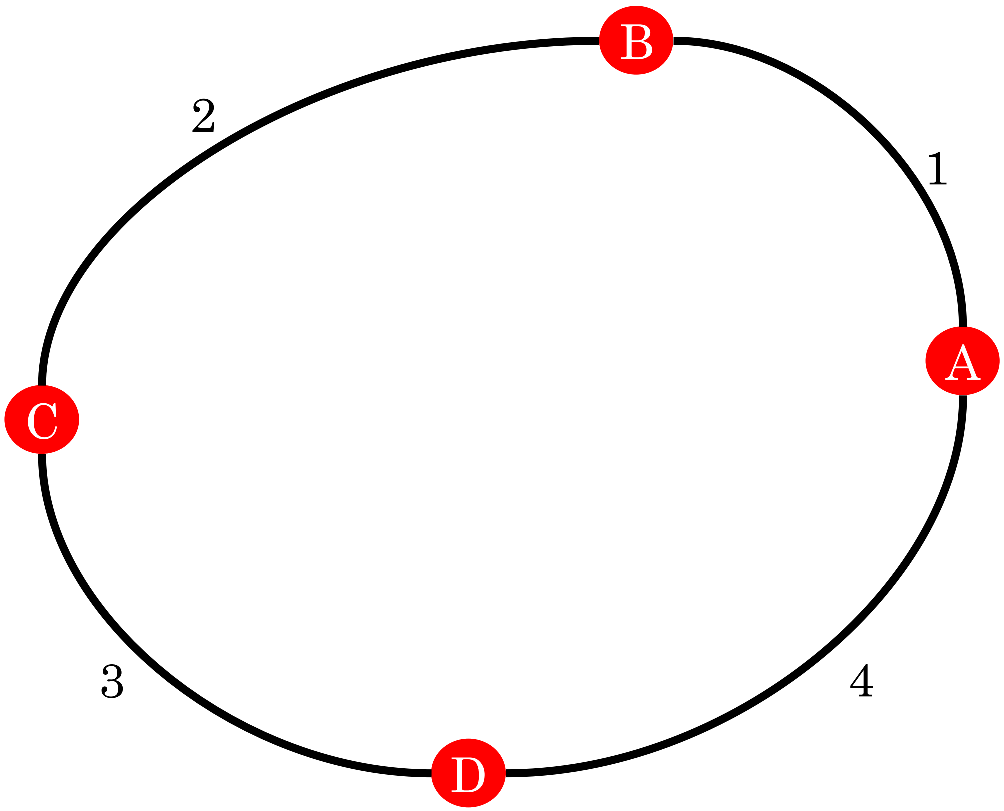


Leonhard Euler: 1736

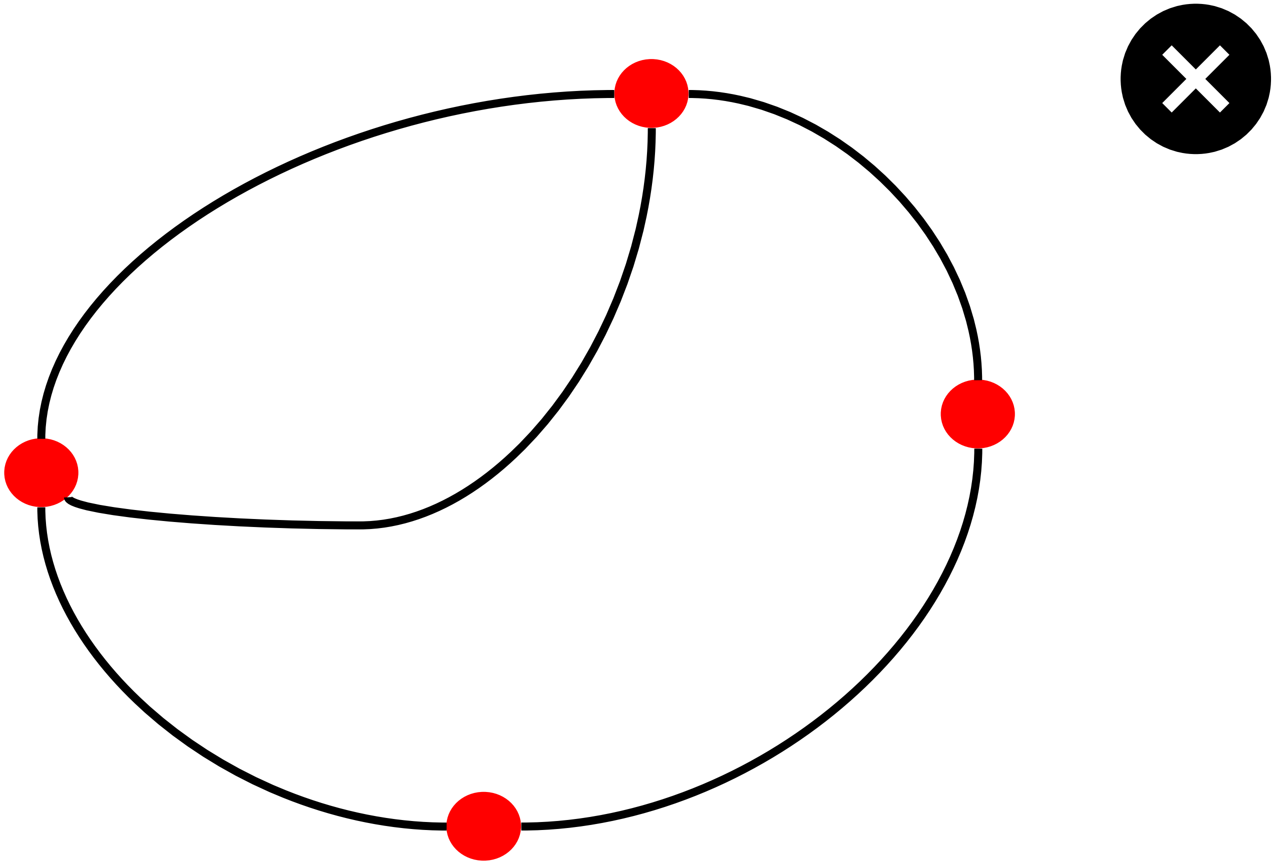
The Königsberg Bridges Problem (*briefly* KBP):
Is it possible for a pedestrian to walk across all seven bridges in Königsberg without crossing any bridge twice?



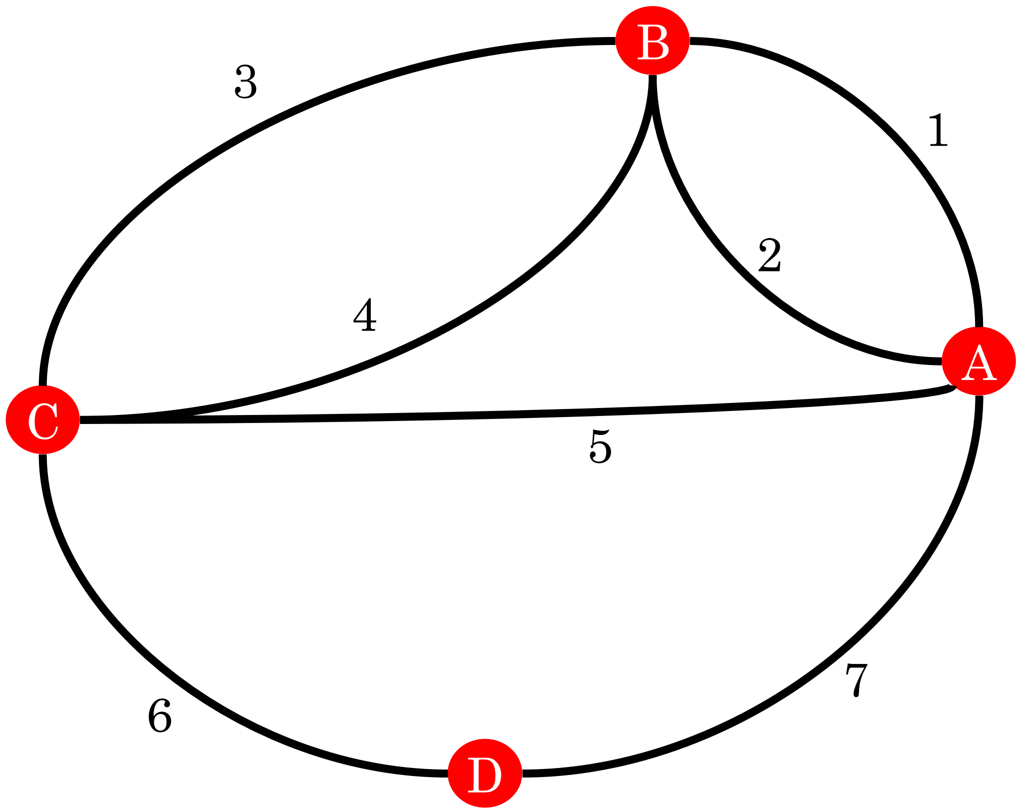
A1-B2-C3-D4



El Problema solo tiene solución si cada nodo tiene un número par de arcos asociado.



A1-B3-C6-D7-A5-C4-B2



Propósito

- Abordar de forma teórica y práctica, los problemas asociados al sector eléctrico.
- Motivar a los estudiantes en el conocimiento de la investigación de operaciones.

Justificación

El sector eléctrico en Colombia se encuentra en camino hacia la **transición energética** que incluye entre otras, el uso de **generación limpia** y la **participación de la demanda**. Esto genera nuevos retos para todos los actores del sector, y hace necesario el uso del estado del arte en **algoritmos de programación matemática**, para procesar **gran cantidad de información** y usarla para **tomar decisiones** en todos los horizontes de planeación.

Objetivo

- Guiar a los estudiantes en las bases teóricas del funcionamiento del mercado eléctrico colombiano.
- Guiar a los estudiantes en el aprendizaje del lenguaje Python y la librería de optimización Pyomo.
- Simular procesos reales relacionados con la transición energética

Programa

- Introducción al mercado eléctrico colombiano (4 h) - Teórica
- Introducción a Python-Pyomo (4h) - Práctica
- Características del Sistema Interconectado Nacional -SIN (4h) – Teórica
- Teoría de optimización lineal. (8h) – Teórica
- Teoría de subastas. (8h) – Teórica
- Programación de subastas de sobre cerrado. (4h) - Práctica
- Programación de subastas de dos puntas. (4h) – Práctica
- Modelos de planeación energética. (8h) – Teórica
- Programación del despacho económico. (4h) – Práctica
- Programación de un despacho hidrotérmico. (4h) -Práctica
- Retos de la Transición energética en Colombia. (4h) – Teórica
- Programación de modelos intradiarios. (8h) - Práctica

Evaluación

1. Parcial 20 % - **22 octubre**
2. Final 25 % - **Enero 2021**
3. Talleres prácticos python 25 % (individual) - Durante todo el curso (4-6)
4. Artículo técnico 25 % (grupos de 4) – **15 y 17 diciembre**
   - Formato IEEE – Resumen, palabras claves, introducción, modelo matemático, pruebas y resultados, conclusiones, bibliografía (al menos 2 referencias).
   - Resumen en Inglés.
   - Presentación (15 minutos).
5. 5% - Video individual de 2 minutos en Youtube. Tutorial de Pyomo (individual)

Bibliografía

1. BAZARAA M., JARVIS J. Programación lineal y flujo en redes
2. L. A. WOLSEY . Integer Programming.
3. Revista IEEE Transactions on Power Systems https://ieeexplore.ieee.org/
4. A. J. Wood and B. F. Wollenberg. Power Generation, Operation and Control.
5. Páginas CREG, UPME, XM.
6. http://www.pyomo.org/
7. Notas de clase. https://github.com/rightsidesas/claseUDEA

Requisito durante virtualidad

Para el curso el estudiante debe disponer de un computador, preferiblemente **Windows** y deberá instalar el software **Python** con la librería de optimización **Pyomo** y el software de optimización **CBC**. Además, el ambiente de desarrollo **Visual Studio Code**. Todas estas herramientas de software están disponibles de forma gratuita en internet.

IEEE TRANSACTIONS ON

POWER SYSTEMS

IEEE POWER & ENERGY SOCIETY

SEPTEMBER 2020 VOLUME 35 NUMBER 5 ITPSEG (ISSN 0885-8950)

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9170931

Buscar en el índice de la revista la palabra opt

Optimización Aplicada a Sistemas de Potencia

Universidad de Antioquia–2020

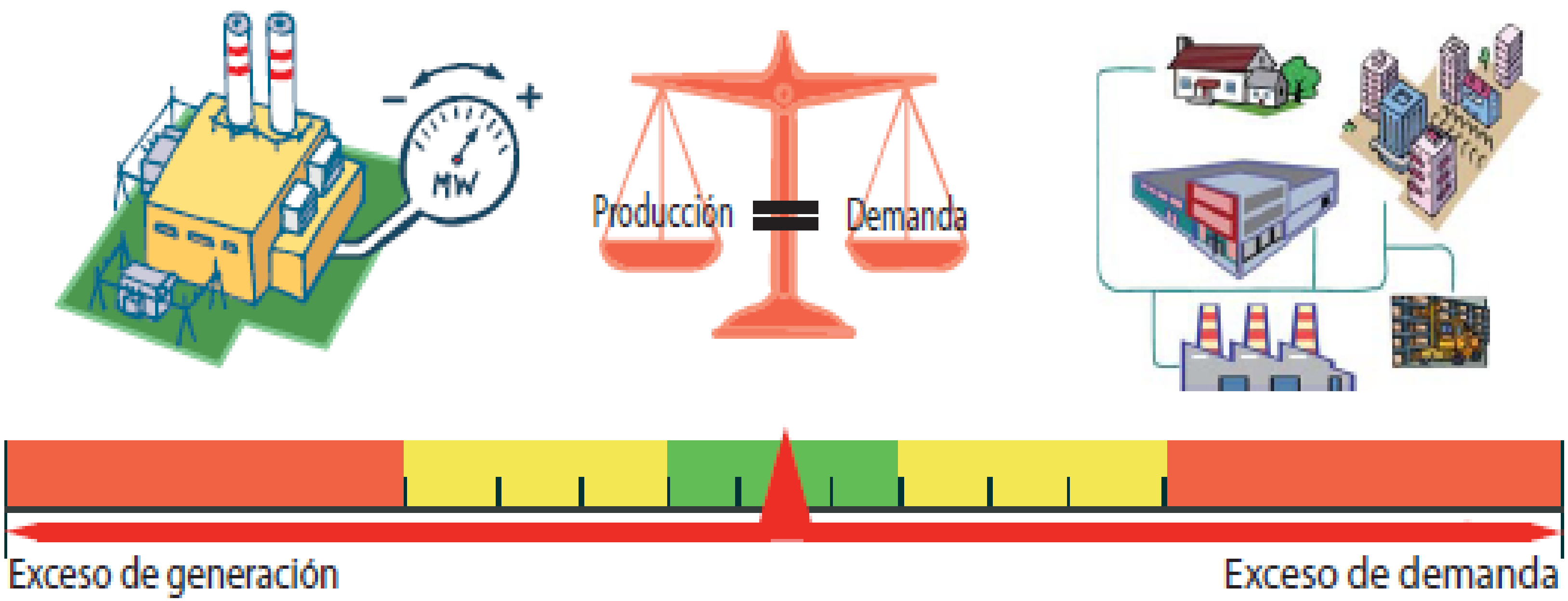
2

Introducción Al Mercado Eléctrico Colombiano

Cadena Eléctrica

Fuente www.xm.com.co

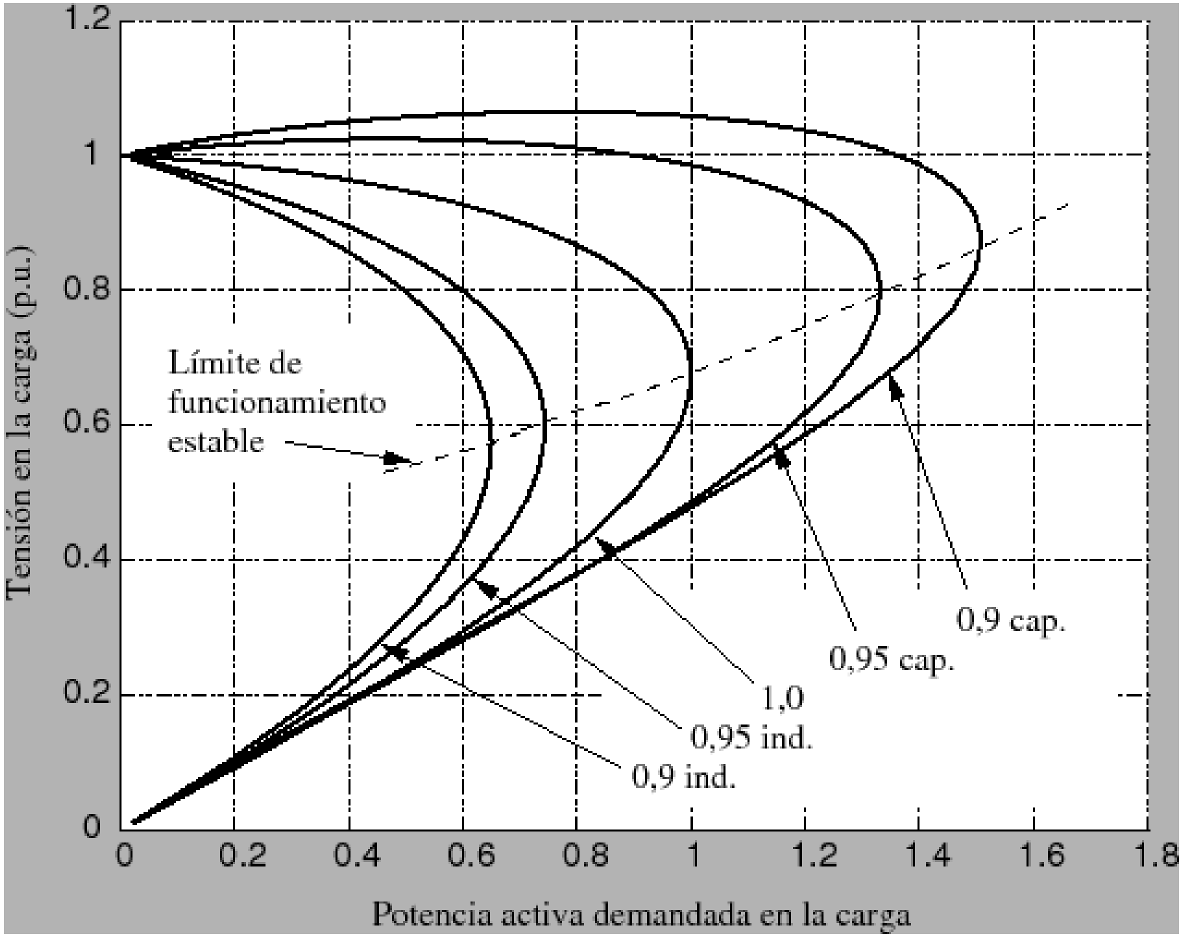
Frecuencia



59.8 - **60 Hz** – 60.2

Fuente www.ree.es

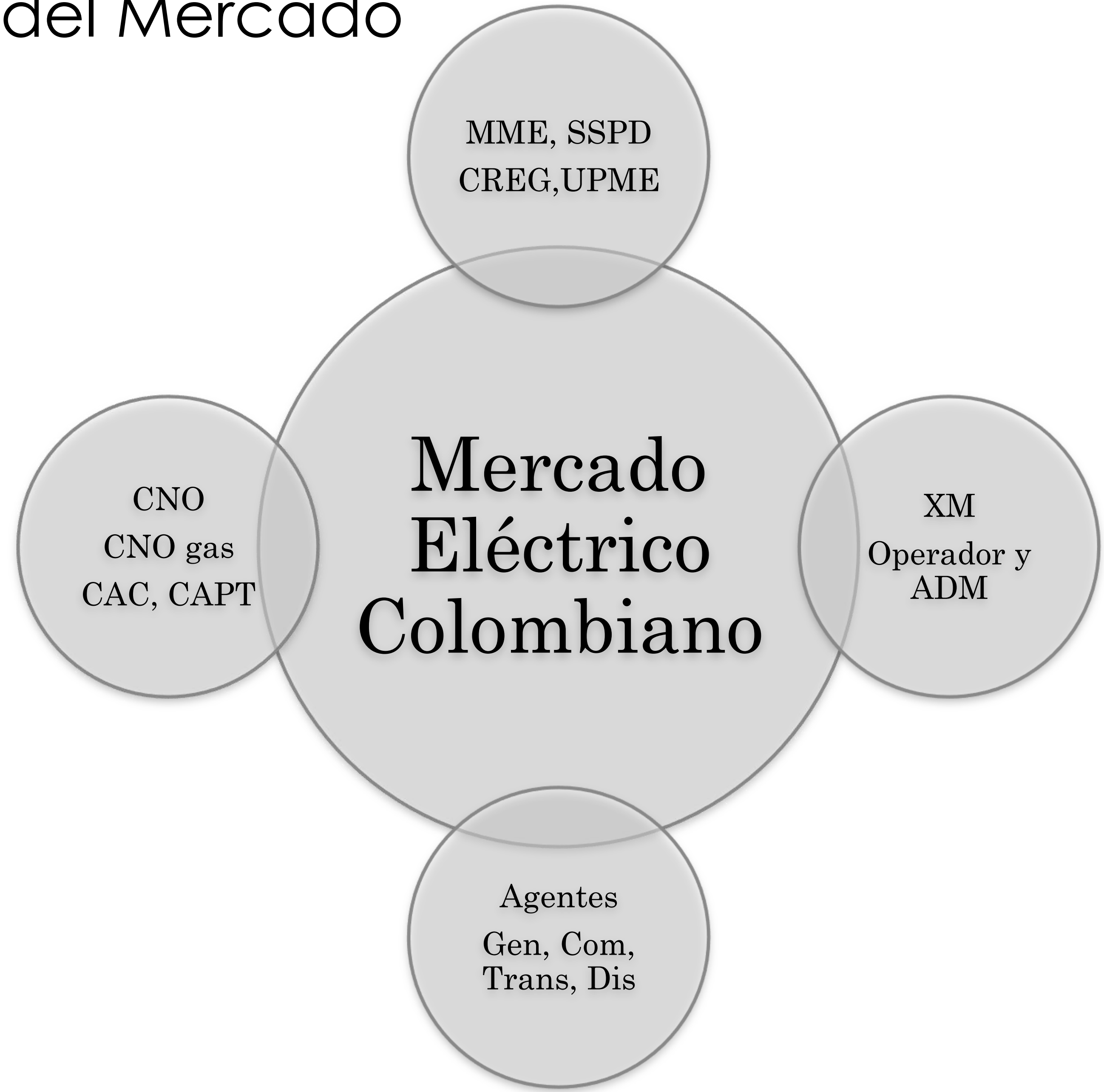
Voltaje



0.9 – **1 p.u** – 1.1 (1.05)

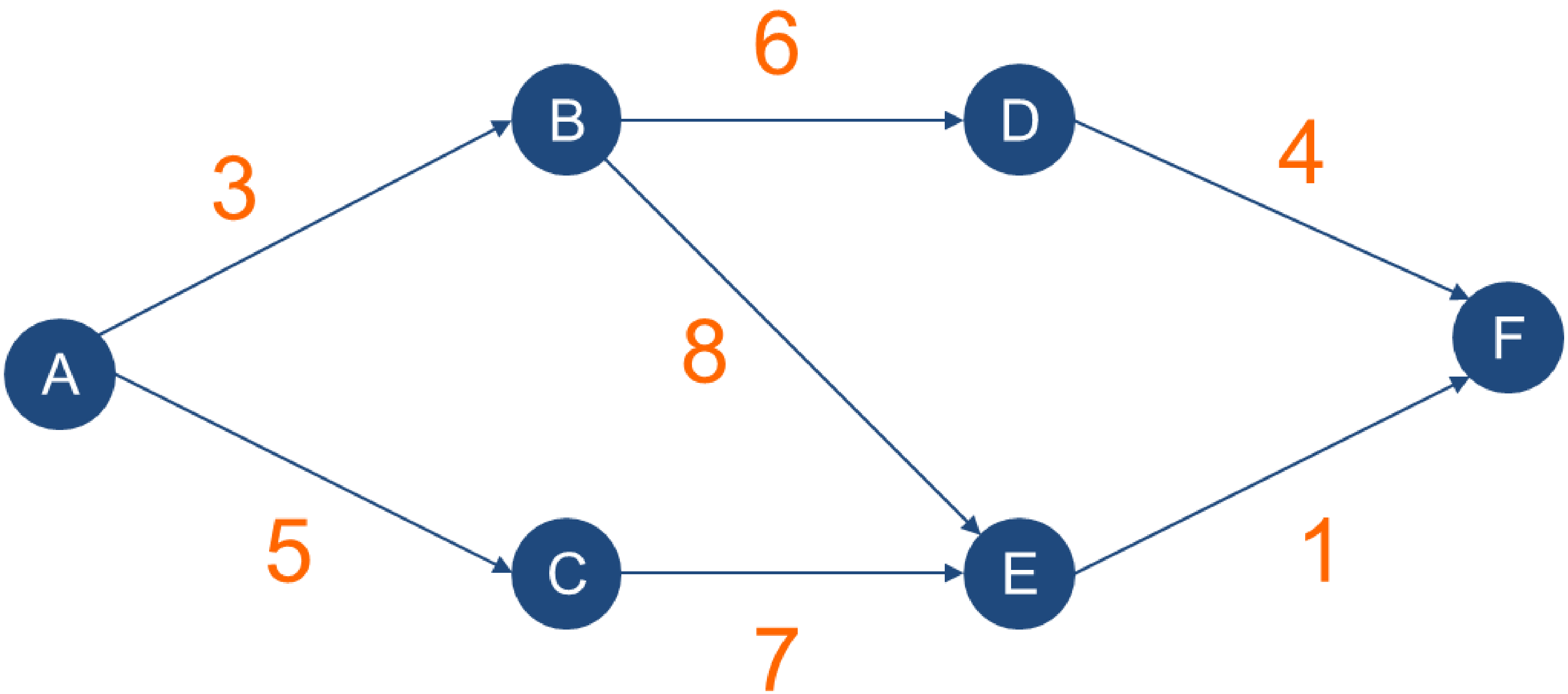
Fuente P. Kundur Power system stability and control

Actores del Mercado



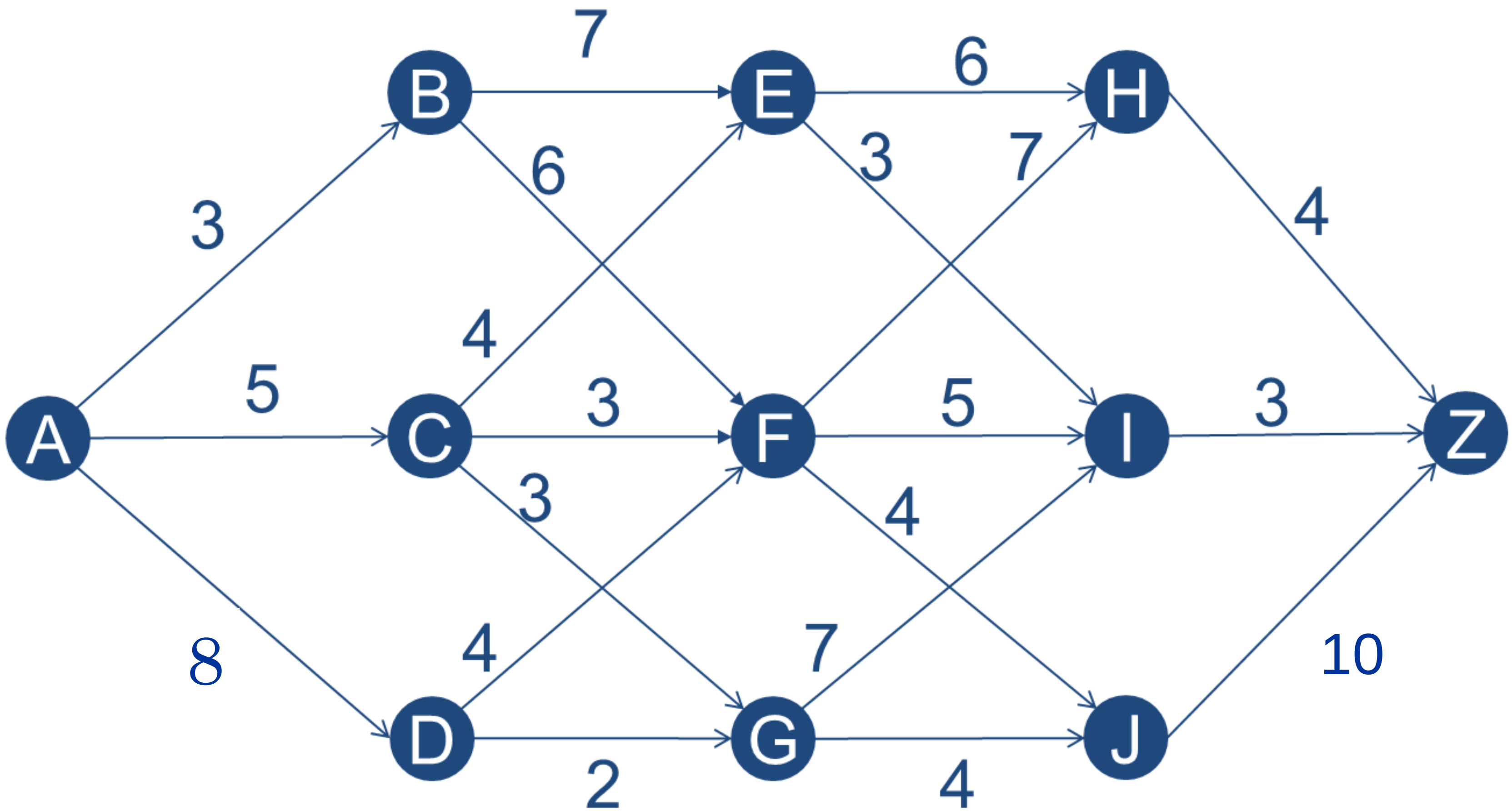
Introducción a la Investigación de Operaciones

Mejor Camino Entre A-F



Los números en los arcos representan costos o distancia

Mejor Camino Entre A-Z



Toma de Decisiones

- Una "Buena" Decisión La Puede Tomar Cualquiera, Bien Sea Por El Azar O Por Intuición.
- La Finalidad De Los Métodos Para Toma De Decisiones Es Servir De Soporte. (Información)
- Los Métodos Pueden Ser Cualitativos O Cuantitativos. Exactos O Aproximados, Determinísticos O Estocásticos, Lineales O No-lineales.
- El Desarrollo De Los Computadores Impondrán La "Fuerza Bruta" Sobre Cualquier Método Sofisticado. ¿Cuándo?

Historia

- George B. Dantzig 1947
- Programa De Despliegue Y Logístico Para La Fuerza Aérea De EU
- Programación En Una Estructura Lineal → Programación Lineal 1948
- 1949 Método Simplex
- ¿Pero Si El Mundo Es No Lineal? Richard Bellman – Programación Dinámica 1953

Problemas Típicos

- Problema De La Dieta
- Problema Del Agente Viajero
- Problema De Ruta Mas Corta
- Problema De Knapsack
- Problema De Patrones De Corte
- Problema De Planeación De La Producción
- Despacho Económico
- Despacho Hidrotérmico
- Flujo De Potencia Óptimo

Programación Dinámica

- Principio De Bellman: La Decisión Óptima Inmediata, Solo Depende Del Estado Actual, No De Cómo Se Llegó Hasta Él.
- Surge Por La Necesidad De Disponer De Un Algoritmo Más Sencillo Que El Simplex. (Computadores De La Época)
- Permite Solucionar Problemas No Lineales.
- Es Un Algoritmo Tipo Ascendente. Usa La Solución De Problemas Pequeños (Tablas) Para Encontrar La Solución Del Problema Completo.
- El Problema Debe Tener Una Estructura Especial En Donde Se Tomen Decisiones En Etapas Sucesivas.
- No Tiene Una Forma Estándar. Cada Problema Es Abordado De Forma Diferente.

Heurísticas

- Surgen Ante La Imposibilidad De Solucionar Problemas Complejos Con Métodos Exactos
- Son Algoritmos Basados En El Conocimiento Del Problema Y En Reglas Que Obedecen Mas A La Observación Y Experimentación Que A La Demostración Matemática.
- No Aseguran Un Óptimo. Velocidad Vs Precisión.
- Son De Baja Complejidad.
- Criterios De Parada.

Heurísticas

- Tabú Search
- Algoritmos Genéticos
- Enfriamiento Simulado
- Colonia De Hormigas
- Scatter Search
- Inteligencia Artificial

3

Programación Lineal

Representación estándar

$$\min | \max \to c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$$

$$\begin{aligned}
a_{11} x_1 + a_{12} x_2 + \ldots + a_{1n} x_n &= b_1 \\
a_{21} x_1 + a_{22} x_2 + \ldots + a_{2n} x_n &= b_2 \\
&\vdots \\
a_{m1} x_1 + a_{m2} x_2 + \ldots + a_{mn} x_n &= b_m
\end{aligned}$$

$$x_1, x_2, \ldots x_n \geq 0$$

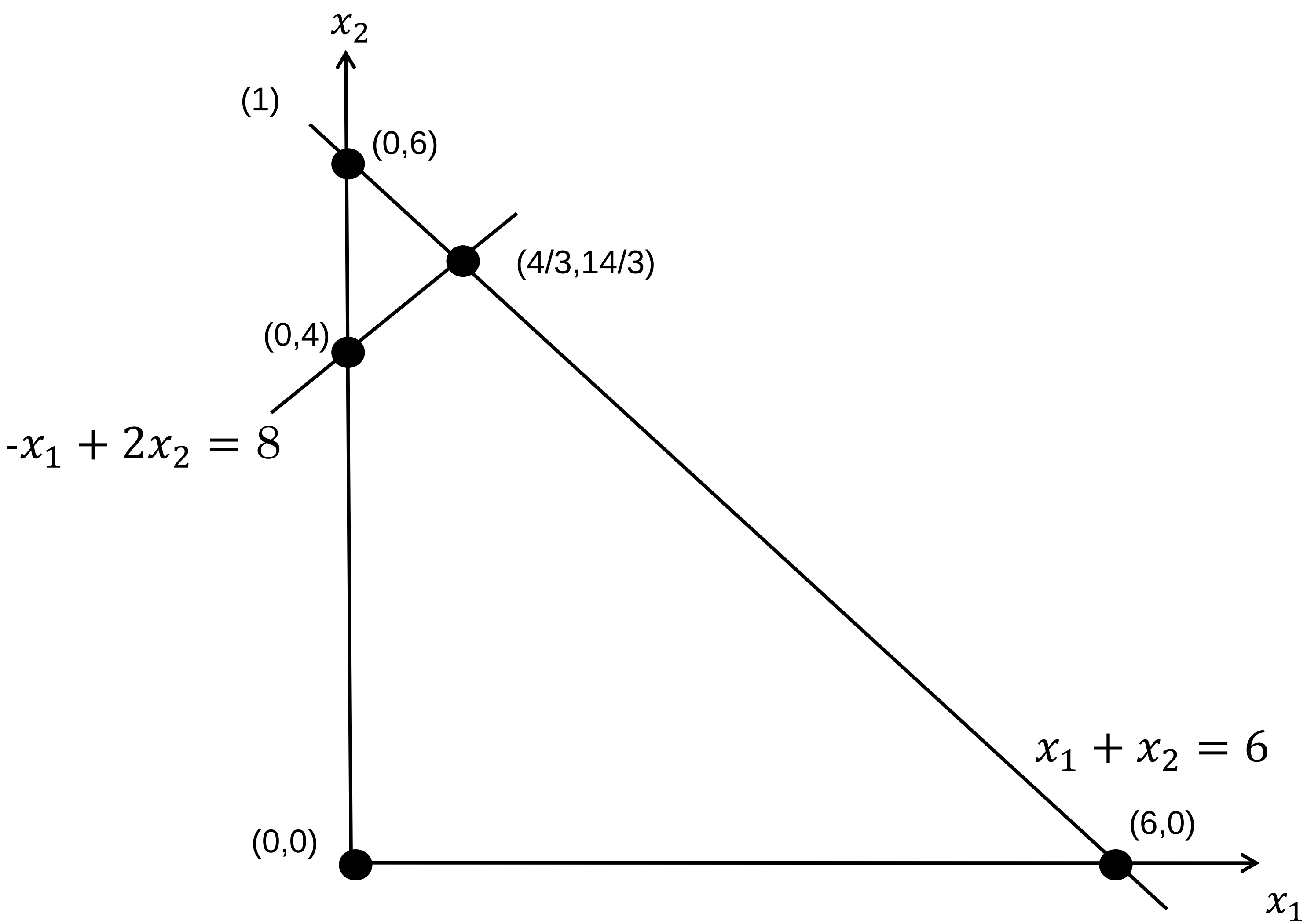
Representación estándar

$$x_1 + x_2 = 6$$

$$-x_1 + 2x_2 = 8$$

- Encontrar la solución del sistema de ecuaciones
- Graficar las ecuaciones y la solución

Solución geométrica



Sistema de inecuaciones

$$x_1 + x_2 \leq 6$$

$$-x_1 + 2x_2 \leq 8$$

- Encontrar la solución del sistema de ecuaciones
- Graficar las ecuaciones y la solución

Sistema de inecuaciones

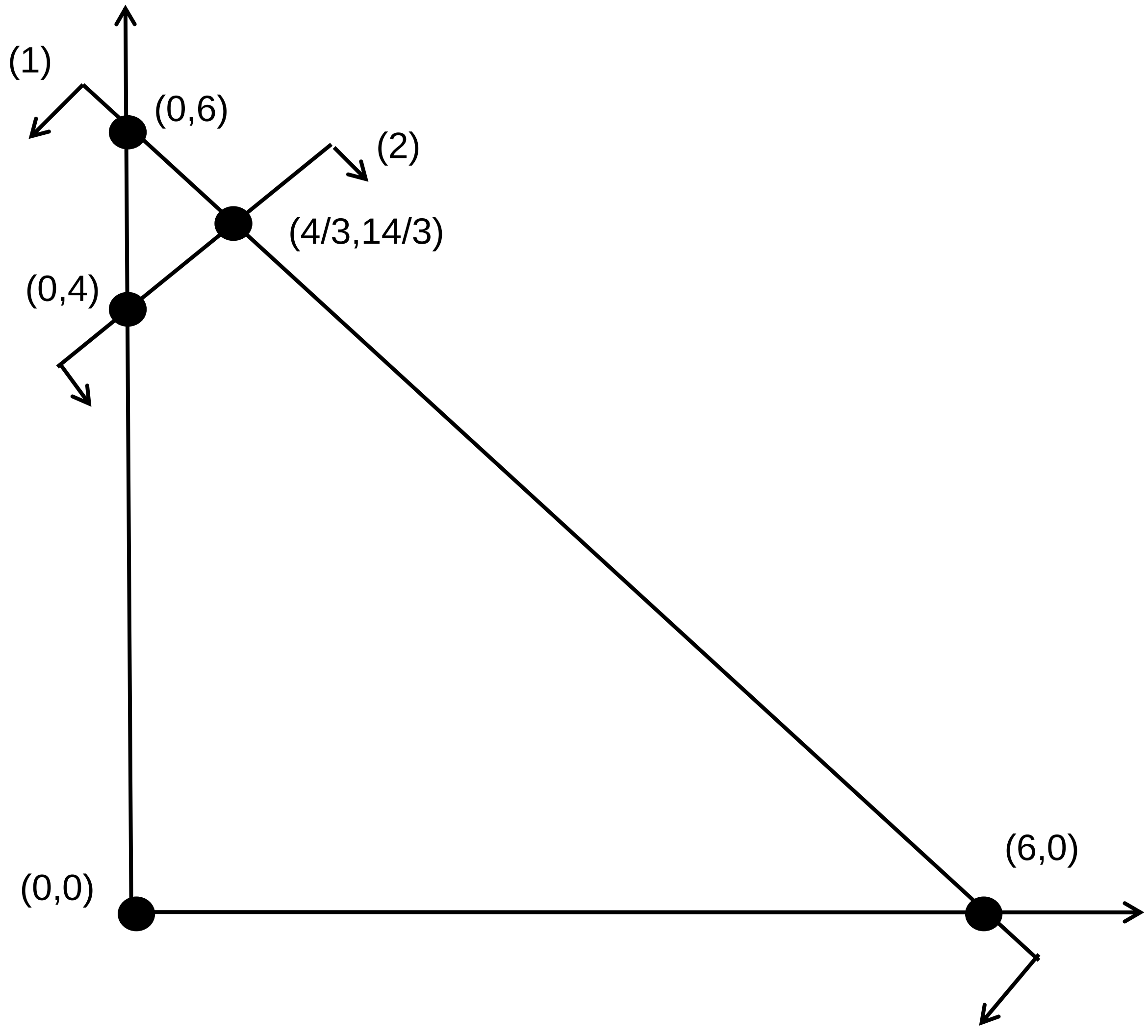
$$x_1 + x_2 \leq 6$$

$$-x_1 + 2x_2 \leq 8$$

$$x_1 \geq 0,\ x_2 \geq 0$$

- Encontrar la solución del sistema de ecuaciones
- Graficar las ecuaciones y la solución

Solución geométrica



Problema de programación lineal

$$min - x_1 - 3x_2$$

Función objetivo

$$s.a$$

$$x_1 + x_2 \leq 6$$

Restricciones

$$-x_1 + 2x_2 \leq 8$$

$$x_1 \geq 0, x_2 \geq 0$$

- Graficar las ecuaciones y la solución

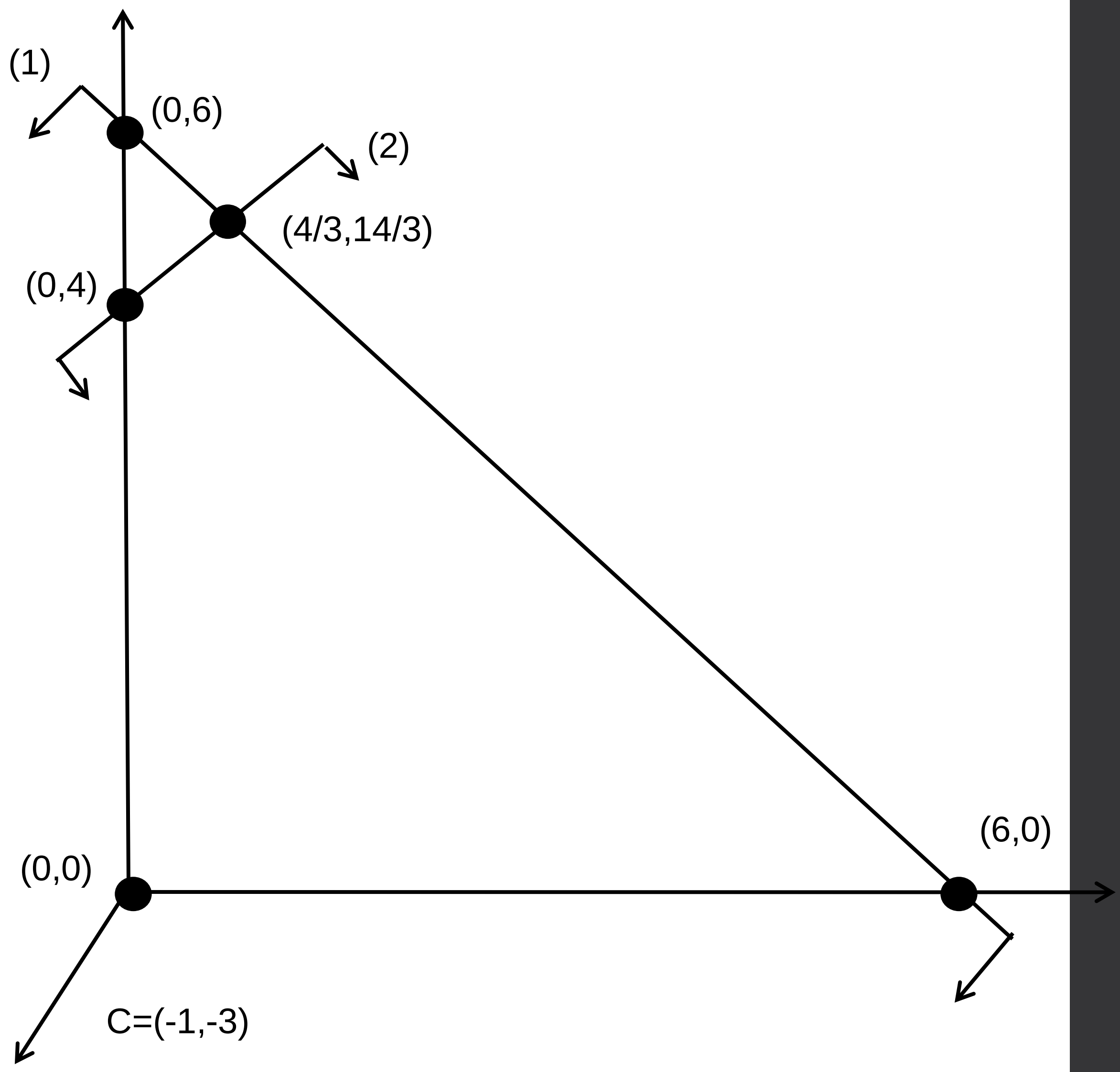
Solución geométrica

$$min - x_1 - 3x_2$$

$$x_1 + x_2 \leq 6$$

$$-x_1 + 2x_2 \leq 8$$

$$x_1 \geq 0, x_2 \geq 0$$



Solución geométrica

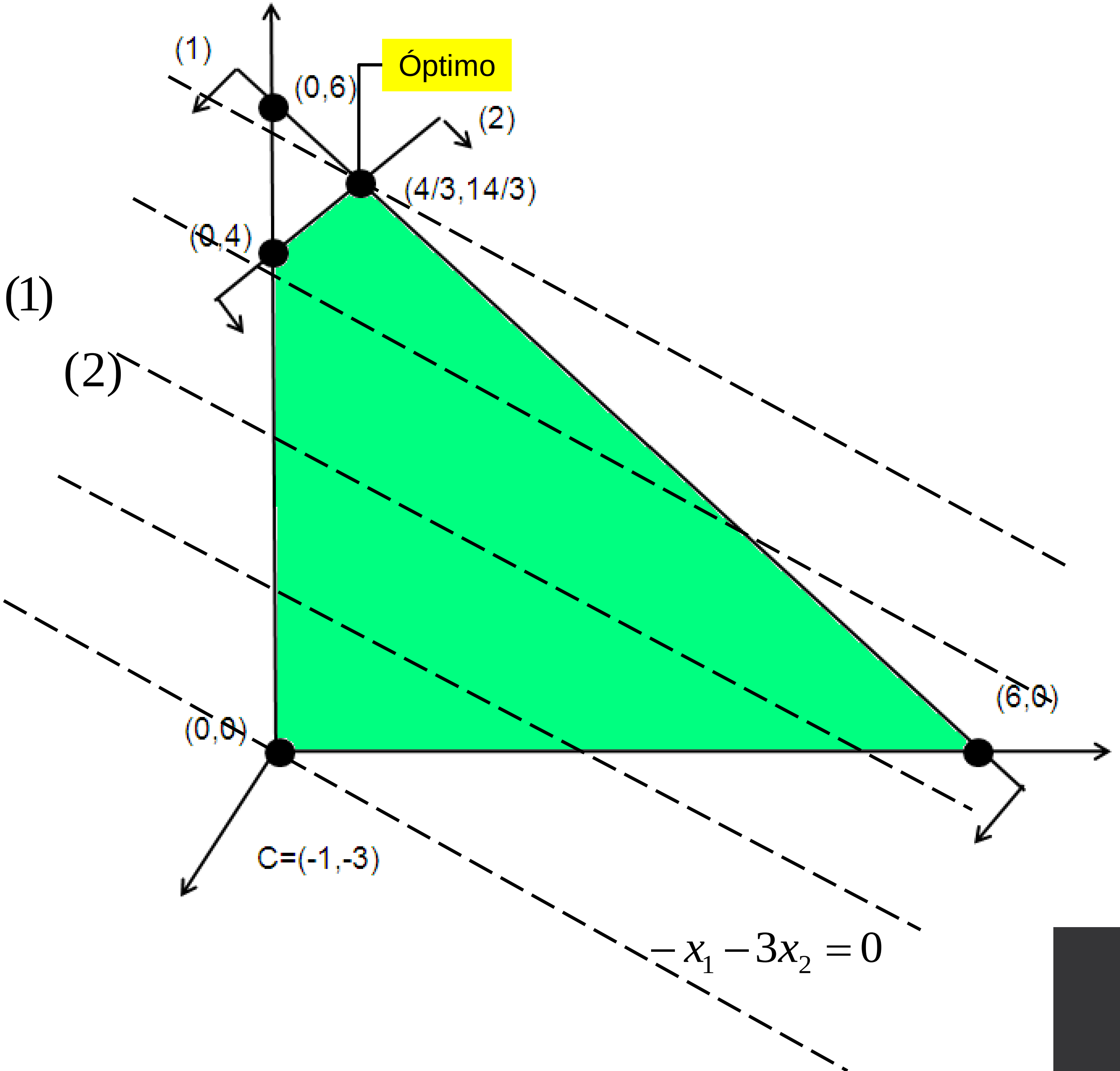
$$\min \rightarrow -x_1 - 3x_2$$

$s.a$

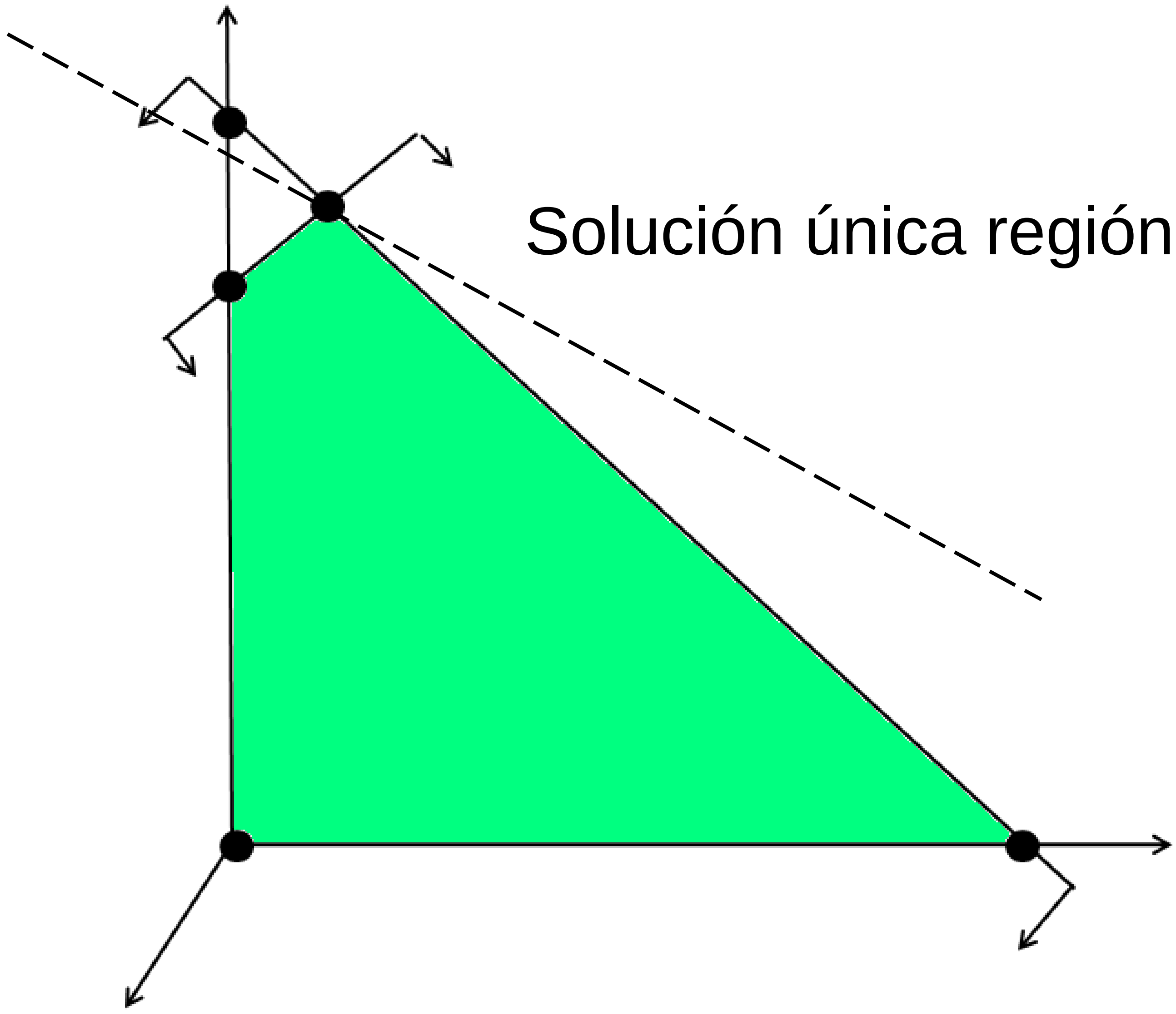
$$x_1 + x_2 \leq 6 \qquad (1)$$

$$-x_1 + 2x_2 \leq 8 \qquad (2)$$

$$x_1, x_2 \geq 0$$



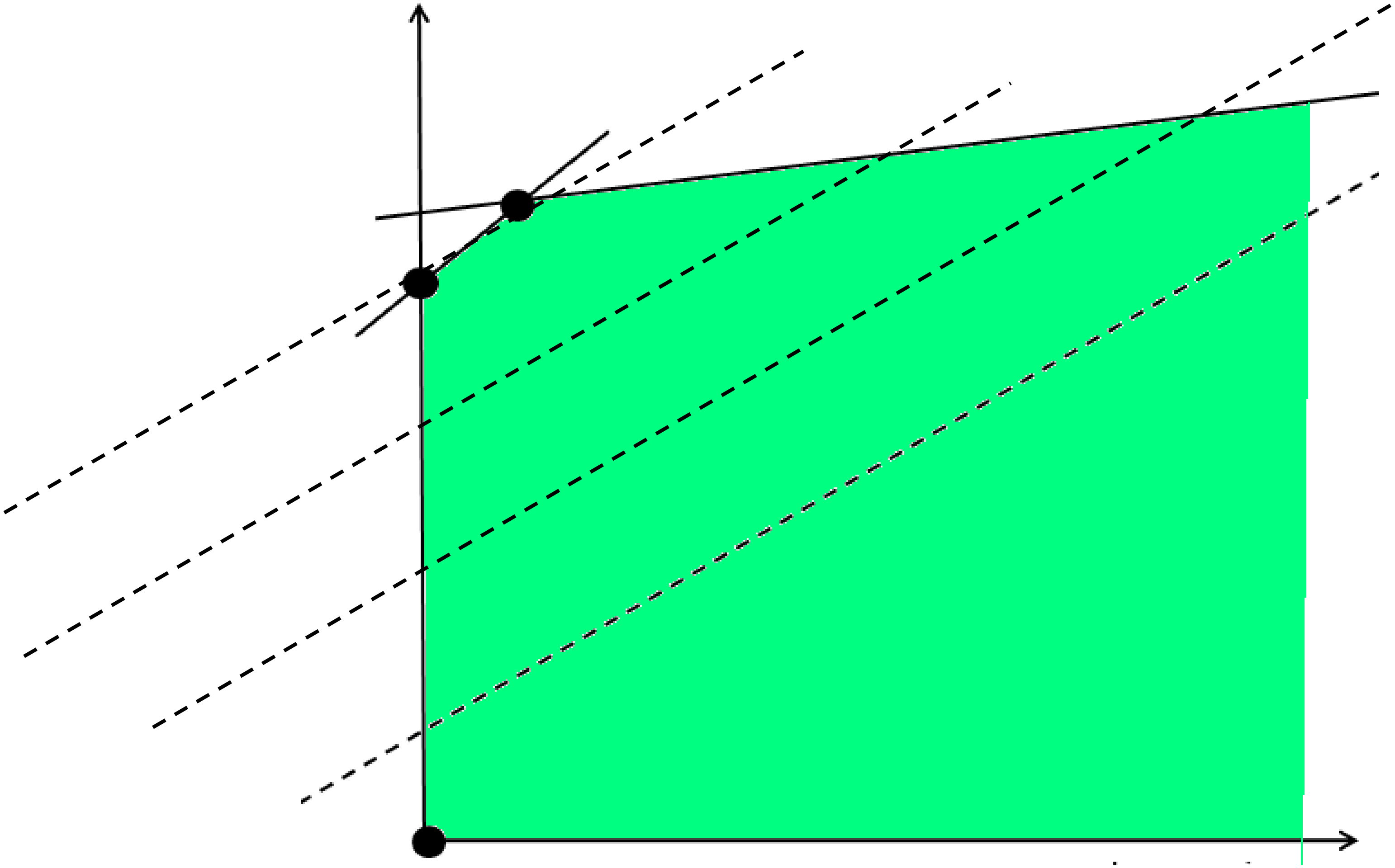
Tipos de soluciones



Solución única región acotada

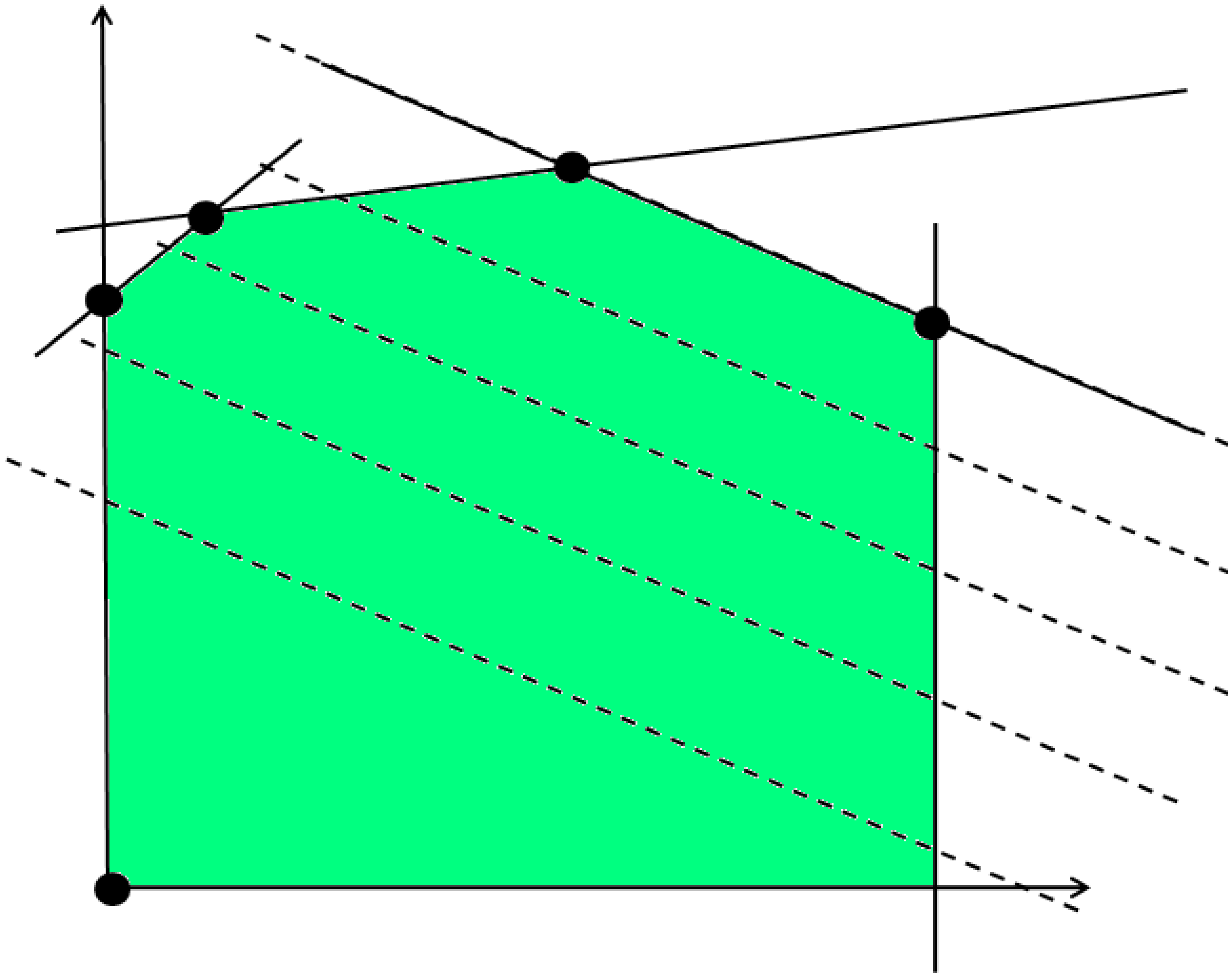
Tipos de soluciones

Solución única región no acotada



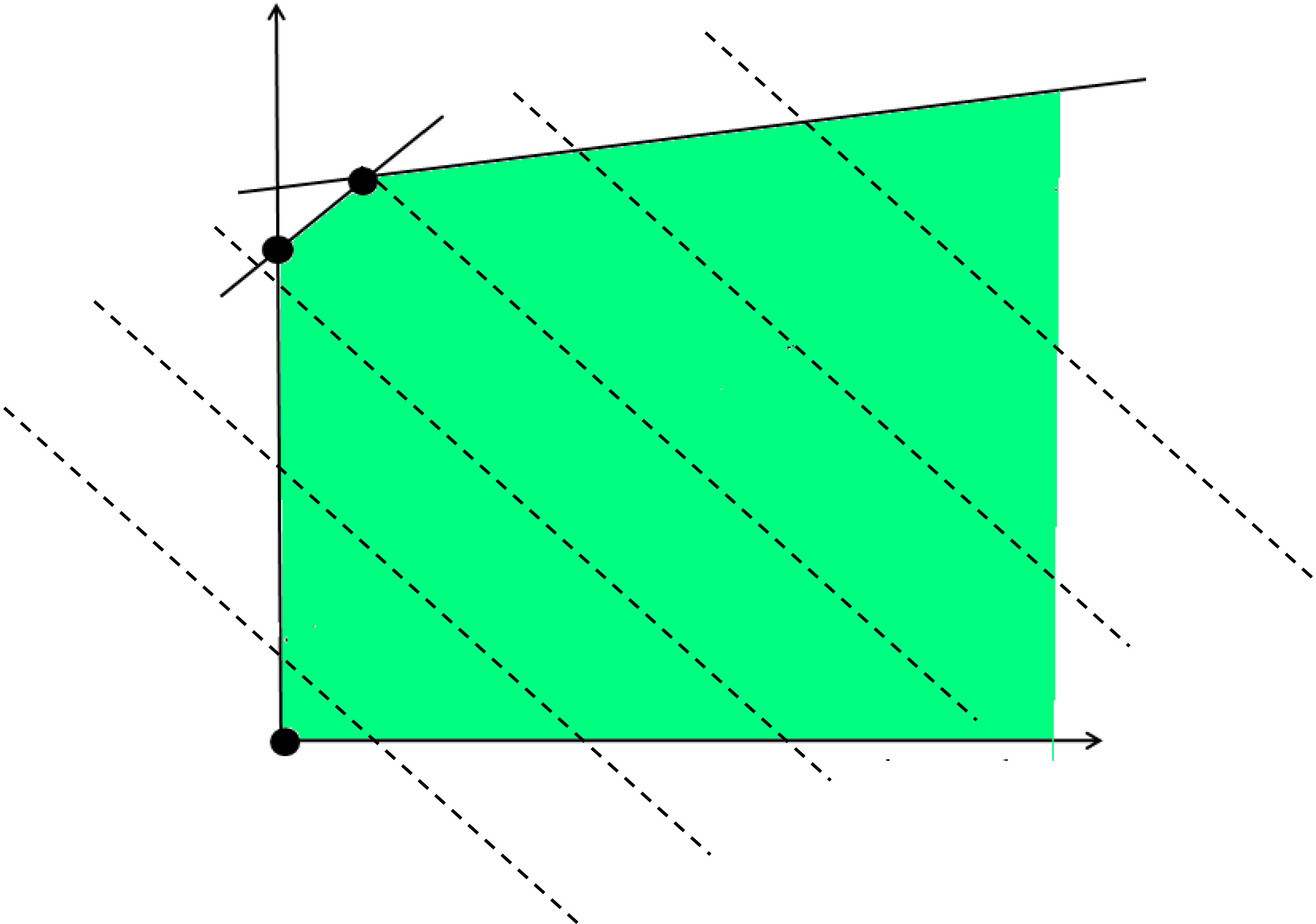
Tipos de soluciones

Infinitas soluciones



Tipos de soluciones

Solución no acotada



Convertir a forma estándar

$$x_1 + x_2 \leq 6 \Rightarrow x_1 + x_2 + h = 6$$

$$x_1 + x_2 \geq 8 \Rightarrow x_1 + x_2 - e = 8$$

$$x_1 \geq 0, x_2 \geq 0, h \geq 0, e \geq 0$$

Convertir a forma estándar

$$\max\ (x) \Rightarrow \min\ (-x)$$

- ¿Cómo convertir variables no acotadas en variables positivas?

Problema de la mochila

variable	Peso	Valor
x1	5	8
x2	7	3
x3	4	6
x4	3	11

$$pesoMax = 14$$

$$\max \quad 8x_1 + 3x_2 + 6x_3 + 11x_4$$

$$s.a$$

$$5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$

$$x_1 \geq 0, x_2 \geq 0$$

- ¿Qué diferencia hay entre este problema y el resuelto en el taller cero?

Forma matricial

$$\begin{aligned} &\min \rightarrow cx \\ &s.a \\ &Ax = b \\ &x \geq 0 \end{aligned} \quad \Rightarrow \quad \begin{aligned} &\min \rightarrow cx \\ &s.a \\ &Bx_B + Nx_N = b \\ &x_B, x_N \geq 0 \end{aligned}$$

Forma matricial

$$\max \sum_{i=1}^{4} VALOR_i \cdot x_i$$

$$\sum_{i=1}^{4} PESO_i \cdot x_i \leq 14$$

$$x_i \geq 0 \quad \forall i$$

Problema de la mochila

Forma algebraica

$$\min/\max \sum_{i=1}^{n} \sum_{j=1}^{m} C_{i,j} \cdot x_{i,j}$$

$$\sum_{i=1}^{n} A_{i,j} \cdot x_{i,j} \leq B_j \qquad \forall_j$$

$$x_{i,j} \geq 0 \quad \forall i, j$$

Forma algebraica

$$\max \sum_{i=1}^{4} VALOR_i \cdot x_i$$

$$\sum_{i=1}^{4} PESO_i \cdot x_i \leq 14$$

$$x_i \geq 0 \quad \forall i$$

Problema de la mochila

Pasos iteración simplex

1. Convertir el problema a la forma estándar
2. Elegir una solución que cumpla las restricciones (factible)
3. Validar si es la solución óptima. (Costo reducido)
4. Elegir que variable entra y que variable sale
5. Encontrar el valor de las variables de decisión y validar si es factible

Ejemplo simplex

Introduction
to Linear Optimization

Dimitris Bertsimas
John N. Tsitsiklis

Example 3.1 Consider the linear programming problem

$$\begin{array}{lrcrcrcrcl} \text{minimize} & c_1x_1 & + & c_2x_2 & + & c_3x_3 & + & c_4x_4 & & \\ \text{subject to} & x_1 & + & x_2 & + & x_3 & + & x_4 & = & 2 \\ & 2x_1 & & & + & 3x_3 & + & 4x_4 & = & 2 \\ & \multicolumn{9}{l}{x_1, x_2, x_3, x_4 \geq 0.} \end{array}$$

The first two columns of the matrix $\mathbf{A}$ are $\mathbf{A}_1 = (1,2)$ and $\mathbf{A}_2 = (1,0)$. Since they are linearly independent, we can choose x_1 and x_2 as our basic variables. The corresponding basis matrix is

$$\mathbf{B} = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix}.$$

Ejemplo simplex

Introduction
to Linear Optimization

Dimitris Bertsimas
John N. Tsitsiklis

We set $x_3 = x_4 = 0$, and solve for x_1, x_2, to obtain $x_1 = 1$ and $x_2 = 1$.

reduced cost $\bar{c}_3$ of the nonbasic variable x_3 was found to be $-3c_1/2 + c_2/2 + c_3$.

$$\bar{c}_j = c_j - \mathbf{c}'_B \mathbf{B}^{-1} \mathbf{A}_j.$$

Suppose that $\mathbf{c} = (2, 0, 0, 0)$, in which case, we have $\bar{c}_3 = -3$.

Problema lineal Entera

$$\max \rightarrow x_1 + 0.64x_2$$

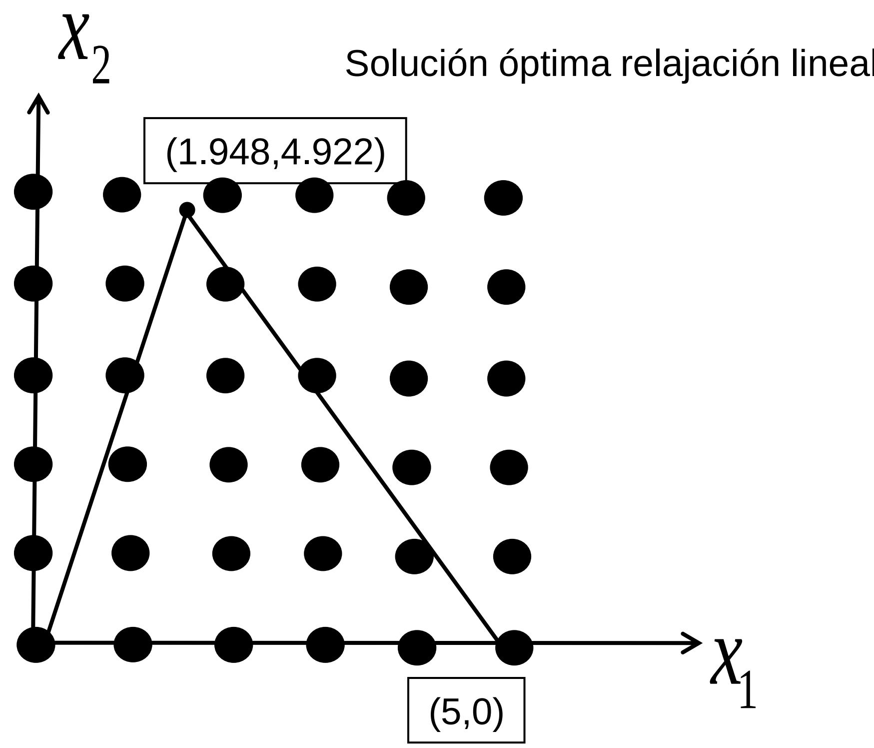
$$s.a$$

$$50x_1 + 31x_2 \leq 250$$

$$3x_1 - 2x_2 \geq -4$$

$$x_1, x_2 \geq 0$$

$$x_1, x_2 \in Z$$



Branch and Bound

$$\max \to 8x_1 + 11x_2 + 6x_3 + 4x_4$$

$$s.a$$

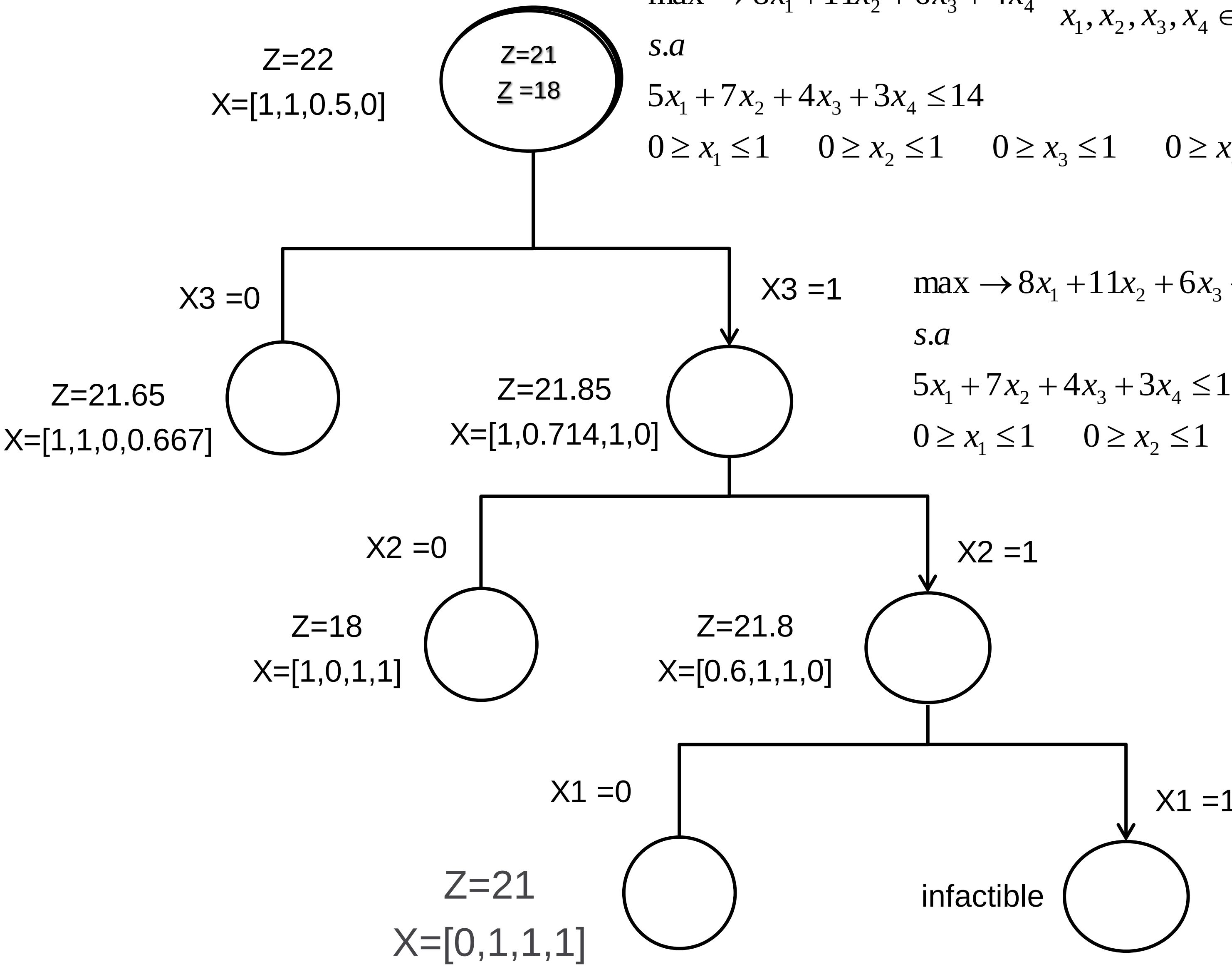
$$5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$

$$x_1, x_2, x_3, x_4 \in \{0,1\}$$

Branch and Bound

$$\max \rightarrow 8x_1 + 11x_2 + 6x_3 + 4x_4$$
$$s.a$$
$$5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$
$$x_1, x_2, x_3, x_4 \in \{0,1\}$$

$$\max \rightarrow 8x_1 + 11x_2 + 6x_3 + 4x_4$$
$$s.a$$
$$5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$
$$0 \geq x_1 \leq 1 \quad 0 \geq x_2 \leq 1 \quad 0 \geq x_3 \leq 1 \quad 0 \geq x_4 \leq 1$$



$$\max \rightarrow 8x_1 + 11x_2 + 6x_3 + 4x_4$$
$$s.a$$
$$5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$
$$0 \geq x_1 \leq 1 \quad 0 \geq x_2 \leq 1 \quad x_3 = 1 \quad 0 \geq x_4 \leq 1$$

Pasos para implementar una herramienta de optimización

1. Entender el Problema.
2. Identificar el objetivo. Definir si hay o no un problema de optimización.
3. Identificar variables de decisión.
4. Definir subíndices y conjuntos.
5. Definir Datos de entrada.
6. Definir función objetivo y restricciones.
7. Escribir el modelo en un software de programación algebraica. (Prototipo)
8. Solucionar el problema con un optimizador y validar resultados.
9. Implementar software de usuario final. (industrial)