Programación Lineal Programacion entera Clase 22

Investigación Operativa UTN FRBA
Curso: I4051
Docente: Martín Palazzo

Programacion entera

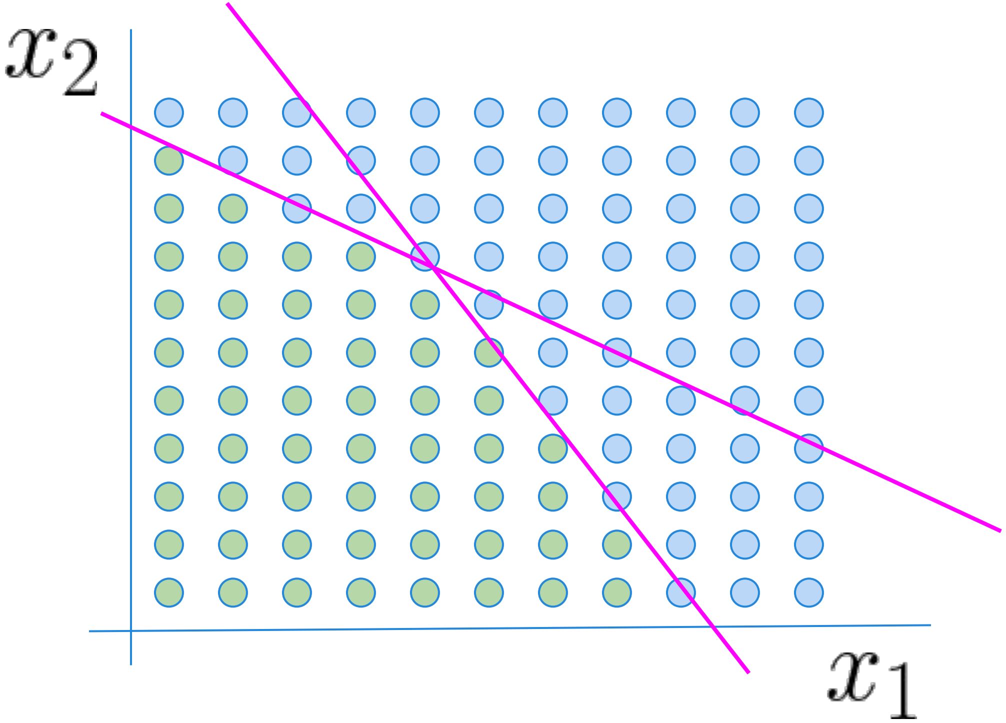
$$\max_x c^T x$$

$$\text{s.t. } Ax \leq b$$

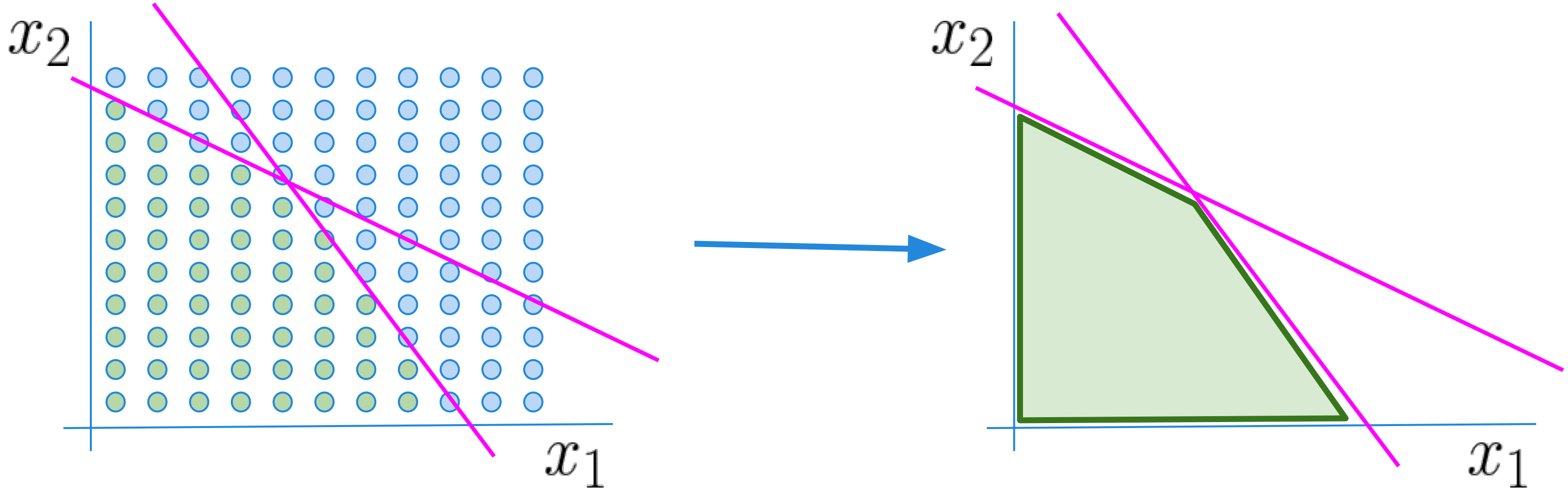
$$x \geq 0$$

$$x \in \mathbb{Z}^d$$

$$x \in [0, 1, 2, 3, 4, ...,]$$



Programacion entera: relajación via programacion lineal



$$\max_{x}\ c^T x \quad \text{s.t.}\ \ Ax \leq b,\quad x \geq 0,\quad x \in \mathbb{Z}^d$$

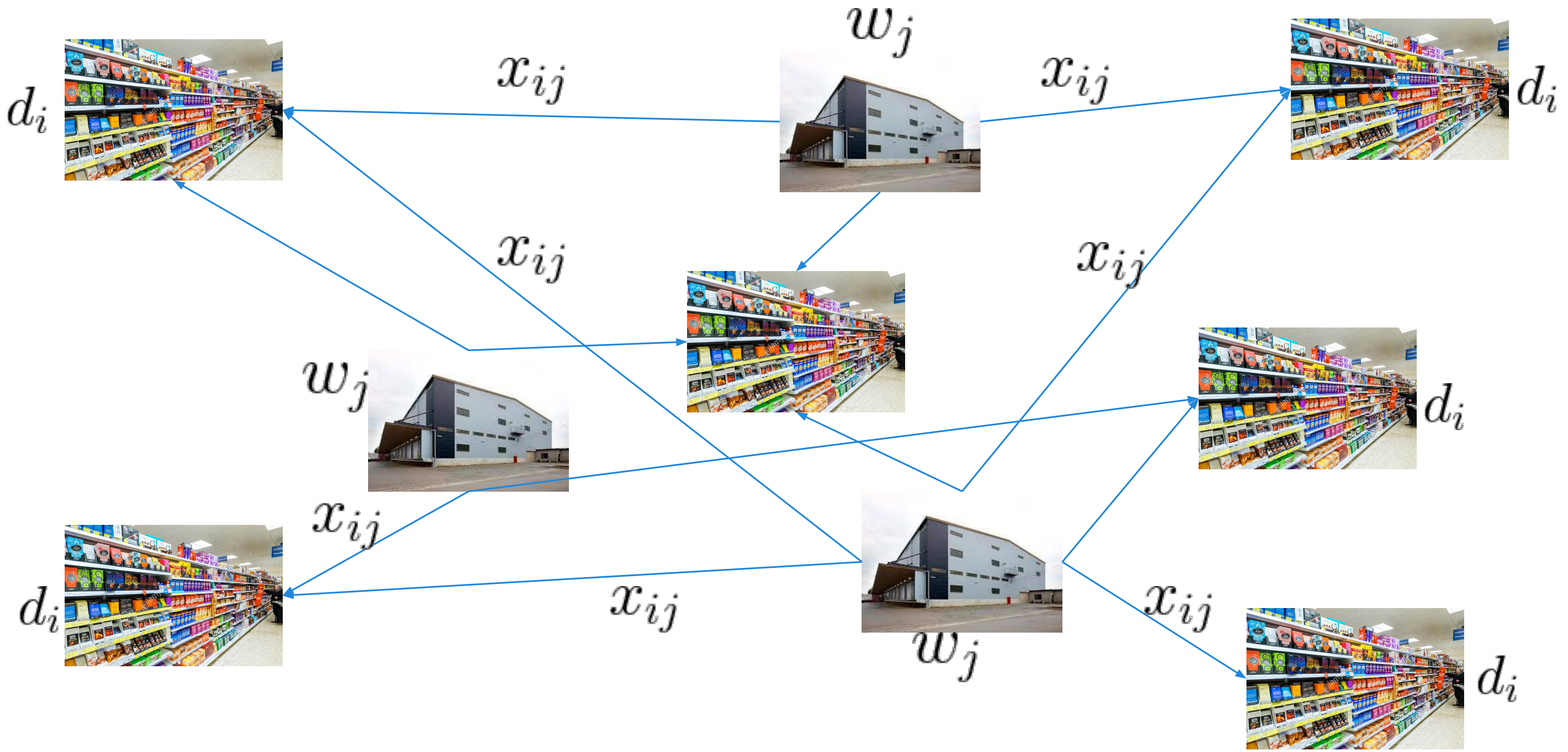
$x \in [0, 1, 2, 3, 4, ...,]$

$$\max_{x}\ c^T x \quad \text{s.t.}\ \ Ax \leq b,\quad x \geq 0,\quad x \in \mathbb{R}^d$$

Un problema de programación entera puede relajarse al transformar las variables de decisión enteras a variables de decisión reales continuas. La solución obtenida de P.L. puede redondearse a una solución factible en programacion entera.

Ejercicio warehouse location

Warehouse Location problem



Warehouse location problem

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

$$\begin{aligned}
c_{ij} &= \text{costo de transportar del centro j al cliente i} \\
x_{ij} &= \text{el centro j provee al cliente i} \\
u_j &= \text{costo fijo de construir el centro j} \\
w_j &= \text{variable de existencia del centro j} \\
d_i &= \text{demanda cliente i}
\end{aligned}$$

$$x_{ij} \in \{0, 1\} \qquad w_j \in \{0, 1\}$$

Warehouse location problem

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

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

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

$$x_{ij} \in \{0, 1\}$$

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

Warehouse location problem: datos del problema

```python
# 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

```python
# 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]
```

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

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

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

$$x_{ij} \in \{0, 1\}$$

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

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
```

ioperativ_clase22_warehouse_location_problem.ipynb