```
In [1]:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from gurobipy import *
```

Parte I: Facility Location Problem Capacitado Mono-Producto y Mono-Periodo

a) Caso restricción:

$$Y_{ij} < X_i \tag{1}$$

Conjuntos:

- I = Bodegas = 10
- J = Clientes = 20

Parámetros:

CF_i = Costo fijo de instalar la bodega i

R_i = Costo unitario de transportar productos desde la bodega i

T_{ij} = Costo unitario de transporte productos desde la bodega i hasta el cliente j

Q_i = Capacidad de almacenamiento de productos de la bodega i

d_j = Demanda de productos del cliente j

```
In [2]:
bodegas = 10
clientes = 20
I = [i+1 for i in range(bodegas)]
J = [j+1 for j in range(clientes)]
arcos=[(i,j) for i in I for j in J]

np.random.seed(2)
x_bodegas=list(np.random.random(len(I))*100)
y_bodegas=list(np.random.random(len(I))*100)
x_clientes=list(np.random.random(len(J))*100)
y_clientes=list(np.random.random(len(J))*100)
distancias={(i+1,j+1):np.hypot(x_bodegas[i]-x_clientes[j],y_bodegas[i]-y_clientes[j]) for i in range(len(I)) fo

K = {i:np.random.randint(2000,3000) for i in I}
R = {i:np.random.randint(100,300) for i in I}
T = distancias
Q = {i:np.random.randint(300,800) for i in I}
D = {j:np.random.randint(200,300) for j in J}
```

Variables de decisión:

X_i = 1 Si se instala una bodega en el sitio i, 0 de otro modo.

Y_{ij} = 1 Si se asigna el cliente j a la bodega i, 0 de otro modo.

```
In [3]:
model_FLP_a = Model('model_FLP_a')
#model_FLP_a.setParam('LogToConsole', 0)
```

```
Academic license - for non-commercial use only - expires 2022-01-10
Using license file C:\Users\chris\gurobi.lic
```

```
In [4]:
IJ = [(i,j) for i in I for j in J]
X = model_FLP_a.addVars(I, vtype=GRB.BINARY, name='X')
Y = model_FLP_a.addVars(IJ, vtype=GRB.BINARY, name='Y')
```

Modelo

$$Min\sum_{i=1}^{I}K_i*X_i+\sum_{i=1}^{I}\sum_{j=1}^{J}(T_{ij}+R_i)*D_j*Y_{ij} \tag{2}$$

$$\sum_{i=1}^{I}Y_{ij}=1 \qquad \forall j=1,\dots,J \tag{3}$$

$$Y_{ij}\le X_i \qquad \forall i=1,\dots,I,\forall j=1,\dots,J \tag{4}$$

$$\sum_{i=1}^{I}X_i\ge 1 \tag{5}$$

$$\sum_{j=1}^{J}Y_{ij}*D_j\le X_i*Q_i \qquad \forall i=1,\dots,I \tag{6}$$

```
In [5]:
objective = quicksum(K[i]*X[i] for i in I) + quicksum((T[(i,j)]+R[i])*D[j]*Y[(i,j)] for i,j in IJ)
model_FLP_a.setObjective(expr=objective, sense=GRB.MINIMIZE)
model_FLP_a.addConstrs(quicksum(Y[(i,j)] for i in I)==1 for j in J)
model_FLP_a.addConstrs(Y[(i,j)]<=X[i] for i,j in IJ)
model_FLP_a.addConstr(quicksum(X[i] for i in I)>=1)
model_FLP_a.addConstrs(quicksum(Y[(i,j)]*D[j] for j in J)<=X[i]*Q[i] for i in I)
model_FLP_a.update()
```

```
In [6]:
model_FLP_a.optimize()
```

```
Gurobi Optimizer version 9.1.2 build v9.1.2rc0 (win64)
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads
Optimize a model with 231 rows, 210 columns and 820 nonzeros
Model fingerprint: 0xe24a6303
Variable types: 0 continuous, 210 integer (210 binary)
Coefficient statistics:
  Matrix range     [1e+00, 8e+02]
  Objective range  [2e+03, 1e+05]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 1303466.2293
Presolve removed 56 rows and 0 columns
Presolve time: 0.00s
Presolved: 175 rows, 210 columns, 841 nonzeros
Variable types: 0 continuous, 210 integer (210 binary)

Root relaxation: objective 1.121027e+06, 89 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 1121027.03    0   16 1303466.23 1121027.03  14.0%     -    0s
H    0     0                    1140197.2413 1121027.03  1.68%     -    0s
H    0     0                    1140054.2543 1121130.58  1.66%     -    0s
H    0     0                    1139100.7472 1121130.58  1.58%     -    0s
     0     0 1128136.50    0   24 1139100.75 1128136.50  0.96%     -    0s
H    0     0                    1137338.1399 1128136.50  0.81%     -    0s
     0     0 1136129.83    0   12 1137338.14 1136129.83  0.11%     -    0s
     0     0 1136783.85    0   12 1137338.14 1136783.85  0.05%     -    0s
     0     0     cutoff    0      1137338.14 1137338.14  0.00%     -    0s

Explored 1 nodes (168 simplex iterations) in 0.05 seconds
Thread count was 8 (of 8 available processors)

Solution count 5: 1.13734e+06 1.1391e+06 1.14005e+06 ... 1.30347e+06

Optimal solution found (tolerance 1.00e-04)
Best objective 1.137338139886e+06, best bound 1.137338139886e+06, gap 0.0000%
```

```
In [7]:
for v in model_FLP_a.getVars():
    print(str(v.VarName)+'='+str(round(v.x,2)))
```

```
X[1]=1.0
X[2]=1.0
X[3]=1.0
X[4]=1.0
X[5]=1.0
X[6]=1.0
X[7]=1.0
X[8]=1.0
X[9]=1.0
X[10]=1.0
Y[1,1]=-0.0
Y[1,2]=-0.0
Y[1,3]=-0.0
Y[1,4]=-0.0
Y[1,5]=1.0
Y[1,6]=-0.0
Y[1,7]=-0.0
Y[1,8]=-0.0
Y[1,9]=1.0
Y[1,10]=-0.0
Y[1,11]=1.0
Y[1,12]=-0.0
Y[1,13]=-0.0
Y[1,14]=-0.0
Y[1,15]=-0.0
Y[1,16]=-0.0
Y[1,17]=-0.0
Y[1,18]=-0.0
Y[1,19]=-0.0
Y[1,20]=-0.0
Y[2,1]=-0.0
Y[2,2]=1.0
Y[2,3]=-0.0
Y[2,4]=-0.0
Y[2,5]=-0.0
Y[2,6]=-0.0
Y[2,7]=-0.0
Y[2,8]=-0.0
Y[2,9]=-0.0
Y[2,10]=-0.0
Y[2,11]=-0.0
Y[2,12]=-0.0
Y[2,13]=-0.0
Y[2,14]=-0.0
Y[2,15]=-0.0
Y[2,16]=-0.0
Y[2,17]=-0.0
Y[2,18]=-0.0
Y[2,19]=-0.0
Y[2,20]=-0.0
Y[3,1]=-0.0
Y[3,2]=-0.0
Y[3,3]=-0.0
Y[3,4]=-0.0
Y[3,5]=-0.0
Y[3,6]=-0.0
Y[3,7]=-0.0
Y[3,8]=-0.0
Y[3,9]=-0.0
Y[3,10]=-0.0
Y[3,11]=-0.0
Y[3,12]=-0.0
Y[3,13]=-0.0
Y[3,14]=-0.0
Y[3,15]=-0.0
Y[3,16]=-0.0
Y[3,17]=1.0
Y[3,18]=1.0
Y[3,19]=-0.0
Y[3,20]=-0.0
Y[4,1]=-0.0
Y[4,2]=0.0
Y[4,3]=-0.0
Y[4,4]=-0.0
Y[4,5]=-0.0
Y[4,6]=-0.0
Y[4,7]=-0.0
Y[4,8]=0.0
Y[4,9]=-0.0
Y[4,10]=-0.0
Y[4,11]=-0.0
Y[4,12]=-0.0
Y[4,13]=1.0
Y[4,14]=-0.0
Y[4,15]=-0.0
Y[4,16]=-0.0
Y[4,17]=-0.0
Y[4,18]=-0.0
Y[4,19]=-0.0
Y[4,20]=-0.0
Y[5,1]=-0.0
Y[5,2]=-0.0
Y[5,3]=-0.0
Y[5,4]=1.0
Y[5,5]=-0.0
Y[5,6]=-0.0
Y[5,7]=1.0
Y[5,8]=-0.0
Y[5,9]=-0.0
Y[5,10]=-0.0
Y[5,11]=-0.0
Y[5,12]=-0.0
Y[5,13]=-0.0
Y[5,14]=-0.0
Y[5,15]=-0.0
Y[5,16]=-0.0
Y[5,17]=-0.0
Y[5,18]=-0.0
Y[5,19]=-0.0
Y[5,20]=-0.0
Y[6,1]=1.0
Y[6,2]=-0.0
Y[6,3]=-0.0
Y[6,4]=-0.0
Y[6,5]=-0.0
Y[6,6]=-0.0
Y[6,7]=-0.0
Y[6,8]=-0.0
Y[6,9]=-0.0
Y[6,10]=-0.0
Y[6,11]=-0.0
Y[6,12]=-0.0
Y[6,13]=-0.0
Y[6,14]=1.0
Y[6,15]=-0.0
Y[6,16]=-0.0
Y[6,17]=-0.0
Y[6,18]=-0.0
Y[6,19]=-0.0
Y[6,20]=-0.0
Y[7,1]=-0.0
Y[7,2]=0.0
Y[7,3]=1.0
Y[7,4]=-0.0
Y[7,5]=-0.0
Y[7,6]=-0.0
Y[7,7]=-0.0
Y[7,8]=1.0
Y[7,9]=-0.0
Y[7,10]=-0.0
Y[7,11]=-0.0
Y[7,12]=-0.0
Y[7,13]=0.0
Y[7,14]=-0.0
Y[7,15]=-0.0
Y[7,16]=-0.0
Y[7,17]=-0.0
Y[7,18]=-0.0
Y[7,19]=-0.0
Y[7,20]=-0.0
Y[8,1]=-0.0
Y[8,2]=-0.0
Y[8,3]=-0.0
Y[8,4]=-0.0
Y[8,5]=-0.0
Y[8,6]=1.0
Y[8,7]=-0.0
Y[8,8]=-0.0
Y[8,9]=-0.0
Y[8,10]=-0.0
Y[8,11]=-0.0
Y[8,12]=-0.0
Y[8,13]=-0.0
Y[8,14]=-0.0
Y[8,15]=1.0
Y[8,16]=-0.0
Y[8,17]=-0.0
Y[8,18]=-0.0
Y[8,19]=-0.0
Y[8,20]=1.0
Y[9,1]=-0.0
Y[9,2]=-0.0
Y[9,3]=-0.0
Y[9,4]=-0.0
Y[9,5]=-0.0
Y[9,6]=-0.0
Y[9,7]=-0.0
Y[9,8]=-0.0
Y[9,9]=-0.0
Y[9,10]=1.0
Y[9,11]=-0.0
Y[9,12]=-0.0
Y[9,13]=-0.0
Y[9,14]=-0.0
Y[9,15]=-0.0
Y[9,16]=-0.0
Y[9,17]=-0.0
Y[9,18]=-0.0
Y[9,19]=1.0
Y[9,20]=-0.0
Y[10,1]=-0.0
Y[10,2]=-0.0
Y[10,3]=-0.0
Y[10,4]=-0.0
Y[10,5]=-0.0
Y[10,6]=-0.0
Y[10,7]=-0.0
Y[10,8]=-0.0
Y[10,9]=-0.0
Y[10,10]=-0.0
Y[10,11]=-0.0
Y[10,12]=1.0
Y[10,13]=-0.0
Y[10,14]=-0.0
Y[10,15]=-0.0
Y[10,16]=1.0
Y[10,17]=-0.0
Y[10,18]=-0.0
Y[10,19]=-0.0
Y[10,20]=-0.0
```

```
In [8]:
arcos_activos=[k for k in arcos if Y[k].x>0.99]
```

```
In [9]:
model_FLP_a.ObjVal
```

Out[9]: 1137338.1398864486

```
In [10]:
model_FLP_a.Runtime
```

Out[10]: 0.05983924865722656

```
In [11]:
model_FLP_a_relaxed = model_FLP_a.relax()
model_FLP_a_relaxed.optimize()
```

```
Gurobi Optimizer version 9.1.2 build v9.1.2rc0 (win64)
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads
Optimize a model with 231 rows, 210 columns and 820 nonzeros
Model fingerprint: 0xb3c6a492
Coefficient statistics:
  Matrix range     [1e+00, 8e+02]
  Objective range  [2e+03, 1e+05]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Presolve time: 0.01s
Presolved: 231 rows, 210 columns, 820 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    7.9767608e+05   6.873438e+02   0.000000e+00      0s
      59    1.1084056e+06   0.000000e+00   0.000000e+00      0s

Solved in 59 iterations and 0.01 seconds
Optimal objective  1.108405554e+06
```

```
In [12]:
for v in model_FLP_a_relaxed.getVars():
    print(str(v.VarName)+'='+str(round(v.x,2)))
```

```
X[1]=1.0
X[2]=1.0
X[3]=1.0
X[4]=0.0
X[5]=1.0
X[6]=0.82
X[7]=1.0
X[8]=1.0
X[9]=1.0
X[10]=1.0
Y[1,1]=0.0
Y[1,2]=0.0
Y[1,3]=0.0
Y[1,4]=0.0
Y[1,5]=0.59
Y[1,6]=0.0
Y[1,7]=0.0
Y[1,8]=0.0
Y[1,9]=1.0
Y[1,10]=0.0
Y[1,11]=1.0
Y[1,12]=0.0
Y[1,13]=0.0
Y[1,14]=0.0
Y[1,15]=0.0
Y[1,16]=0.0
Y[1,17]=0.0
Y[1,18]=0.0
Y[1,19]=0.0
Y[1,20]=0.43
Y[2,1]=0.0
Y[2,2]=1.0
Y[2,3]=0.0
Y[2,4]=0.0
Y[2,5]=0.41
Y[2,6]=0.0
Y[2,7]=0.0
Y[2,8]=0.0
Y[2,9]=0.0
Y[2,10]=0.0
Y[2,11]=0.0
Y[2,12]=0.0
Y[2,13]=0.0
Y[2,14]=0.0
Y[2,15]=0.0
Y[2,16]=0.0
Y[2,17]=0.0
Y[2,18]=0.0
Y[2,19]=0.0
Y[2,20]=0.0
Y[3,1]=0.0
Y[3,2]=0.0
Y[3,3]=0.0
Y[3,4]=0.0
Y[3,5]=0.0
Y[3,6]=0.0
Y[3,7]=0.0
Y[3,8]=0.0
Y[3,9]=0.0
Y[3,10]=0.0
Y[3,11]=0.0
Y[3,12]=0.0
Y[3,13]=0.0
Y[3,14]=0.0
Y[3,15]=0.0
Y[3,16]=0.12
Y[3,17]=1.0
Y[3,18]=1.0
Y[3,19]=0.0
Y[3,20]=0.0
Y[4,1]=0.0
Y[4,2]=0.0
Y[4,3]=0.0
Y[4,4]=0.0
Y[4,5]=0.0
Y[4,6]=0.0
Y[4,7]=0.0
Y[4,8]=0.0
Y[4,9]=0.0
Y[4,10]=0.0
Y[4,11]=0.0
Y[4,12]=0.0
Y[4,13]=0.0
Y[4,14]=0.0
Y[4,15]=0.0
Y[4,16]=0.0
Y[4,17]=0.0
Y[4,18]=0.0
Y[4,19]=0.0
Y[4,20]=0.0
Y[5,1]=0.0
Y[5,2]=0.0
Y[5,3]=0.0
Y[5,4]=0.58
Y[5,5]=0.0
Y[5,6]=0.0
Y[5,7]=1.0
Y[5,8]=0.0
Y[5,9]=0.0
Y[5,10]=0.0
Y[5,11]=0.0
Y[5,12]=0.0
Y[5,13]=0.0
Y[5,14]=0.0
Y[5,15]=0.0
Y[5,16]=0.88
Y[5,17]=0.0
Y[5,18]=0.0
Y[5,19]=0.0
Y[5,20]=0.0
Y[6,1]=0.0
Y[6,2]=0.0
Y[6,3]=0.74
Y[6,4]=0.0
Y[6,5]=0.0
Y[6,6]=0.0
Y[6,7]=0.0
Y[6,8]=0.0
Y[6,9]=0.0
Y[6,10]=0.0
Y[6,11]=0.0
Y[6,12]=0.0
Y[6,13]=0.0
Y[6,14]=0.82
Y[6,15]=0.0
Y[6,16]=0.0
Y[6,17]=0.0
Y[6,18]=0.0
Y[6,19]=0.0
Y[6,20]=0.36
Y[7,1]=0.0
Y[7,2]=0.0
Y[7,3]=0.0
Y[7,4]=0.0
Y[7,5]=0.0
Y[7,6]=0.0
Y[7,7]=0.0
Y[7,8]=1.0
Y[7,9]=0.0
Y[7,10]=0.0
Y[7,11]=0.0
Y[7,12]=0.0
Y[7,13]=0.0
Y[7,14]=0.0
Y[7,15]=0.0
Y[7,16]=0.0
Y[7,17]=0.0
Y[7,18]=0.0
Y[7,19]=0.72
Y[7,20]=0.0
Y[8,1]=0.0
Y[8,2]=0.0
Y[8,3]=0.0
Y[8,4]=0.0
Y[8,5]=0.0
Y[8,6]=1.0
Y[8,7]=0.0
Y[8,8]=0.0
Y[8,9]=0.0
Y[8,10]=0.0
Y[8,11]=0.0
Y[8,12]=0.0
Y[8,13]=1.0
Y[8,14]=0.0
Y[8,15]=1.0
Y[8,16]=0.0
Y[8,17]=0.0
Y[8,18]=0.0
Y[8,19]=0.0
Y[8,20]=0.21
Y[9,1]=1.0
Y[9,2]=0.0
Y[9,3]=0.26
Y[9,4]=0.0
Y[9,5]=0.0
Y[9,6]=0.0
Y[9,7]=0.0
Y[9,8]=0.0
Y[9,9]=0.0
Y[9,10]=1.0
Y[9,11]=0.0
Y[9,12]=0.0
Y[9,13]=0.0
Y[9,14]=0.18
Y[9,15]=0.0
Y[9,16]=0.0
Y[9,17]=0.0
Y[9,18]=0.0
Y[9,19]=0.28
Y[9,20]=0.0
Y[10,1]=0.0
Y[10,2]=0.0
Y[10,3]=0.0
Y[10,4]=0.42
Y[10,5]=0.0
Y[10,6]=0.0
Y[10,7]=0.0
Y[10,8]=0.0
Y[10,9]=0.0
Y[10,10]=0.0
Y[10,11]=0.0
Y[10,12]=1.0
Y[10,13]=0.0
Y[10,14]=0.0
Y[10,15]=0.0
Y[10,16]=0.0
Y[10,17]=0.0
Y[10,18]=0.0
Y[10,19]=0.0
Y[10,20]=0.0
```

```
In [13]:
model_FLP_a_relaxed.ObjVal
```

Out[13]: 1108405.5542667445

```
In [14]:
model_FLP_a_relaxed.Runtime
```

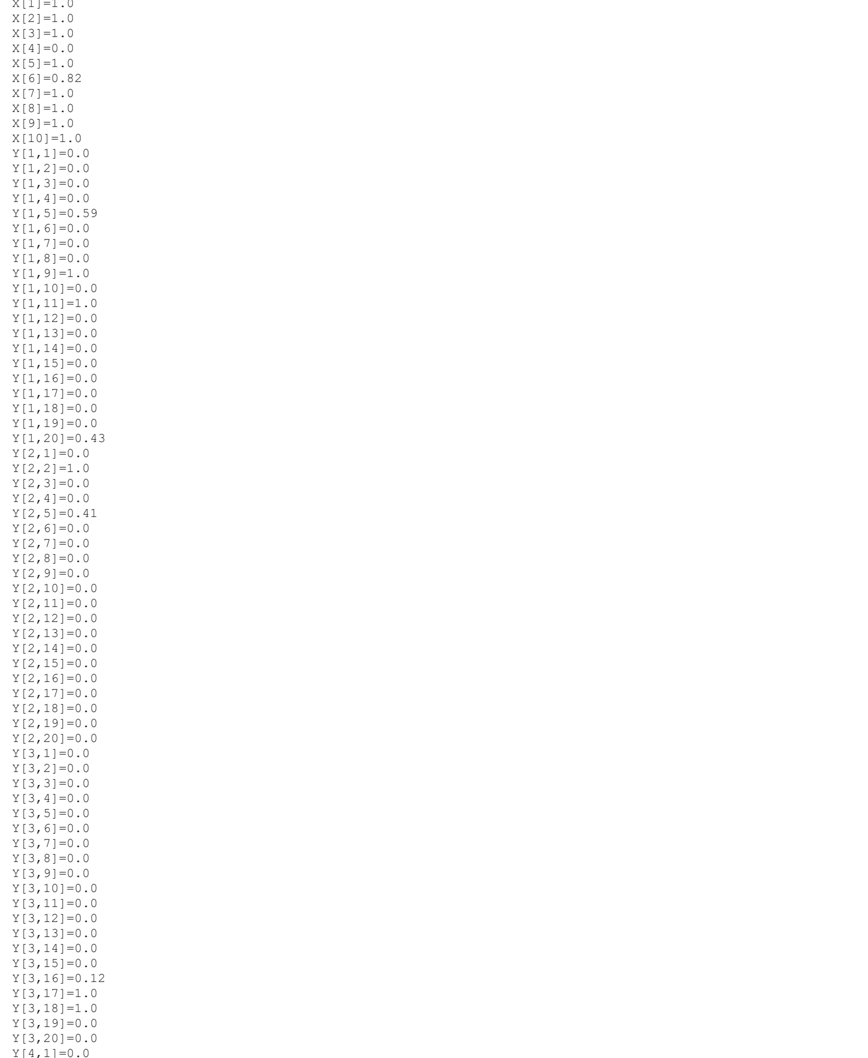
Out[14]: 0.012964248657226562

```
In [15]:
GAP = (model_FLP_a.ObjVal-model_FLP_a_relaxed.ObjVal)/model_FLP_a.ObjVal
round(GAP*100,2)
```

Out[15]: 2.54

Gráfico Solución

```
In [16]:
plt.figure(figsize=(14,11));
plt.scatter(x_bodegas,y_bodegas,color="blue",marker="s");
plt.scatter(x_clientes,y_clientes,color="red");

for i in range(len(I)):
    plt.annotate("$Bodega_{%d}$"%(i+1),(x_bodegas[i]+1,y_bodegas[i]-1),size=13, color="blue");

for j in range(len(J)):
    plt.annotate("$Cliente_{%d}$"%(j+1),(x_clientes[j]+1,y_clientes[j]-1.8),size=13, color="red",weight="bold")

for n in arcos_activos:
    i=n[0]
    j=n[1]
    plt.plot([x_bodegas[i-1],x_clientes[j-1]],[y_bodegas[i-1],y_clientes[j-1]],color="green")

plt.xlabel("Eje x");
plt.ylabel("Eje y");
plt.title("Solución $Y_{ij}$");
plt.show();
```



b) Caso restricción:

$$\sum_{j=1}^{J}Y_{ij} <= M*Xi \qquad \forall i=1,\dots,J \tag{7}$$

```
In [17]:
model_FLP_b = Model('model_FLP_b')
#model_FLP_b.setParam('LogToConsole', 0)
```

```
In [18]:
X,Y = None, None
M = len(J)
```

```
In [19]:
X = model_FLP_b.addVars(I, vtype=GRB.BINARY, name='X')
Y = model_FLP_b.addVars(IJ, vtype=GRB.BINARY, name='Y')
```

```
In [20]:
objective = quicksum(K[i]*X[i] for i in I) + quicksum((T[(i,j)]+R[i])*D[j]*Y[(i,j)] for i,j in IJ)
model_FLP_b.setObjective(expr=objective, sense=GRB.MINIMIZE)
model_FLP_b.addConstrs(quicksum(Y[(i,j)] for i in I)==1 for j in J)
model_FLP_b.addConstrs(quicksum(Y[(i,j)] for j in J)<=X[i]*M for i in I)
model_FLP_b.addConstr(quicksum(X[i] for i in I)>=1)
model_FLP_b.addConstrs(quicksum(Y[(i,j)]*D[j] for j in J)<=X[i]*Q[i] for i in I)
model_FLP_b.update()
model_FLP_b.optimize()
```

```
Gurobi Optimizer version 9.1.2 build v9.1.2rc0 (win64)
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads
Optimize a model with 41 rows, 210 columns and 630 nonzeros
Model fingerprint: 0xc3f84208
Variable types: 0 continuous, 210 integer (210 binary)
Coefficient statistics:
  Matrix range     [1e+00, 8e+02]
  Objective range  [2e+03, 1e+05]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 1303466.2293
Presolve removed 10 rows and 0 columns
Presolve time: 0.00s
Presolved: 31 rows, 210 columns, 420 nonzeros
Variable types: 0 continuous, 210 integer (210 binary)

Root relaxation: objective 1.117058e+06, 56 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 1117057.64    0   14 1303466.23 1117057.64  14.3%     -    0s
H    0     0                    1226058.8463 1117057.64  8.89%     -    0s
H    0     0                    1165174.5592 1117057.64  4.13%     -    0s
H    0     0                    1159395.0012 1117057.64  3.65%     -    0s
H    0     0                    1155969.3872 1117057.64  3.37%     -    0s
H    0     0                    1153016.1026 1117057.64  3.12%     -    0s
H    0     0                    1139100.7472 1117057.64  1.94%     -    0s
     0     0 1126068.92    0   10 1139100.75 1126068.92  1.14%     -    0s
H    0     0                    1138975.9082 1126068.92  1.13%     -    0s
H    0     0                    1137338.1399 1126068.92  0.99%     -    0s
     0     0 1131622.83    0   15 1137338.14 1131622.83  0.50%     -    0s
     0     0 1131622.83    0   19 1137338.14 1131622.83  0.50%     -    0s

Cutting planes:
  Gomory: 5
  Cover: 10
  MIR: 7
  StrongCG: 5
  Mod-K: 1
  RLT: 3

Explored 1 nodes (92 simplex iterations) in 0.05 seconds
Thread count was 8 (of 8 available processors)

Solution count 9: 1.13734e+06 1.13898e+06 1.1391e+06 ... 1.30347e+06

Optimal solution found (tolerance 1.00e-04)
Best objective 1.137338139886e+06, best bound 1.137338139886e+06, gap 0.0000%
```

```
In [21]:
for v in model_FLP_b.getVars():
    print(str(v.VarName)+'='+str(round(v.x,2)))
```

```
X[1]=1.0
X[2]=1.0
X[3]=1.0
X[4]=1.0
X[5]=1.0
X[6]=1.0
X[7]=1.0
X[8]=1.0
X[9]=1.0
X[10]=1.0
Y[1,1]=-0.0
Y[1,2]=-0.0
Y[1,3]=-0.0
Y[1,4]=-0.0
Y[1,5]=1.0
Y[1,6]=-0.0
Y[1,7]=-0.0
Y[1,8]=-0.0
Y[1,9]=1.0
Y[1,10]=-0.0
Y[1,11]=1.0
Y[1,12]=-0.0
Y[1,13]=0.0
Y[1,14]=-0.0
Y[1,15]=-0.0
Y[1,16]=-0.0
Y[1,17]=-0.0
Y[1,18]=-0.0
Y[1,19]=-0.0
Y[1,20]=0.0
Y[2,1]=-0.0
Y[2,2]=1.0
Y[2,3]=-0.0
Y[2,4]=-0.0
Y[2,5]=-0.0
Y[2,6]=-0.0
Y[2,7]=-0.0
Y[2,8]=-0.0
Y[2,9]=-0.0
Y[2,10]=-0.0
Y[2,11]=-0.0
Y[2,12]=-0.0
Y[2,13]=-0.0
Y[2,14]=-0.0
Y[2,15]=-0.0
Y[2,16]=-0.0
Y[2,17]=-0.0
Y[2,18]=-0.0
Y[2,19]=-0.0
Y[2,20]=-0.0
Y[3,1]=-0.0
Y[3,2]=-0.0
Y[3,3]=-0.0
Y[3,4]=-0.0
Y[3,5]=-0.0
Y[3,6]=-0.0
Y[3,7]=-0.0
Y[3,8]=-0.0
Y[3,9]=-0.0
Y[3,10]=-0.0
Y[3,11]=-0.0
Y[3,12]=-0.0
Y[3,13]=-0.0
Y[3,14]=-0.0
Y[3,15]=-0.0
Y[3,16]=-0.0
Y[3,17]=1.0
Y[3,18]=1.0
Y[3,19]=-0.0
Y[3,20]=-0.0
Y[4,1]=0.0
Y[4,2]=-0.0
Y[4,3]=-0.0
Y[4,4]=-0.0
Y[4,5]=-0.0
Y[4,6]=-0.0
Y[4,7]=-0.0
Y[4,8]=-0.0
Y[4,9]=-0.0
Y[4,10]=-0.0
Y[4,11]=0.0
Y[4,12]=-0.0
Y[4,13]=1.0
Y[4,14]=-0.0
Y[4,15]=-0.0
Y[4,16]=-0.0
Y[4,17]=-0.0
Y[4,18]=-0.0
Y[4,19]=-0.0
Y[4,20]=-0.0
Y[5,1]=-0.0
Y[5,2]=-0.0
Y[5,3]=-0.0
Y[5,4]=1.0
Y[5,5]=-0.0
Y[5,6]=-0.0
Y[5,7]=1.0
Y[5,8]=-0.0
Y[5,9]=-0.0
Y[5,10]=-0.0
Y[5,11]=-0.0
Y[5,12]=-0.0
Y[5,13]=-0.0
Y[5,14]=-0.0
Y[5,15]=-0.0
Y[5,16]=-0.0
Y[5,17]=-0.0
Y[5,18]=-0.0
Y[5,19]=-0.0
Y[5,20]=-0.0
Y[6,1]=1.0
Y[6,2]=-0.0
Y[6,3]=-0.0
Y[6,4]=-0.0
Y[6,5]=-0.0
Y[6,6]=-0.0
Y[6,7]=-0.0
Y[6,8]=-0.0
Y[6,9]=-0.0
Y[6,10]=0.0
Y[6,11]=-0.0
Y[6,12]=-0.0
Y[6,13]=0.0
Y[6,14]=1.0
Y[6,15]=-0.0
Y[6,16]=-0.0
Y[6,17]=-0.0
Y[6,18]=-0.0
Y[6,19]=-0.0
Y[6,20]=0.0
Y[7,1]=-0.0
Y[7,2]=-0.0
Y[7,3]=1.0
Y[7,4]=-0.0
Y[7,5]=-0.0
Y[7,6]=-0.0
Y[7,7]=-0.0
Y[7,8]=1.0
Y[7,9]=-0.0
Y[7,10]=-0.0
Y[7,11]=-0.0
Y[7,12]=-0.0
Y[7,13]=-0.0
Y[7,14]=-0.0
Y[7,15]=-0.0
Y[7,16]=-0.0
Y[7,17]=-0.0
Y[7,18]=-0.0
Y[7,19]=-0.0
Y[7,20]=-0.0
Y[8,1]=-0.0
Y[8,2]=-0.0
Y[8,3]=-0.0
Y[8,4]=-0.0
Y[8,5]=-0.0
Y[8,6]=1.0
Y[8,7]=-0.0
Y[8,8]=-0.0
Y[8,9]=-0.0
Y[8,10]=-0.0
Y[8,11]=-0.0
Y[8,12]=-0.0
Y[8,13]=-0.0
Y[8,14]=-0.0
Y[8,15]=1.0
Y[8,16]=-0.0
Y[8,17]=-0.0
Y[8,18]=-0.0
Y[8,19]=-0.0
Y[8,20]=1.0
Y[9,1]=0.0
Y[9,2]=-0.0
Y[9,3]=-0.0
Y[9,4]=-0.0
Y[9,5]=-0.0
Y[9,6]=-0.0
Y[9,7]=-0.0
Y[9,8]=-0.0
Y[9,9]=-0.0
Y[9,10]=1.0
Y[9,11]=-0.0
Y[9,12]=-0.0
Y[9,13]=-0.0
Y[9,14]=-0.0
Y[9,15]=-0.0
Y[9,16]=-0.0
Y[9,17]=-0.0
Y[9,18]=-0.0
Y[9,19]=1.0
Y[9,20]=0.0
Y[10,1]=-0.0
Y[10,2]=-0.0
Y[10,3]=-0.0
Y[10,4]=-0.0
Y[10,5]=-0.0
Y[10,6]=-0.0
Y[10,7]=-0.0
Y[10,8]=-0.0
Y[10,9]=-0.0
Y[10,10]=-0.0
Y[10,11]=-0.0
Y[10,12]=1.0
Y[10,13]=-0.0
Y[10,14]=-0.0
Y[10,15]=-0.0
Y[10,16]=1.0
Y[10,17]=-0.0
Y[10,18]=-0.0
Y[10,19]=-0.0
Y[10,20]=-0.0
```

In [22]:

```python
arcos_activos=[k for k in arcos if Y[k].x>0.99]
```

In [23]:

```python
model_FLP_b.ObjVal
```

Out[23]:

```
1137338.1398864486
```

In [24]:

```python
model_FLP_b.Runtime
```

Out[24]:

```
0.051860809326171875
```

In [25]:

```python
model_FLP_b_relaxed = model_FLP_b.relax()
model_FLP_b_relaxed.optimize()
```

```
Gurobi Optimizer version 9.1.2 build v9.1.2rc0 (win64)
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads
Optimize a model with 41 rows, 210 columns and 630 nonzeros
Model fingerprint: 0x9405ff5c
Coefficient statistics:
  Matrix range     [1e+00, 8e+02]
  Objective range  [2e+03, 1e+05]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Presolve time: 0.01s
Presolved: 41 rows, 210 columns, 630 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    7.9767608e+05   2.273438e+02   0.000000e+00      0s
      39    1.1076967e+06   0.000000e+00   0.000000e+00      0s

Solved in 39 iterations and 0.01 seconds
Optimal objective  1.107696670e+06
```

In [26]:

```python
for v in model_FLP_b_relaxed.getVars():
    print(str(v.VarName)+'='+str(round(v.x,2)))
```

```
X[1]=1.0
X[2]=1.0
X[3]=1.0
X[4]=0.0
X[5]=1.0
X[6]=0.82
X[7]=1.0
X[8]=1.0
X[9]=1.0
X[10]=0.79
Y[1,1]=0.0
Y[1,2]=0.0
Y[1,3]=0.0
Y[1,4]=0.0
Y[1,5]=0.59
Y[1,6]=0.0
Y[1,7]=0.0
Y[1,8]=0.0
Y[1,9]=1.0
Y[1,10]=0.0
Y[1,11]=1.0
Y[1,12]=0.0
Y[1,13]=0.0
Y[1,14]=0.0
Y[1,15]=0.0
Y[1,16]=0.0
Y[1,17]=0.0
Y[1,18]=0.0
Y[1,19]=0.0
Y[1,20]=0.43
Y[2,1]=0.0
Y[2,2]=1.0
Y[2,3]=0.0
Y[2,4]=0.0
Y[2,5]=0.41
Y[2,6]=0.0
Y[2,7]=0.0
Y[2,8]=0.0
Y[2,9]=0.0
Y[2,10]=0.0
Y[2,11]=0.0
Y[2,12]=0.0
Y[2,13]=0.0
Y[2,14]=0.0
Y[2,15]=0.0
Y[2,16]=0.0
Y[2,17]=0.0
Y[2,18]=0.0
Y[2,19]=0.0
Y[2,20]=0.0
Y[3,1]=0.0
Y[3,2]=0.0
Y[3,3]=0.0
Y[3,4]=0.0
Y[3,5]=0.0
Y[3,6]=0.0
Y[3,7]=0.0
Y[3,8]=0.0
Y[3,9]=0.0
Y[3,10]=0.0
Y[3,11]=0.0
Y[3,12]=0.0
Y[3,13]=0.0
Y[3,14]=0.0
Y[3,15]=0.0
Y[3,16]=0.12
Y[3,17]=1.0
Y[3,18]=1.0
Y[3,19]=0.0
Y[3,20]=0.0
Y[4,1]=0.0
Y[4,2]=0.0
Y[4,3]=0.0
Y[4,4]=0.0
Y[4,5]=0.0
Y[4,6]=0.0
Y[4,7]=0.0
Y[4,8]=0.0
Y[4,9]=0.0
Y[4,10]=0.0
Y[4,11]=0.0
Y[4,12]=0.0
Y[4,13]=0.0
Y[4,14]=0.0
Y[4,15]=0.0
Y[4,16]=0.0
Y[4,17]=0.0
Y[4,18]=0.0
Y[4,19]=0.0
Y[4,20]=0.0
Y[5,1]=0.0
Y[5,2]=0.0
Y[5,3]=0.0
Y[5,4]=0.58
Y[5,5]=0.0
Y[5,6]=0.0
Y[5,7]=1.0
Y[5,8]=0.0
Y[5,9]=0.0
Y[5,10]=0.0
Y[5,11]=0.0
Y[5,12]=0.0
Y[5,13]=0.0
Y[5,14]=0.0
Y[5,15]=0.0
Y[5,16]=0.88
Y[5,17]=0.0
Y[5,18]=0.0
Y[5,19]=0.0
Y[5,20]=0.0
Y[6,1]=0.0
Y[6,2]=0.0
Y[6,3]=0.51
Y[6,4]=0.0
Y[6,5]=0.0
Y[6,6]=0.0
Y[6,7]=0.0
Y[6,8]=0.0
Y[6,9]=0.0
Y[6,10]=0.0
Y[6,11]=0.0
Y[6,12]=0.0
Y[6,13]=0.0
Y[6,14]=1.0
Y[6,15]=0.0
Y[6,16]=0.0
Y[6,17]=0.0
Y[6,18]=0.0
Y[6,19]=0.0
Y[6,20]=0.36
```

```
W[7,7]=2801.0
W[7,8]=356.0
W[7,9]=0.0
W[7,10]=0.0
W[8,8]=2019.0
W[8,9]=1138.0
W[8,10]=0.0
W[9,9]=3145.0
W[9,10]=12.0
W[10,10]=3157.0
Y[1]=1.0
Y[2]=0.57
Y[3]=0.72
Y[4]=0.46
Y[5]=0.47
Y[6]=0.62
Y[7]=0.62
Y[8]=0.85
Y[9]=0.73
Y[10]=1.0
I[0]=0.0
I[1]=1902.0
I[2]=664.0
I[3]=1472.0
I[4]=2517.0
I[5]=894.0
I[6]=1725.0
I[7]=356.0
I[8]=1138.0
I[9]=12.0
I[10]=0.0
```

In [56]:

```python
model_LZP_b_relaxed.ObjVal
```

Out[56]: 5775873.757469927

In [57]:

```python
model_LZP_b_relaxed.Runtime
```

Out[57]: 0.015949249267578125

In [58]:

```python
GAP = (model_LZP_b.ObjVal-model_LZP_b_relaxed.ObjVal)/model_LZP_b.ObjVal
round(GAP*100,2)
```

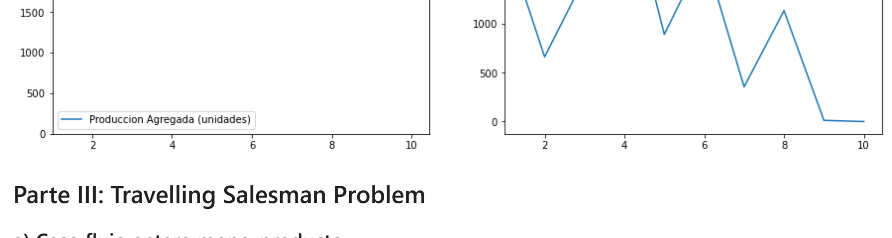
Out[58]: 0.08

Gráfico Solución

In [59]:

```python
fig = plt.figure(figsize=(15,5))

ax = fig.add_subplot(121)
ax.plot(T,produccion,label="Produccion Agregada (unidades)")
ax.set_title("Produccion $w_{ts}$")
ax.set_ylim(0.0)
ax.set_xlim(1)
ax.legend()

ax = fig.add_subplot(122)
ax.plot(T,inventario,label="Inventario (unidades)")
ax.set_title("Inventario $I_{t}$")
ax.set_xlim(1)
ax.legend()
```

Out[59]: <matplotlib.legend.Legend at 0x1e5db007220>



Parte III: Travelling Salesman Problem

a) Caso flujo entero mono-producto

In [60]:

```python
np.random.seed(1)
Nodos = 10
N = [n+1 for n in range(Nodos)]
arcos=[(i,j) for i in N for j in N if i!=j]

x_coor=list(np.random.random(len(N))*100)
y_coor=list(np.random.random(len(N))*100)

distancias={(i+1,j+1):np.hypot(x_coor[i]-x_coor[j],y_coor[i]-y_coor[j]) for i in range(len(N)) for j in range(l

C = distancias
IJ = [(i,j) for i in N for j in N if i!=j]
```

In [61]:

```python
model_TSP_a = Model('model_TSP_a')
#model_TSP_a.setParam('LogToConsole', 0)
```

In [62]:

```python
Y, F = None, None
```

In [63]:

```python
Y = model_TSP_a.addVars(IJ, vtype=GRB.BINARY, name='Y')
F = model_TSP_a.addVars(IJ, vtype=GRB.CONTINUOUS, name='F')
```

In [64]:

```python
objective = quicksum(C[(i,j)]*Y[(i,j)] for i,j in IJ)
model_TSP_a.setObjective(expr=objective, sense=GRB.MINIMIZE)
```

In [65]:

```python
model_TSP_a.addConstrs(quicksum(Y[(i,j)] for j in N if i!=j)==1 for i in N)
model_TSP_a.addConstrs(quicksum(Y[(j,i)] for j in N if i!=j)==1 for i in N)
model_TSP_a.addConstrs(F[(i,j)]<=(len(N)-1)*Y[(i,j)] for i,j in IJ)
model_TSP_a.addConstr(quicksum(F[(1,j)] for j in N if 1!=j)-quicksum(F[(j,1)] for j in N if 1!=j)==(len(N)-1))
model_TSP_a.addConstrs(quicksum(F[(i,j)] for j in N if i!=j)-quicksum(F[(j,i)] for j in N if i!=j)==-1 for i ir
model_TSP_a.update()
model_TSP_a.optimize()
```

```
Gurobi Optimizer version 9.1.2 build v9.1.2rc0 (win64)
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads
Optimize a model with 120 rows, 180 columns and 540 nonzeros
Model fingerprint: 0x0c422d62
Variable types: 90 continuous, 90 integer (90 binary)
Coefficient statistics:
  Matrix range     [1e+00, 9e+00]
  Objective range  [2e+01, 9e+01]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 9e+00]
Presolve removed 19 rows and 18 columns
Presolve time: 0.00s
Presolved: 101 rows, 162 columns, 477 nonzeros
Variable types: 72 continuous, 90 integer (90 binary)
Found heuristic solution: objective 590.5987440
Found heuristic solution: objective 482.8126201

Root relaxation: objective 2.647399e+02, 109 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0  264.73991    0   18  482.81262  264.73991  45.2%     -    0s
H    0     0                     333.8455502  264.73991  20.7%     -    0s
H    0     0                     277.0761938  265.47255  4.19%     -    0s
     0     0          -    0       277.07619  277.07619  0.00%     -    0s

Cutting planes:
  Gomory: 3
  Implied bound: 8
  Clique: 1
  MIR: 10
  Flow cover: 16
  Network: 6
  RLT: 3
  Relax-and-lift: 5

Explored 1 nodes (128 simplex iterations) in 0.04 seconds
Thread count was 8 (of 8 available processors)

Solution count 4: 277.076 333.846 482.813 590.599

Optimal solution found (tolerance 1.00e-04)
Best objective 2.770761938420e+02, best bound 2.770761938420e+02, gap 0.0000%
```

In [66]:

```python
for v in model_TSP_a.getVars():
    print(str(v.VarName)+'='+str(round(v.x,2)))
```

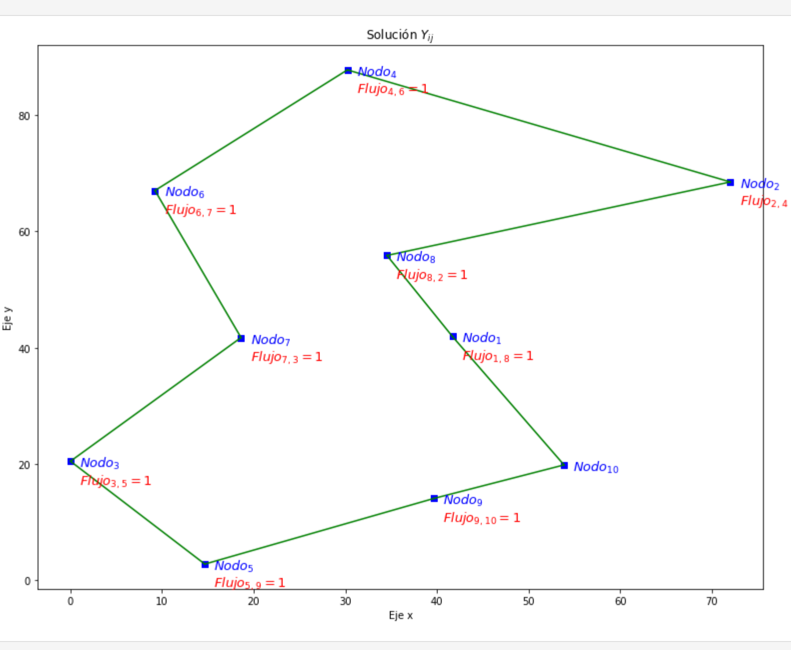
```
Y[1,2]=-0.0
Y[1,3]=-0.0
Y[1,4]=-0.0
Y[1,5]=-0.0
Y[1,6]=-0.0
Y[1,7]=-0.0
Y[1,8]=1.0
Y[1,9]=-0.0
Y[1,10]=0.0
Y[2,1]=0.0
Y[2,3]=-0.0
Y[2,4]=1.0
Y[2,5]=-0.0
Y[2,6]=-0.0
Y[2,7]=-0.0
Y[2,8]=0.0
Y[2,9]=-0.0
Y[2,10]=-0.0
Y[3,1]=-0.0
Y[3,2]=-0.0
Y[3,4]=-0.0
Y[3,5]=1.0
Y[3,6]=-0.0
Y[3,7]=0.0
Y[3,8]=-0.0
Y[3,9]=-0.0
Y[3,10]=-0.0
Y[4,1]=-0.0
Y[4,2]=0.0
Y[4,3]=-0.0
Y[4,5]=-0.0
Y[4,6]=1.0
Y[4,7]=-0.0
Y[4,8]=-0.0
Y[4,9]=-0.0
Y[4,10]=-0.0
Y[5,1]=-0.0
Y[5,2]=-0.0
Y[5,3]=0.0
Y[5,4]=-0.0
Y[5,6]=-0.0
Y[5,7]=-0.0
Y[5,8]=-0.0
Y[5,9]=1.0
Y[5,10]=-0.0
Y[6,1]=-0.0
Y[6,2]=-0.0
Y[6,3]=-0.0
Y[6,4]=0.0
Y[6,5]=-0.0
Y[6,7]=1.0
Y[6,8]=-0.0
Y[6,9]=-0.0
Y[6,10]=-0.0
Y[7,1]=-0.0
Y[7,2]=-0.0
Y[7,3]=1.0
Y[7,4]=-0.0
Y[7,5]=-0.0
Y[7,6]=0.0
Y[7,8]=-0.0
Y[7,9]=-0.0
Y[7,10]=-0.0
Y[8,1]=0.0
Y[8,2]=1.0
Y[8,3]=-0.0
Y[8,4]=-0.0
Y[8,5]=-0.0
Y[8,6]=-0.0
Y[8,7]=-0.0
Y[8,9]=-0.0
Y[8,10]=-0.0
Y[9,1]=0.0
Y[9,2]=-0.0
Y[9,3]=-0.0
Y[9,4]=-0.0
Y[9,5]=0.0
Y[9,6]=-0.0
Y[9,7]=-0.0
Y[9,8]=-0.0
Y[9,10]=1.0
Y[10,1]=1.0
Y[10,2]=-0.0
Y[10,3]=-0.0
Y[10,4]=-0.0
Y[10,5]=-0.0
Y[10,6]=-0.0
Y[10,7]=-0.0
Y[10,8]=-0.0
Y[10,9]=0.0
F[1,2]=0.0
F[1,3]=0.0
F[1,4]=0.0
F[1,5]=0.0
F[1,6]=0.0
F[1,7]=0.0
F[1,8]=9.0
F[1,9]=0.0
F[1,10]=0.0
F[2,1]=0.0
F[2,3]=0.0
F[2,4]=7.0
F[2,5]=0.0
F[2,6]=0.0
F[2,7]=0.0
F[2,8]=0.0
F[2,9]=0.0
F[2,10]=0.0
F[3,1]=0.0
F[3,2]=0.0
F[3,4]=0.0
F[3,5]=3.0
F[3,6]=0.0
F[3,7]=0.0
F[3,8]=0.0
F[3,9]=0.0
F[3,10]=0.0
```

```
Y[1,2]=0.0
Y[1,3]=-0.0
Y[1,4]=-0.0
Y[1,5]=-0.0
Y[1,6]=-0.0
Y[1,7]=-0.0
Y[1,8]=1.0
Y[1,9]=0.0
Y[1,10]=0.0
Y[2,1]=-0.0
Y[2,3]=-0.0
Y[2,4]=1.0
Y[2,5]=-0.0
Y[2,6]=-0.0
Y[2,7]=-0.0
Y[2,8]=0.0
Y[2,9]=-0.0
Y[2,10]=-0.0
Y[3,1]=-0.0
Y[3,2]=-0.0
Y[3,4]=-0.0
Y[3,5]=1.0
Y[3,6]=-0.0
Y[3,7]=0.0
Y[3,8]=-0.0
Y[3,9]=-0.0
Y[3,10]=-0.0
Y[4,1]=-0.0
Y[4,2]=0.0
Y[4,3]=-0.0
Y[4,5]=-0.0
Y[4,6]=1.0
Y[4,7]=-0.0
Y[4,8]=-0.0
Y[4,9]=-0.0
Y[4,10]=-0.0
Y[5,1]=-0.0
Y[5,2]=-0.0
Y[5,3]=0.0
Y[5,4]=-0.0
Y[5,6]=-0.0
Y[5,7]=-0.0
Y[5,8]=-0.0
Y[5,9]=1.0
Y[5,10]=-0.0
Y[6,1]=-0.0
Y[6,2]=-0.0
Y[6,3]=-0.0
Y[6,4]=0.0
Y[6,5]=-0.0
Y[6,7]=1.0
Y[6,8]=-0.0
Y[6,9]=-0.0
Y[6,10]=-0.0
Y[7,1]=-0.0
Y[7,2]=-0.0
Y[7,3]=1.0
Y[7,4]=-0.0
Y[7,5]=-0.0
Y[7,6]=0.0
Y[7,8]=-0.0
Y[7,9]=-0.0
Y[7,10]=-0.0
Y[8,1]=0.0
Y[8,2]=1.0
Y[8,3]=-0.0
Y[8,4]=-0.0
Y[8,5]=-0.0
Y[8,6]=-0.0
Y[8,7]=0.0
Y[8,9]=-0.0
Y[8,10]=-0.0
Y[9,1]=-0.0
Y[9,2]=-0.0
Y[9,3]=-0.0
Y[9,4]=-0.0
Y[9,5]=0.0
Y[9,6]=-0.0
Y[9,7]=-0.0
Y[9,8]=-0.0
Y[9,10]=1.0
Y[10,1]=1.0
Y[10,2]=-0.0
Y[10,3]=-0.0
Y[10,4]=-0.0
Y[10,5]=-0.0
Y[10,6]=-0.0
Y[10,7]=-0.0
Y[10,8]=-0.0
Y[10,9]=0.0
F[1,2,1]=0.0
F[1,2,2]=0.0
F[1,2,3]=0.0
F[1,2,4]=0.0
F[1,2,5]=0.0
F[1,2,6]=0.0
F[1,2,7]=0.0
F[1,2,8]=0.0
F[1,2,9]=0.0
F[1,2,10]=0.0
F[1,3,1]=0.0
F[1,3,2]=0.0
F[1,3,3]=0.0
F[1,3,4]=0.0
F[1,3,5]=0.0
F[1,3,6]=0.0
F[1,3,7]=0.0
F[1,3,8]=0.0
F[1,3,9]=0.0
F[1,3,10]=0.0
F[1,4,1]=0.0
F[1,4,2]=0.0
F[1,4,3]=0.0
F[1,4,4]=0.0
F[1,4,5]=0.0
F[1,4,6]=0.0
F[1,4,7]=0.0
F[1,4,8]=0.0
F[1,4,9]=0.0
F[1,4,10]=0.0
F[1,5,1]=0.0
F[1,5,2]=0.0
F[1,5,3]=0.0
F[1,5,4]=0.0
F[1,5,5]=0.0
F[1,5,6]=0.0
F[1,5,7]=0.0
F[1,5,8]=0.0
F[1,5,9]=0.0
F[1,5,10]=0.0
F[1,6,1]=0.0
F[1,6,2]=0.0
F[1,6,3]=0.0
F[1,6,4]=0.0
F[1,6,5]=0.0
F[1,6,6]=0.0
F[1,6,7]=0.0
F[1,6,8]=0.0
F[1,6,9]=0.0
F[1,6,10]=0.0
F[1,7,1]=0.0
F[1,7,2]=0.0
F[1,7,3]=0.0
F[1,7,4]=0.0
F[1,7,5]=0.0
F[1,7,6]=0.0
F[1,7,7]=0.0
F[1,7,8]=0.0
F[1,7,9]=0.0
F[1,7,10]=0.0
F[1,8,1]=0.0
F[1,8,2]=1.0
F[1,8,3]=1.0
F[1,8,4]=1.0
F[1,8,5]=1.0
F[1,8,6]=1.0
F[1,8,7]=1.0
F[1,8,8]=1.0
F[1,8,9]=1.0
F[1,8,10]=1.0
F[1,9,1]=0.0
F[1,9,2]=0.0
F[1,9,3]=0.0
F[1,9,4]=0.0
F[1,9,5]=0.0
F[1,9,6]=0.0
F[1,9,7]=0.0
F[1,9,8]=0.0
F[1,9,9]=0.0
F[1,9,10]=0.0
F[1,10,1]=0.0
F[1,10,2]=0.0
F[1,10,3]=0.0
F[1,10,4]=0.0
F[1,10,5]=0.0
F[1,10,6]=0.0
F[1,10,7]=0.0
F[1,10,8]=0.0
F[1,10,9]=0.0
F[1,10,10]=0.0
F[2,1,1]=0.0
F[2,1,2]=0.0
F[2,1,3]=0.0
F[2,1,4]=0.0
F[2,1,5]=0.0
F[2,1,6]=0.0
F[2,1,7]=0.0
F[2,1,8]=0.0
F[2,1,9]=0.0
F[2,1,10]=0.0
F[2,3,1]=0.0
F[2,3,2]=0.0
F[2,3,3]=0.0
F[2,3,4]=0.0
F[2,3,5]=0.0
F[2,3,6]=0.0
F[2,3,7]=0.0
F[2,3,8]=0.0
F[2,3,9]=0.0
F[2,3,10]=0.0
F[2,4,1]=0.0
F[2,4,2]=0.0
F[2,4,3]=1.0
F[2,4,4]=1.0
F[2,4,5]=1.0
F[2,4,6]=1.0
F[2,4,7]=1.0
F[2,4,8]=0.0
F[2,4,9]=1.0
F[2,4,10]=1.0
F[2,5,1]=0.0
F[2,5,2]=0.0
F[2,5,3]=0.0
F[2,5,4]=0.0
F[2,5,5]=0.0
F[2,5,6]=0.0
F[2,5,7]=0.0
F[2,5,8]=0.0
F[2,5,9]=0.0
F[2,5,10]=0.0
F[2,6,1]=0.0
F[2,6,2]=0.0
F[2,6,3]=0.0
F[2,6,4]=0.0
F[2,6,5]=0.0
F[2,6,6]=0.0
F[2,6,7]=0.0
F[2,6,8]=0.0
F[2,6,9]=0.0
F[2,6,10]=0.0
F[2,7,1]=0.0
F[2,7,2]=0.0
F[2,7,3]=0.0
F[2,7,4]=0.0
F[2,7,5]=0.0
F[2,7,6]=0.0
F[2,7,7]=0.0
F[2,7,8]=0.0
F[2,7,9]=0.0
F[2,7,10]=0.0
F[2,8,1]=0.0
F[2,8,2]=0.0
F[2,8,3]=0.0
F[2,8,4]=0.0
F[2,8,5]=0.0
F[2,8,6]=0.0
F[2,8,7]=0.0
F[2,8,8]=0.0
F[2,8,9]=0.0
F[2,8,10]=0.0
F[2,9,1]=0.0
F[2,9,2]=0.0
F[2,9,3]=0.0
F[2,9,4]=0.0
F[2,9,5]=0.0
F[2,9,6]=0.0
F[2,9,7]=0.0
F[2,9,8]=0.0
F[2,9,9]=0.0
F[2,9,10]=0.0
F[2,10,1]=0.0
F[2,10,2]=0.0
F[2,10,3]=0.0
F[2,10,4]=0.0
F[2,10,5]=0.0
F[2,10,6]=0.0
F[2,10,7]=0.0
F[2,10,8]=0.0
F[2,10,9]=0.0
F[2,10,10]=0.0
F[3,1,1]=0.0
F[3,1,2]=0.0
F[3,1,3]=0.0
F[3,1,4]=0.0
F[3,1,5]=0.0
F[3,1,6]=0.0
F[3,1,7]=0.0
F[3,1,8]=0.0
F[3,1,9]=0.0
F[3,1,10]=0.0
F[3,2,1]=0.0
F[3,2,2]=0.0
F[3,2,3]=0.0
F[3,2,4]=0.0
F[3,2,5]=0.0
F[3,2,6]=0.0
F[3,2,7]=0.0
F[3,2,8]=0.0
F[3,2,9]=0.0
F[3,2,10]=0.0
F[3,4,1]=0.0
F[3,4,2]=0.0
F[3,4,3]=0.0
F[3,4,4]=0.0
F[3,4,5]=0.0
F[3,4,6]=0.0
F[3,4,7]=0.0
F[3,4,8]=0.0
F[3,4,9]=0.0
F[3,4,10]=0.0
F[3,5,1]=0.0
F[3,5,2]=0.0
F[3,5,3]=0.0
F[3,5,4]=0.0
F[3,5,5]=1.0
F[3,5,6]=0.0
F[3,5,7]=0.0
F[3,5,8]=0.0
F[3,5,9]=1.0
F[3,5,10]=1.0
F[3,6,1]=0.0
F[3,6,2]=0.0
F[3,6,3]=0.0
F[3,6,4]=0.0
F[3,6,5]=0.0
F[3,6,6]=0.0
F[3,6,7]=0.0
F[3,6,8]=0.0
F[3,6,9]=0.0
F[3,6,10]=0.0
F[3,7,1]=0.0
F[3,7,2]=0.0
F[3,7,3]=0.0
F[3,7,4]=0.0
F[3,7,5]=0.0
F[3,7,6]=0.0
F[3,7,7]=0.0
F[3,7,8]=0.0
F[3,7,9]=0.0
F[3,7,10]=0.0
F[3,8,1]=0.0
F[3,8,2]=0.0
F[3,8,3]=0.0
F[3,8,4]=0.0
F[3,8,5]=0.0
F[3,8,6]=0.0
F[3,8,7]=0.0
F[3,8,8]=0.0
F[3,8,9]=0.0
F[3,8,10]=0.0
F[3,9,1]=0.0
F[3,9,2]=0.0
F[3,9,3]=0.0
F[3,9,4]=0.0
F[3,9,5]=0.0
F[3,9,6]=0.0
F[3,9,7]=0.0
F[3,9,8]=0.0
F[3,9,9]=0.0
F[3,9,10]=0.0
F[3,10,1]=0.0
F[3,10,2]=0.0
F[3,10,3]=0.0
F[3,10,4]=0.0
F[3,10,5]=0.0
F[3,10,6]=0.0
F[3,10,7]=0.0
F[3,10,8]=0.0
F[3,10,9]=0.0
F[3,10,10]=0.0
F[4,1,1]=0.0
F[4,1,2]=0.0
F[4,1,3]=0.0
F[4,1,4]=0.0
F[4,1,5]=0.0
F[4,1,6]=0.0
F[4,1,7]=0.0
F[4,1,8]=0.0
F[4,1,9]=0.0
F[4,1,10]=0.0
F[4,2,1]=0.0
F[4,2,2]=0.0
F[4,2,3]=0.0
F[4,2,4]=0.0
F[4,2,5]=0.0
F[4,2,6]=0.0
F[4,2,7]=0.0
F[4,2,8]=0.0
F[4,2,9]=0.0
F[4,2,10]=0.0
F[4,3,1]=0.0
F[4,3,2]=0.0
F[4,3,3]=0.0
F[4,3,4]=0.0
F[4,3,5]=0.0
F[4,3,6]=0.0
F[4,3,7]=0.0
F[4,3,8]=0.0
F[4,3,9]=0.0
F[4,3,10]=0.0
F[4,5,1]=0.0
F[4,5,2]=0.0
F[4,5,3]=0.0
F[4,5,4]=0.0
F[4,5,5]=0.0
F[4,5,6]=0.0
F[4,5,7]=0.0
F[4,5,8]=0.0
F[4,5,9]=0.0
F[4,5,10]=0.0
F[4,6,1]=0.0
F[4,6,2]=0.0
F[4,6,3]=1.0
F[4,6,4]=0.0
F[4,6,5]=1.0
F[4,6,6]=1.0
F[4,6,7]=1.0
F[4,6,8]=0.0
F[4,6,9]=1.0
F[4,6,10]=1.0
F[4,7,1]=0.0
F[4,7,2]=0.0
F[4,7,3]=0.0
F[4,7,4]=0.0
F[4,7,5]=0.0
F[4,7,6]=0.0
F[4,7,7]=0.0
F[4,7,8]=0.0
F[4,7,9]=0.0
F[4,7,10]=0.0
F[4,8,1]=0.0
F[4,8,2]=0.0
F[4,8,3]=0.0
F[4,8,4]=0.0
F[4,8,5]=0.0
F[4,8,6]=0.0
F[4,8,7]=0.0
F[4,8,8]=0.0
F[4,8,9]=0.0
F[4,8,10]=0.0
F[4,9,1]=0.0
F[4,9,2]=0.0
F[4,9,3]=0.0
F[4,9,4]=0.0
F[4,9,5]=0.0
F[4,9,6]=0.0
F[4,9,7]=0.0
F[4,9,8]=0.0
F[4,9,9]=0.0
F[4,9,10]=0.0
F[4,10,1]=0.0
F[4,10,2]=0.0
F[4,10,3]=0.0
F[4,10,4]=0.0
F[4,10,5]=0.0
F[4,10,6]=0.0
F[4,10,7]=0.0
F[4,10,8]=0.0
F[4,10,9]=0.0
F[4,10,10]=0.0
F[5,1,1]=0.0
F[5,1,2]=0.0
F[5,1,3]=0.0
F[5,1,4]=0.0
F[5,1,5]=0.0
F[5,1,6]=0.0
F[5,1,7]=0.0
F[5,1,8]=0.0
F[5,1,9]=0.0
F[5,1,10]=0.0
F[5,2,1]=0.0
F[5,2,2]=0.0
F[5,2,3]=0.0
F[5,2,4]=0.0
F[5,2,5]=0.0
F[5,2,6]=0.0
F[5,2,7]=0.0
F[5,2,8]=0.0
F[5,2,9]=0.0
F[5,2,10]=0.0
F[5,3,1]=0.0
F[5,3,2]=0.0
F[5,3,3]=0.0
F[5,3,4]=0.0
F[5,3,5]=0.0
F[5,3,6]=0.0
F[5,3,7]=0.0
F[5,3,8]=0.0
F[5,3,9]=0.0
F[5,3,10]=0.0
F[5,4,1]=0.0
F[5,4,2]=0.0
F[5,4,3]=0.0
F[5,4,4]=0.0
F[5,4,5]=0.0
F[5,4,6]=0.0
F[5,4,7]=0.0
F[5,4,8]=0.0
F[5,4,9]=0.0
F[5,4,10]=0.0
F[5,6,1]=0.0
F[5,6,2]=0.0
F[5,6,3]=0.0
F[5,6,4]=0.0
F[5,6,5]=0.0
F[5,6,6]=0.0
F[5,6,7]=0.0
F[5,6,8]=0.0
F[5,6,9]=0.0
F[5,6,10]=0.0
F[5,7,1]=0.0
F[5,7,2]=0.0
F[5,7,3]=0.0
F[5,7,4]=0.0
F[5,7,5]=0.0
F[5,7,6]=0.0
F[5,7,7]=0.0
F[5,7,8]=0.0
F[5,7,9]=0.0
F[5,7,10]=0.0
F[5,8,1]=0.0
F[5,8,2]=0.0
F[5,8,3]=0.0
F[5,8,4]=0.0
F[5,8,5]=0.0
F[5,8,6]=0.0
F[5,8,7]=0.0
F[5,8,8]=0.0
F[5,8,9]=0.0
F[5,8,10]=0.0
F[5,9,1]=0.0
F[5,9,2]=0.0
F[5,9,3]=0.0
F[5,9,4]=0.0
F[5,9,5]=0.0
F[5,9,6]=0.0
F[5,9,7]=0.0
F[5,9,8]=0.0
F[5,9,9]=1.0
F[5,9,10]=1.0
F[5,10,1]=0.0
F[5,10,2]=0.0
F[5,10,3]=0.0
F[5,10,4]=0.0
F[5,10,5]=0.0
F[5,10,6]=0.0
F[5,10,7]=0.0
F[5,10,8]=0.0
F[5,10,9]=0.0
F[5,10,10]=0.0
F[6,1,1]=0.0
F[6,1,2]=0.0
F[6,1,3]=0.0
F[6,1,4]=0.0
F[6,1,5]=0.0
F[6,1,6]=0.0
F[6,1,7]=0.0
F[6,1,8]=0.0
F[6,1,9]=0.0
F[6,1,10]=0.0
F[6,2,1]=0.0
F[6,2,2]=0.0
F[6,2,3]=0.0
F[6,2,4]=0.0
F[6,2,5]=0.0
F[6,2,6]=0.0
F[6,2,7]=0.0
F[6,2,8]=0.0
F[6,2,9]=0.0
F[6,2,10]=0.0
F[6,3,1]=0.0
F[6,3,2]=0.0
F[6,3,3]=0.0
F[6,3,4]=0.0
F[6,3,5]=0.0
F[6,3,6]=0.0
F[6,3,7]=0.0
F[6,3,8]=0.0
F[6,3,9]=0.0
F[6,3,10]=0.0
F[6,4,1]=0.0
F[6,4,2]=0.0
F[6,4,3]=0.0
F[6,4,4]=0.0
F[6,4,5]=0.0
F[6,4,6]=0.0
F[6,4,7]=0.0
F[6,4,8]=0.0
F[6,4,9]=0.0
F[6,4,10]=0.0
F[6,5,1]=0.0
F[6,5,2]=0.0
F[6,5,3]=0.0
F[6,5,4]=0.0
F[6,5,5]=0.0
F[6,5,6]=0.0
F[6,5,7]=0.0
F[6,5,8]=0.0
F[6,5,9]=0.0
F[6,5,10]=0.0
F[6,7,1]=0.0
F[6,7,2]=0.0
F[6,7,3]=1.0
F[6,7,4]=0.0
F[6,7,5]=1.0
F[6,7,6]=0.0
F[6,7,7]=1.0
F[6,7,8]=0.0
F[6,7,9]=1.0
F[6,7,10]=1.0
F[6,8,1]=0.0
F[6,8,2]=0.0
F[6,8,3]=0.0
F[6,8,4]=0.0
F[6,8,5]=0.0
F[6,8,6]=0.0
F[6,8,7]=0.0
F[6,8,8]=0.0
F[6,8,9]=0.0
F[6,8,10]=0.0
F[6,9,1]=0.0
F[6,9,2]=0.0
F[6,9,3]=0.0
F[6,9,4]=0.0
F[6,9,5]=0.0
F[6,9,6]=0.0
F[6,9,7]=0.0
F[6,9,8]=0.0
F[6,9,9]=0.0
F[6,9,10]=0.0
F[6,10,1]=0.0
F[6,10,2]=0.0
F[6,10,3]=0.0
F[6,10,4]=0.0
F[6,10,5]=0.0
F[6,10,6]=0.0
F[6,10,7]=0.0
F[6,10,8]=0.0
F[6,10,9]=0.0
F[6,10,10]=0.0
F[7,1,1]=0.0
F[7,1,2]=0.0
F[7,1,3]=0.0
F[7,1,4]=0.0
F[7,1,5]=0.0
F[7,1,6]=0.0
F[7,1,7]=0.0
F[7,1,8]=0.0
F[7,1,9]=0.0
F[7,1,10]=0.0
F[7,2,1]=0.0
F[7,2,2]=0.0
F[7,2,3]=0.0
F[7,2,4]=0.0
F[7,2,5]=0.0
F[7,2,6]=0.0
F[7,2,7]=0.0
F[7,2,8]=0.0
F[7,2,9]=0.0
F[7,2,10]=0.0
F[7,3,1]=0.0
F[7,3,2]=0.0
F[7,3,3]=1.0
F[7,3,4]=0.0
F[7,3,5]=1.0
F[7,3,6]=0.0
F[7,3,7]=0.0
F[7,3,8]=0.0
F[7,3,9]=1.0
F[7,3,10]=1.0
F[7,4,1]=0.0
F[7,4,2]=0.0
F[7,4,3]=0.0
F[7,4,4]=0.0
F[7,4,5]=0.0
F[7,4,6]=0.0
F[7,4,7]=0.0
F[7,4,8]=0.0
F[7,4,9]=0.0
F[7,4,10]=0.0
F[7,5,1]=0.0
F[7,5,2]=0.0
F[7,5,3]=0.0
F[7,5,4]=0.0
F[7,5,5]=0.0
F[7,5,6]=0.0
F[7,5,7]=0.0
F[7,5,8]=0.0
F[7,5,9]=0.0
F[7,5,10]=0.0
F[7,6,1]=0.0
F[7,6,2]=0.0
F[7,6,3]=0.0
F[7,6,4]=0.0
F[7,6,5]=0.0
F[7,6,6]=0.0
F[7,6,7]=0.0
F[7,6,8]=0.0
F[7,6,9]=0.0
F[7,6,10]=0.0
F[7,8,1]=0.0
F[7,8,2]=0.0
F[7,8,3]=0.0
F[7,8,4]=0.0
F[7,8,5]=0.0
F[7,8,6]=0.0
F[7,8,7]=0.0
F[7,8,8]=0.0
F[7,8,9]=0.0
F[7,8,10]=0.0
F[7,9,1]=0.0
F[7,9,2]=0.0
F[7,9,3]=0.0
F[7,9,4]=0.0
F[7,9,5]=0.0
F[7,9,6]=0.0
F[7,9,7]=0.0
F[7,9,8]=0.0
F[7,9,9]=0.0
F[7,9,10]=0.0
F[7,10,1]=0.0
F[7,10,2]=0.0
F[7,10,3]=0.0
F[7,10,4]=0.0
F[7,10,5]=0.0
F[7,10,6]=0.0
F[7,10,7]=0.0
F[7,10,8]=0.0
F[7,10,9]=0.0
F[7,10,10]=0.0
F[8,1,1]=0.0
F[8,1,2]=0.0
F[8,1,3]=0.0
F[8,1,4]=0.0
F[8,1,5]=0.0
F[8,1,6]=0.0
F[8,1,7]=0.0
F[8,1,8]=0.0
F[8,1,9]=0.0
F[8,1,10]=0.0
F[8,2,1]=0.0
F[8,2,2]=1.0
F[8,2,3]=1.0
F[8,2,4]=1.0
F[8,2,5]=1.0
F[8,2,6]=1.0
F[8,2,7]=1.0
F[8,2,8]=0.0
F[8,2,9]=1.0
F[8,2,10]=1.0
F[8,3,1]=0.0
F[8,3,2]=0.0
F[8,3,3]=0.0
F[8,3,4]=0.0
F[8,3,5]=0.0
F[8,3,6]=0.0
F[8,3,7]=0.0
F[8,3,8]=0.0
F[8,3,9]=0.0
F[8,3,10]=0.0
F[8,4,1]=0.0
F[8,4,2]=0.0
F[8,4,3]=0.0
F[8,4,4]=0.0
F[8,4,5]=0.0
F[8,4,6]=0.0
F[8,4,7]=0.0
F[8,4,8]=0.0
F[8,4,9]=0.0
F[8,4,10]=0.0
F[8,5,1]=0.0
F[8,5,2]=0.0
F[8,5,3]=0.0
F[8,5,4]=0.0
F[8,5,5]=0.0
F[8,5,6]=0.0
F[8,5,7]=0.0
F[8,5,8]=0.0
F[8,5,9]=0.0
F[8,5,10]=0.0
F[8,6,1]=0.0
F[8,6,2]=0.0
F[8,6,3]=0.0
F[8,6,4]=0.0
F[8,6,5]=0.0
F[8,6,6]=0.0
F[8,6,7]=0.0
F[8,6,8]=0.0
F[8,6,9]=0.0
F[8,6,10]=0.0
F[8,7,1]=0.0
F[8,7,2]=0.0
F[8,7,3]=0.0
F[8,7,4]=0.0
F[8,7,5]=0.0
F[8,7,6]=0.0
F[8,7,7]=0.0
F[8,7,8]=0.0
F[8,7,9]=0.0
F[8,7,10]=0.0
F[8,9,1]=0.0
F[8,9,2]=0.0
F[8,9,3]=0.0
F[8,9,4]=0.0
F[8,9,5]=0.0
F[8,9,6]=0.0
F[8,9,7]=0.0
F[8,9,8]=0.0
F[8,9,9]=0.0
F[8,9,10]=0.0
F[8,10,1]=0.0
F[8,10,2]=0.0
F[8,10,3]=0.0
F[8,10,4]=0.0
F[8,10,5]=0.0
F[8,10,6]=0.0
F[8,10,7]=0.0
F[8,10,8]=0.0
F[8,10,9]=0.0
F[8,10,10]=0.0
F[9,1,1]=0.0
F[9,1,2]=0.0
F[9,1,3]=0.0
F[9,1,4]=0.0
F[9,1,5]=0.0
F[9,1,6]=0.0
F[9,1,7]=0.0
F[9,1,8]=0.0
F[9,1,9]=0.0
F[9,1,10]=0.0
F[9,2,1]=0.0
F[9,2,2]=0.0
F[9,2,3]=0.0
F[9,2,4]=0.0
F[9,2,5]=0.0
F[9,2,6]=0.0
F[9,2,7]=0.0
F[9,2,8]=0.0
F[9,2,9]=0.0
F[9,2,10]=0.0
F[9,3,1]=0.0
F[9,3,2]=0.0
F[9,3,3]=0.0
F[9,3,4]=0.0
F[9,3,5]=0.0
F[9,3,6]=0.0
F[9,3,7]=0.0
F[9,3,8]=0.0
F[9,3,9]=0.0
F[9,3,10]=0.0
F[9,4,1]=0.0
F[9,4,2]=0.0
F[9,4,3]=0.0
F[9,4,4]=0.0
F[9,4,5]=0.0
F[9,4,6]=0.0
F[9,4,7]=0.0
F[9,4,8]=0.0
F[9,4,9]=0.0
F[9,4,10]=0.0
F[9,5,1]=0.0
F[9,5,2]=0.0
F[9,5,3]=0.0
F[9,5,4]=0.0
F[9,5,5]=0.0
F[9,5,6]=0.0
F[9,5,7]=0.0
F[9,5,8]=0.0
F[9,5,9]=0.0
F[9,5,10]=0.0
F[9,6,1]=0.0
F[9,6,2]=0.0
F[9,6,3]=0.0
F[9,6,4]=0.0
F[9,6,5]=0.0
F[9,6,6]=0.0
F[9,6,7]=0.0
F[9,6,8]=0.0
F[9,6,9]=0.0
F[9,6,10]=0.0
F[9,7,1]=0.0
F[9,7,2]=0.0
F[9,7,3]=0.0
F[9,7,4]=0.0
F[9,7,5]=0.0
F[9,7,6]=0.0
F[9,7,7]=0.0
F[9,7,8]=0.0
F[9,7,9]=0.0
F[9,7,10]=0.0
F[9,8,1]=0.0
F[9,8,2]=0.0
F[9,8,3]=0.0
F[9,8,4]=0.0
F[9,8,5]=0.0
F[9,8,6]=0.0
F[9,8,7]=0.0
F[9,8,8]=0.0
F[9,8,9]=0.0
F[9,8,10]=0.0
F[9,10,1]=0.0
F[9,10,2]=0.0
F[9,10,3]=0.0
F[9,10,4]=0.0
F[9,10,5]=0.0
F[9,10,6]=0.0
F[9,10,7]=0.0
F[9,10,8]=0.0
F[9,10,9]=0.0
F[9,10,10]=1.0
F[10,1,1]=0.0
F[10,1,2]=0.0
F[10,1,3]=0.0
F[10,1,4]=0.0
F[10,1,5]=0.0
F[10,1,6]=0.0
F[10,1,7]=0.0
F[10,1,8]=0.0
F[10,1,9]=0.0
F[10,1,10]=0.0
F[10,2,1]=0.0
F[10,2,2]=0.0
F[10,2,3]=0.0
F[10,2,4]=0.0
F[10,2,5]=0.0
F[10,2,6]=0.0
F[10,2,7]=0.0
F[10,2,8]=0.0
F[10,2,9]=0.0
F[10,2,10]=0.0
F[10,3,1]=0.0
F[10,3,2]=0.0
F[10,3,3]=0.0
F[10,3,4]=0.0
F[10,3,5]=0.0
F[10,3,6]=0.0
F[10,3,7]=0.0
F[10,3,8]=0.0
F[10,3,9]=0.0
F[10,3,10]=0.0
F[10,4,1]=0.0
F[10,4,2]=0.0
F[10,4,3]=0.0
F[10,4,4]=0.0
F[10,4,5]=0.0
F[10,4,6]=0.0
F[10,4,7]=0.0
F[10,4,8]=0.0
F[10,4,9]=0.0
F[10,4,10]=0.0
F[10,5,1]=0.0
F[10,5,2]=0.0
F[10,5,3]=0.0
F[10,5,4]=0.0
F[10,5,5]=0.0
F[10,5,6]=0.0
F[10,5,7]=0.0
F[10,5,8]=0.0
F[10,5,9]=0.0
F[10,5,10]=0.0
F[10,6,1]=0.0
F[10,6,2]=0.0
F[10,6,3]=0.0
F[10,6,4]=0.0
F[10,6,5]=0.0
F[10,6,6]=0.0
F[10,6,7]=0.0
F[10,6,8]=0.0
F[10,6,9]=0.0
F[10,6,10]=0.0
```

```
F[10,7,1]=0.0
F[10,7,2]=0.0
F[10,7,3]=0.0
F[10,7,4]=0.0
F[10,7,5]=0.0
F[10,7,6]=0.0
F[10,7,7]=0.0
F[10,7,8]=0.0
F[10,7,9]=0.0
F[10,7,10]=0.0
F[10,8,1]=0.0
F[10,8,2]=0.0
F[10,8,3]=0.0
F[10,8,4]=0.0
F[10,8,5]=0.0
F[10,8,6]=0.0
F[10,8,7]=0.0
F[10,8,8]=0.0
F[10,8,9]=0.0
F[10,8,10]=0.0
F[10,9,1]=0.0
F[10,9,2]=0.0
F[10,9,3]=0.0
F[10,9,4]=0.0
F[10,9,5]=0.0
F[10,9,6]=0.0
F[10,9,7]=0.0
F[10,9,8]=0.0
F[10,9,9]=0.0
F[10,9,10]=0.0
```

In [78]:
```
arcos_activos=[k for k in arcos if Y[k].x>0.99]
flujos=[k for k in IJK if F[k].x>0.99]
```

In [79]:
```
plt.figure(figsize=(13,10));
plt.scatter(x_coor,y_coor,color="blue",marker="s");

for i in range(len(N)):
    plt.annotate("$Nodo_{%d}$"%(i+1),(x_coor[i]+1,y_coor[i]-1),size=13, color="blue");

graph_arcos_activos = [True]*len(arcos_activos)

for n in arcos_activos:
    i=n[0]
    j=n[1]
    plt.plot([x_coor[i-1], x_coor[j-1]], [y_coor[i-1],y_coor[j-1]], color='green')

for n in IJK:
    i=n[0]
    j=n[1]
    k=n[2]
    if (i,j) in arcos_activos:
        pos = arcos_activos.index((i,j))
        if graph_arcos_activos[pos] and F[i,j,k].x>0.9:
            plt.annotate("$Flujo_{%d,%d}=%d$"%(i,j,F[i,j,k].x),(x_coor[i-1]+1,y_coor[i-1]-4),size=13, color="re
            graph_arcos_activos[pos] = False
plt.xlabel("Eje x");
plt.ylabel("Eje y");
plt.title("Solución $Y_{ij}$");
plt.show();
```



In [80]:
```
model_TSP_b.runtime
```

Out[80]: 0.04288482666015625

In [81]:
```
model_TSP_a.runtime
```

Out[81]: 0.04787254333496094

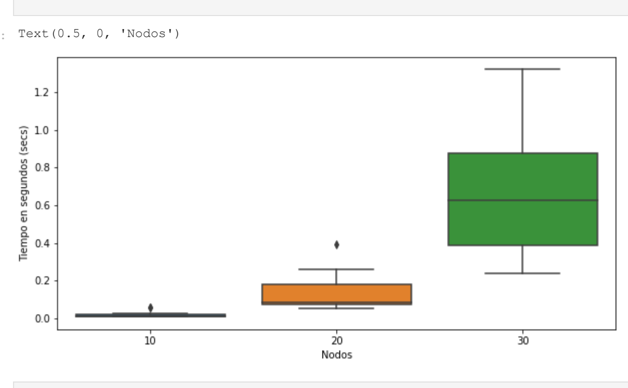
In [82]:
```
import seaborn as sns
```

In [83]:
```
instancias = 10
Nodos_list = [10,20,30]
experiment = 0
rows = []
for i in range(instancias):
    experiment+=1
    for Nodos in Nodos_list:
        #np.random.seed(1)
        N = [n+1 for n in range(Nodos)]
        arcos=[(i,j) for i in N for j in N if i!=j]

        x_coor=list(np.random.random(len(N))*100)
        y_coor=list(np.random.random(len(N))*100)

        distancias={(i+1,j+1):np.hypot(x_coor[i]-x_coor[j],y_coor[i]-y_coor[j]) for i in range(len(N)) for j in

        C = distancias
        IJ = [(i,j) for i in N for j in N if i!=j]
        #Model parte a
        model_TSP_a = Model('model_TSP_a')
        model_TSP_a.setParam('LogToConsole', 0)
        Y, F = None, None

        Y = model_TSP_a.addVars(IJ, vtype=GRB.BINARY, name='Y')
        F = model_TSP_a.addVars(IJ, vtype=GRB.CONTINUOUS, name='F')

        objective = quicksum(C[(i,j)]*Y[(i,j)] for i,j in IJ)
        model_TSP_a.setObjective(expr=objective, sense=GRB.MINIMIZE)

        model_TSP_a.addConstrs(quicksum(Y[(i,j)] for j in N if i!=j)==1 for i in N)
        model_TSP_a.addConstrs(quicksum(Y[(j,i)] for j in N if i!=j)==1 for i in N)
        model_TSP_a.addConstrs(F[(i,j)]<=(len(N)-1)*Y[(i,j)] for i,j in IJ)
        model_TSP_a.addConstr(quicksum(F[(1,j)] for j in N if 1!=j)-quicksum(F[(j,1)] for j in N if 1!=j)==(ler
        model_TSP_a.addConstrs(quicksum(F[(i,j)] for j in N if i!=j)-quicksum(F[(j,i)] for j in N if i!=j)==-1
        model_TSP_a.update()
        model_TSP_a.optimize()

        objective_value_a = model_TSP_a.ObjVal
        time_secs_a = model_TSP_a.runtime

        #Model parte b

        Y, F = None, None
        K = [k+1 for k in range(Nodos)]
        IJK = [(i,j,k) for i,j in IJ for k in K]

        model_TSP_b = Model('model_TSP_b')
        model_TSP_b.setParam('LogToConsole', 0)


        Y = model_TSP_b.addVars(IJ, vtype=GRB.BINARY, name='Y')
        F = model_TSP_b.addVars(IJK, vtype=GRB.CONTINUOUS, name='F')

        objective = quicksum(C[(i,j)]*Y[(i,j)] for i,j in IJ)
        model_TSP_b.setObjective(expr=objective, sense=GRB.MINIMIZE)

        model_TSP_b.addConstrs(quicksum(Y[(i,j)] for j in N if j!=i)==1 for i in N)
        model_TSP_b.addConstrs(quicksum(Y[(j,i)] for j in N if j!=i)==1 for i in N)
        model_TSP_b.addConstrs(F[(i,j,k)]<=Y[(i,j)] for i,j in IJ for k in K[1:])
        model_TSP_b.addConstrs(quicksum(F[(i,j,k)] for j in N if j!=i)-quicksum(F[(j,i,k)] for j in N if j!=i)=
        model_TSP_b.optimize()

        objective_value_b = model_TSP_b.ObjVal
        time_secs_b = model_TSP_b.runtime

        row = {
            'experiment':experiment,
            'nodos':Nodos,
            'objective_value_a':objective_value_a,
            'time_secs_a':time_secs_a,
            'objective_value_b':objective_value_b,
            'time_secs_b':time_secs_b
        }
        rows.append(row)
    print(experiment)
df = pd.DataFrame(rows)
```

```
1
2
3
4
5
6
7
8
9
10
```

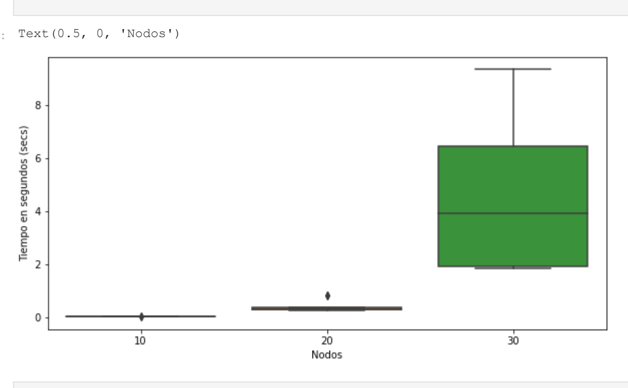
In [84]:
```
plt.figure(figsize=(10,5))
sns.boxplot(data=df, x='nodos', y = 'time_secs_a')
plt.ylabel('Tiempo en segundos (secs)')
plt.xlabel('Nodos')
```

Out[84]: Text(0.5, 0, 'Nodos')



In [85]:
```
plt.figure(figsize=(10,5))
sns.boxplot(data=df, x='nodos', y = 'time_secs_b')
plt.ylabel('Tiempo en segundos (secs)')
plt.xlabel('Nodos')
```

Out[85]: Text(0.5, 0, 'Nodos')



In [86]:
```
time_secs_a, time_secs_b = list(df.time_secs_a), list(df.time_secs_b)
TSP_a = ['A']*len(time_secs_a)
TSP_b = ['B']*len(time_secs_b)
TSP = TSP_a + TSP_b
time_secs = time_secs_a + time_secs_b
df2 = {'TSP':TSP, 'time_secs':time_secs}
df2 = pd.DataFrame(df2)
```

In [87]:
```
df2.head(2)
```

Out[87]:

	TSP	time_secs
0	A	0.013964
1	A	0.258282

In [88]:
```
plt.figure(figsize=(10,5))
sns.boxplot(data=df2, x='TSP', y = 'time_secs', palette = 'PuOr')
plt.ylabel('Tiempo en segundos (secs)')
plt.xlabel('Modelo')
```

Out[88]: Text(0.5, 0, 'Modelo')