Programación Lineal - Pt 1

Investigación Operativa

¿Qué es la Programación Lineal?

Definición:

- Técnica de optimización matemática
- Busca la mejor solución a problemas con función objetivo y restricciones lineales
- Herramienta fundamental para toma de decisiones
- Optimización de recursos limitados

Aplicaciones de la Programación Lineal

Aplicaciones:

- Producción y manufactura
- Logística y transporte
- Finanzas y inversiones

Problema de PL General - Componentes

Elementos del modelo:

- **Variables de decisión** (x_j): Nivel de cada actividad

Problema de PL General - Componentes

Elementos del modelo:

- **Variables de decisión** (x_j): Nivel de cada actividad
- **Función objetivo** (Z): Medida de performance a optimizar

Problema de PL General - Componentes

Elementos del modelo:

- **Variables de decisión** (x_j): Nivel de cada actividad
- **Función objetivo** (Z): Medida de performance a optimizar
- **Parámetros:**
  - c_j: Aumento de Z por unidad de x_j
  - b_i: Cantidad del recurso i disponible
  - a_{ij}: Cantidad del recurso i que consume cada unidad de actividad j

Forma estándar:

$$\textbf{Min } Z$$

Dado

$$Z = cx^t$$

$$ax^t \leq b$$

$$x \geq 0$$

Donde:

- $x = (x_1, x_2, \ldots, x_n)$ - vector de variables
- $c = (c_1, c_2, \ldots, c_n)$ - vector de coeficientes
- $b = (b_1, b_2, \ldots, b_m)$ - vector de recursos
- a - matriz de consumo de recursos por unidad de actividad

Ejemplo 1: Panes y Tortas

Situación: Jorge tiene una panadería que produce panes y tortas.

Recursos por unidad:

- **Pan:** 1 kg harina, 0.2 kg levadura
- **Torta:** 0.5 kg harina, 0.1 kg levadura, 0.2 kg azúcar

Disponibilidad diaria:

- 50 kg de harina
- 15 kg de levadura
- 10 kg de azúcar

Ganancias: Pan \$2.5, Torta \$1.8

Ejemplo 1: Formulación

Variables de decisión:

- x_1: Cantidad de panes a producir
- x_2: Cantidad de tortas a producir

Ejemplo 1: Formulación

Variables de decisión:

- x_1: Cantidad de panes a producir
- x_2: Cantidad de tortas a producir

Función objetivo:

$$\text{Maximizar: } Z = 2{,}5x_1 + 1{,}8x_2$$

Ejemplo 1: Formulación

Variables de decisión:

- x_1: Cantidad de panes a producir
- x_2: Cantidad de tortas a producir

Función objetivo:

$$\text{Maximizar: } Z = 2{,}5x_1 + 1{,}8x_2$$

Restricciones:

$$\begin{aligned} x_1 + 0{,}5x_2 &\leq 50 \quad \text{(harina)} \\ 0{,}2x_1 + 0{,}1x_2 &\leq 15 \quad \text{(levadura)} \\ 0{,}2x_2 &\leq 10 \quad \text{(azúcar)} \\ 20 \leq x_1 &\leq 50, \quad 15 \leq x_2 \leq 20 \end{aligned}$$

Ejemplo 2: Wyndor Glass Co.

Productos:

- **Producto 1:** Ventana 2m con marco de aluminio
- **Producto 2:** Ventana colgante 3m con marco de madera

Recursos de producción:

Planta	Producto 1	Producto 2	Tiempo disponible
1	1	0	4
2	0	2	12
3	3	2	18

Ganancias: Producto 1: \$3000, Producto 2: \$5000

Wyndor Glass Co. - Formulación

Variables de decisión:

- x_1: Lotes por semana del Producto 1
- x_2: Lotes por semana del Producto 2

Wyndor Glass Co. - Formulación

Variables de decisión:

- x_1: Lotes por semana del Producto 1
- x_2: Lotes por semana del Producto 2

Función objetivo:

$$\text{Maximizar: } Z = 3000x_1 + 5000x_2$$

Wyndor Glass Co. - Formulación

Variables de decisión:

- x_1: Lotes por semana del Producto 1
- x_2: Lotes por semana del Producto 2

Función objetivo:

$$\text{Maximizar: } Z = 3000x_1 + 5000x_2$$

Restricciones:

$$\begin{aligned} x_1 &\leq 4 \quad \text{(Planta 1)} \\ 2x_2 &\leq 12 \quad \text{(Planta 2)} \\ 3x_1 + 2x_2 &\leq 18 \quad \text{(Planta 3)} \\ x_1, x_2 &\geq 0 \end{aligned}$$

Método Simplex - Concepto

¿Que es el Metodo Simplex?

- Es un algoritmo iterativo para resolver problemas de programacion lineal
- Se basa en que la solucion optima esta en un vertice de la region factible
- Se mueve de vertice en vertice mejorando la funcion objetivo
- Se detiene cuando no hay vertices adyacentes mejores

Método Simplex - Pasos del algoritmo

Pasos del algoritmo:

- Encuentra un vertice inicial factible

Método Simplex - Pasos del algoritmo

Pasos del algoritmo:

- Encuentra un vertice inicial factible
- Evalua vertices adyacentes

Método Simplex - Pasos del algoritmo

Pasos del algoritmo:

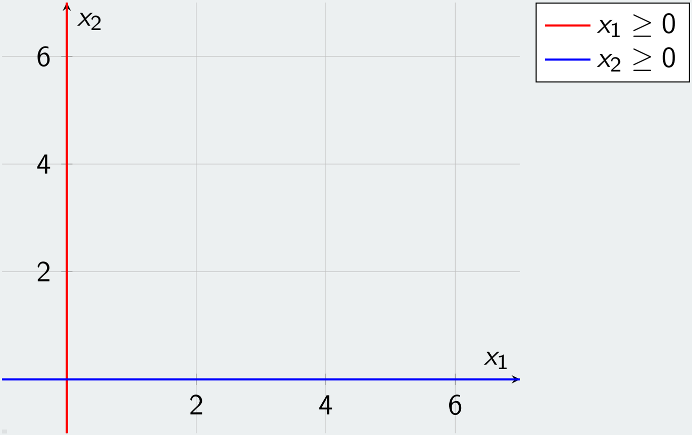
- Encuentra un vertice inicial factible
- Evalua vertices adyacentes
- Se mueve al mejor vertice adyacente

Método Simplex - Pasos del algoritmo

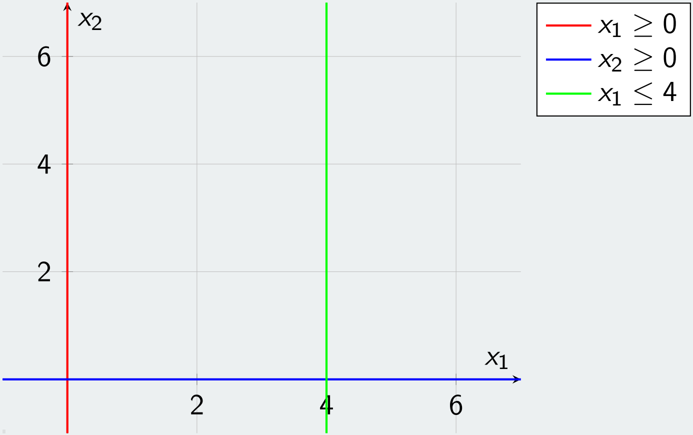
Pasos del algoritmo:

- Encuentra un vertice inicial factible
- Evalua vertices adyacentes
- Se mueve al mejor vertice adyacente
- Repite hasta encontrar el optimo

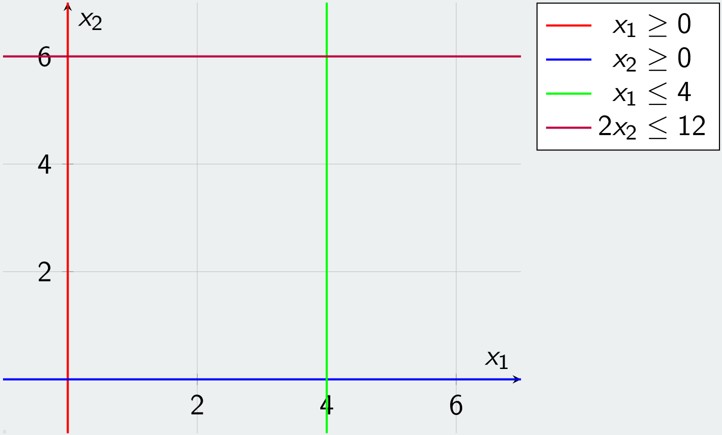
Método Simplex - Paso 1: Restricciones de No Negatividad



Método Simplex - Paso 2: Agregamos $x_1 \leq 4$



Método Simplex - Paso 3: Agregamos $2x_2 \leq 12$



Método Simplex - Paso 4: Agregamos $3x_1 + 2x_2 \leq 18$

Wyndor Glass Co. - Solución Óptima

Evaluación en vértices:

- $(0, 0)$: $Z = 0$

Wyndor Glass Co. - Solución Óptima

Evaluación en vértices:

- $(0, 0)$: $Z = 0$
- $(0, 6)$: $Z = 30000$

Wyndor Glass Co. - Solución Óptima

Evaluación en vértices:

- $(0, 0)$: $Z = 0$
- $(0, 6)$: $Z = 30000$
- $(2, 6)$: $Z = 36000 \leftarrow$ **ÓPTIMO**

Wyndor Glass Co. - Solución Óptima

Evaluación en vértices:

- $(0, 0)$: $Z = 0$
- $(0, 6)$: $Z = 30000$
- $(2, 6)$: $Z = 36000 \leftarrow$ **ÓPTIMO**
- $(4, 3)$: $Z = 27000$

Wyndor Glass Co. - Solución Óptima

Evaluación en vértices:

- $(0, 0)$: $Z = 0$
- $(0, 6)$: $Z = 30000$
- $(2, 6)$: $Z = 36000 \leftarrow$ **ÓPTIMO**
- $(4, 3)$: $Z = 27000$
- $(4, 0)$: $Z = 12000$

Wyndor Glass Co. - Solución Óptima

Evaluación en vértices:

- $(0, 0)$: $Z = 0$
- $(0, 6)$: $Z = 30000$
- $(2, 6)$: $Z = 36000 \leftarrow$ **ÓPTIMO**
- $(4, 3)$: $Z = 27000$
- $(4, 0)$: $Z = 12000$

Solución óptima:

- Producir 2 lotes del Producto 1
- Producir 6 lotes del Producto 2
- Ganancia máxima: \$36000

Implementación con PICOS

¿Qué es PICOS?

- Interfaz de Python para solucionadores de optimización
- Simplifica la formulación de problemas de PL
- Se conecta con solucionadores como GLPK, CPLEX, Gurobi

Implementación con PICOS

¿Qué es PICOS?

- Interfaz de Python para solucionadores de optimización
- Simplifica la formulación de problemas de PL
- Se conecta con solucionadores como GLPK, CPLEX, Gurobi

Ventajas:

- Sintaxis clara y legible
- Manejo automático de matrices
- Múltiples solucionadores disponibles
- Integración con NumPy

PICOS - Ejemplo Wyndor

```python
import picos
import numpy as np

# Crear problema
P = picos.Problem()

# Definir variables
x = picos.RealVariable('x', 2)

# Funcion objetivo
P.set_objective('max', 3000*x[0] + 5000*x[1])

# Restricciones
P.add_constraint(x[0] <= 4)
P.add_constraint(2*x[1] <= 12)
P.add_constraint(3*x[0] + 2*x[1] <= 18)

# Resolver
P.solve(solver='glpk')
print(x)  # [2.0, 6.0]
print(P.value)  # 36000.0
```

PICOS - Forma Matricial

```
P = picos.Problem()
x = picos.RealVariable('x', 2)

# Definir matrices
A = np.array([[1,0], [0,2], [3,2]])
c = np.array([3000,5000])
b = np.array([4,12,18])

# Convertir a constantes PICOS
c = picos.Constant('c', c)
A = picos.Constant('A', A)
b = picos.Constant('b', b)

# Funcion objetivo y restricciones
P.set_objective('max', c|x)
P.add_constraint(A*x <= b)

P.solve(solver='glpk')
```

Casos Especiales en PL

Tipos de soluciones:

- **Solución única:** El óptimo está en un único vértice

Casos Especiales en PL

Tipos de soluciones:

- **Solución única:** El óptimo está en un único vértice
- **Múltiples soluciones:** El óptimo está en una arista completa

Casos Especiales en PL

Tipos de soluciones:

- **Solución única:** El óptimo está en un único vértice
- **Múltiples soluciones:** El óptimo está en una arista completa
- **Problema no acotado:** La función objetivo puede crecer infinitamente

Casos Especiales en PL

Tipos de soluciones:

- **Solución única:** El óptimo está en un único vértice
- **Múltiples soluciones:** El óptimo está en una arista completa
- **Problema no acotado:** La función objetivo puede crecer infinitamente
- **Problema infactible:** No existe solución que satisfaga todas las restricciones

Casos Especiales en PL

Tipos de soluciones:

- **Solución única:** El óptimo está en un único vértice
- **Múltiples soluciones:** El óptimo está en una arista completa
- **Problema no acotado:** La función objetivo puede crecer infinitamente
- **Problema infactible:** No existe solución que satisfaga todas las restricciones

¿Cómo identificarlos?

- El solucionador nos lo indica
- Análisis gráfico (en 2D)
- Verificación de las restricciones

Próximos Pasos

En la siguiente clase veremos:

- Problemas de Redes
- Problema de Transporte (balanceado y desbalanceado)
- Problema de Transshipment
- Problema de Flujo Máximo

Para practicar:

- Resolver los ejercicios de lámparas y dieta óptima
- Resolver los ejercicios de la guía
- Experimentar con PICOS
- Identificar problemas de PL en situaciones reales

¿Dudas?
¿Consultas?

Universidad de SanAndrés