

Programación Lineal

Programacion entera

Clase 22

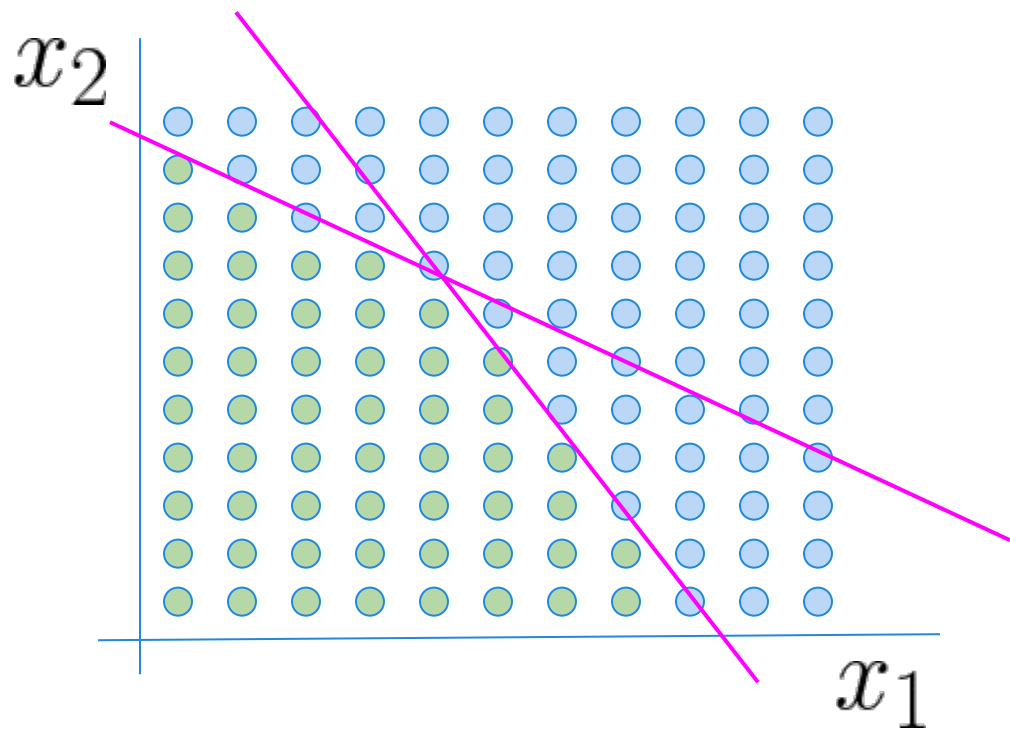
Investigación Operativa UTN FRBA

Curso: I4051

Docente: Martín Palazzo

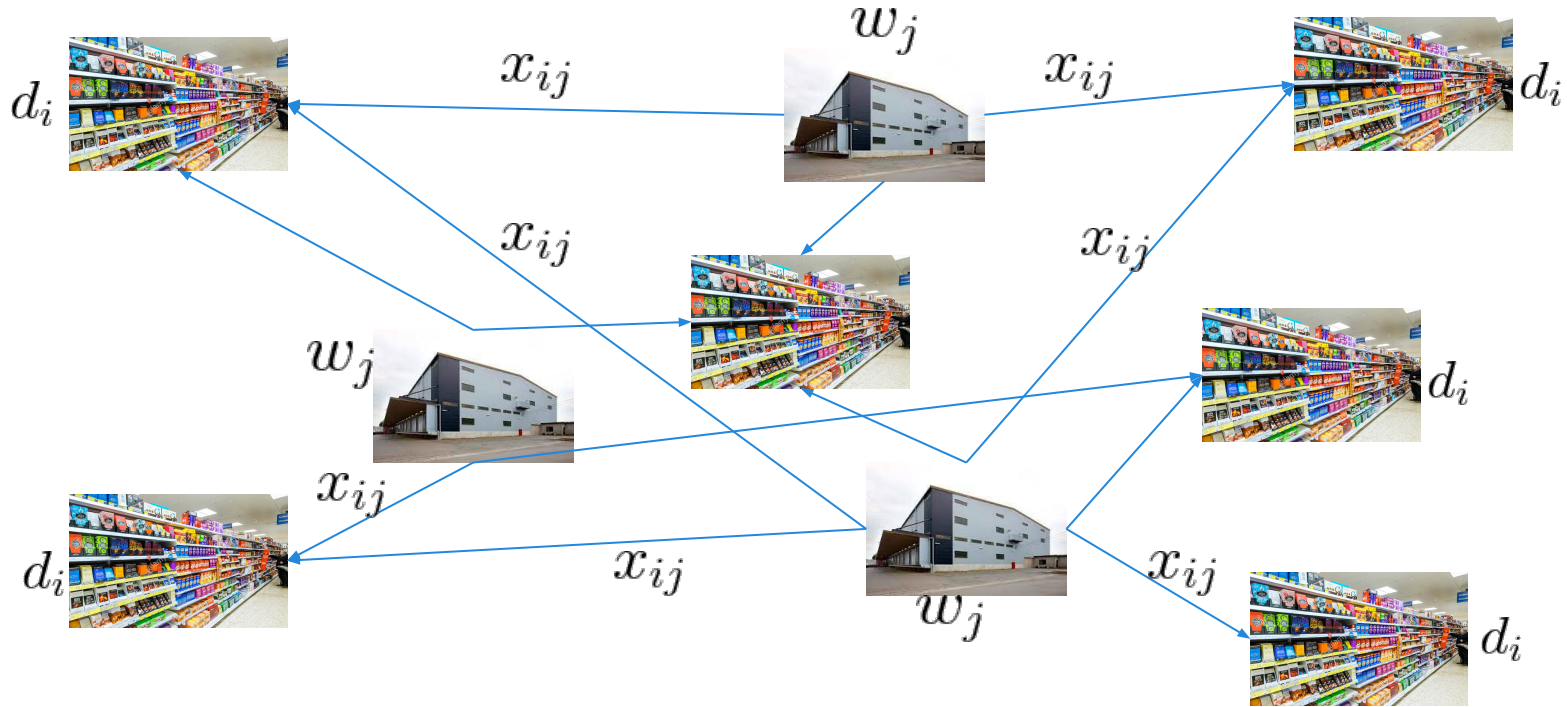
Programacion entera

$$\begin{aligned} & \max_x c^T x \\ & \text{s.t. } Ax \leq b \\ & \quad x \geq 0 \\ & \quad x \in \mathbb{Z}^d \\ & x \in [0, 1, 2, 3, 4, \dots,] \end{aligned}$$



Ejercicio warehouse location

Warehouse Location problem



En este tipo de problema tenemos multiples fuentes de demanda d_i . Existen multiples ubicaciones donde es posible construir un centro de distribucion, cada una con un costo c_{ij} de transportar de dicha ubicacion a cada centro de consumo. Se debe decidir en que ubicacion se construye un centro w_j sabiendo que tiene un costo de construir u_j y determinar a que centro de consumo abastecera.

Warehouse location problem

$$\min_x \quad \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{j=1}^m u_j w_j$$

c_{ij} = costo de transportar del centro j al cliente i

x_{ij} = el centro j provee al cliente i

u_j = costo fijo de construir el centro j

w_j = variable de existencia del centro j

d_i = demanda cliente i

$$x_{ij} \in \mathbb{Z} = \{0, 1, 2, 3, 4, \dots\} \qquad w_j \in \{0, 1\}$$

Warehouse location problem

$$\min_x \quad \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{j=1}^m u_j w_j$$

$$\text{s.t.} \quad \sum_{j=1}^m x_{ij} = d_i$$

Que cada centro reciba todo lo que demanda.

$$\text{s.t.} \quad \sum_{j=1}^n x_{ij} = M_j w_j$$

Solo activar las X asociadas a centros que se construyen

$$x_{ij} \leq d_i w_j$$

Lo que se envia a un centro no puede superar lo que demanda dicho centro.

$$x_{ij} \in \mathbb{Z} = \{0, 1, 2, 3, \dots\}$$

Cantidad a enviar desde un centro J a un consumo i

$$w_j \in \{0, 1\}$$

El centro en la ubicacion J se construye o no.

Warehouse location problem: datos del problema

```
# definimos la cantidad de clientes
clientes = [1,2,3,4,5]
# definimos la cantidad de depositos
warehouses = ['w1','w2','w3']

# demanda de cada cliente
demanda = {1:80,
           2:270,
           3:250,
           4:160,
           5:180}

# costo de activar cada warehouse
wcost = {'w1':1000,'w2':1000,'w3':1000}

# maxima cantidad que puede mover cada warehouse
max_q = {'w1':500,'w2':500,'w3':500}

# costo de transporte warehouse-cliente
transp_c = {'w1': {1 : 4, 2:5 , 3:6 , 4:8, 5:10},
            'w2': {1 : 6, 2:4 , 3:3 , 4:5, 5:8},
            'w3': {1 : 9, 2:7 , 3:4 , 4:3, 5:4}}
```

Warehouse location problem: ecuaciones

```
# definir el problema en pulp
opt = LpProblem("warehouseLocation", LpMinimize)

# definir las variables de decision Xij
xij = LpVariable.dicts("Servicio", [(i,j) for i in clientes
                                     for j in warehouses], 0)

# definir la Wj
Uj = LpVariable.dicts("UsarLocacion", warehouses, 0, 1, LpBinary)

# funcion objetivo
opt += lpSum(wcost[j]*Uj[j] for j in warehouses) + lpSum(transp_c[j][i]*xij[(i,j)] for j in warehouses
for i in clientes)

# restricciones 1
for i in clientes:
    opt += lpSum(xij[(i,j)] for j in warehouses) == demanda[i]

# restruccion 1
for j in warehouses:
    opt += lpSum(xij[(i,j)] for i in clientes) <= max_q[j]*Uj[j]

for i in clientes:
    for j in warehouses:
        opt += xij[(i,j)] <= demanda[i]*Uj[j]
```

$$\min_x \quad \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{j=1}^m u_j w_j$$

$$\text{s.t.} \quad \sum_{j=1}^m x_{ij} = d_i$$

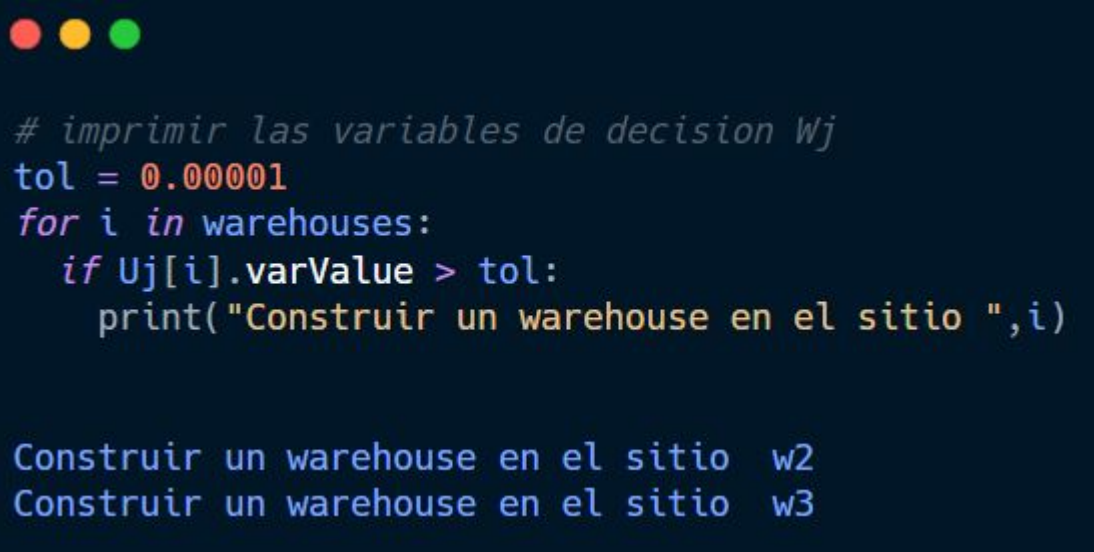
$$\text{s.t.} \quad \sum_{j=1}^n x_{ij} = M_j w_i$$

$$x_{ij} \leq d_i w_j$$

$$x_{ij} \in \mathbb{Z} = \{0, 1, 2, 3, \dots\}$$

$$w_j \in \{0, 1\}$$

Warehouse location problem: solucion

```
  
# imprimir las variables de decision Wj  
tol = 0.00001  
for i in warehouses:  
    if Uj[i].varValue > tol:  
        print("Construir un warehouse en el sitio ",i)
```

```
Construir un warehouse en el sitio  w2  
Construir un warehouse en el sitio  w3
```

Warehouse location problem: solucion

```

# imprimir las variables de decision Xij
for q in opt.variables():
    print(q.name,"=",q.varValue)

Servicio_(1,_'w1') = 0.0
Servicio_(1,_'w2') = 80.0
Servicio_(1,_'w3') = 0.0
Servicio_(2,_'w1') = 0.0
Servicio_(2,_'w2') = 270.0
Servicio_(2,_'w3') = 0.0
Servicio_(3,_'w1') = 0.0
Servicio_(3,_'w2') = 150.0
Servicio_(3,_'w3') = 100.0
Servicio_(4,_'w1') = 0.0
Servicio_(4,_'w2') = 0.0
Servicio_(4,_'w3') = 160.0
Servicio_(5,_'w1') = 0.0
Servicio_(5,_'w2') = 0.0
Servicio_(5,_'w3') = 180.0
UsarLocacion_w1 = 0.0
UsarLocacion_w2 = 1.0
UsarLocacion_w3 = 1.0

# imprimir la funcion objetivo en el optimo
print('El costo de la operacion es de = ', value(opt.objective))

El costo de la operacion es de = 5610.0
```