Lista 1 - Exercício 3

INF2912 - Otimização Combinatória

Prof. Marcus Vinicius Soledade Poggi de Aragão

2015-2

Ciro Cavani

BigData / **Globo.com**

Facility Location Problem

- P - conjunto de centros de atendimento $|P| = p$
- J - conjunto de locais a serem atendidos $|J| = n$
- c_{ij} - custo fixo de atendimento do centro i para o ponto j

Objetivo:

custo total de atendimento mínimo

Modelo ULP

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \\
\text{subject to} \quad & \sum_{i=1}^{n} x_{ij} = 1 && j = 1, \ldots, n \\
& \sum_{i=1}^{n} y_i = p \\
& x_{ij} - y_i \leq 0 && i, j = 1, \ldots, n \\
& x_{ij} \geq 0 && i, j = 1, \ldots, n \\
& y_i \in \{0, 1\} && i = 1, \ldots, n
\end{aligned}
$$

JuMP

http://www.juliaopt.org/ (http://www.juliaopt.org/)

http://jump.readthedocs.org/en/stable/ (http://jump.readthedocs.org/en/stable/)

Modeling language for Mathematical Programming (linear, mixed-integer, conic, nonlinear)

In [1]:
```
using JuMP
```

In [2]:
```
p = 2
n = 3

c = [5 10 7
    8 9 6
    7 11 0]
```

Out[2]:
```
3x3 Array{Int64,2}:
 5  10  7
 8   9  6
 7  11  0
```

In [3]:
```
mlp = Model()

@defVar(mlp, x[1:n,1:n] ≥ 0)
@defVar(mlp, y[1:n], Bin)

for j=1:n
    @addConstraint(mlp, sum{x[i,j], i=1:n} == 1)
end

@addConstraint(mlp, sum{y[i], i=1:n} == p)

for i=1:n, j=1:n
    @addConstraint(mlp, x[i,j] - y[i] ≤ 0)
end

@setObjective(mlp, Min, sum{c[i,j] * x[i,j], i=1:n, j=1:n})

mlp
```

Out[3]:

$$
\begin{aligned}
\min \quad & 5x_{1,1} + 10x_{1,2} + 7x_{1,3} + 8x_{2,1} + 9x_{2,2} + 6x_{2,3} + 7x_{3,1} + 11x_{3,2} \\
\text{Subject to} \quad & x_{1,1} + x_{2,1} + x_{3,1} = 1 \\
& x_{1,2} + x_{2,2} + x_{3,2} = 1 \\
& x_{1,3} + x_{2,3} + x_{3,3} = 1 \\
& y_1 + y_2 + y_3 = 2 \\
& x_{1,1} - y_1 \leq 0 \\
& x_{1,2} - y_1 \leq 0 \\
& x_{1,3} - y_1 \leq 0 \\
& x_{2,1} - y_2 \leq 0 \\
& x_{2,2} - y_2 \leq 0 \\
& x_{2,3} - y_2 \leq 0 \\
& x_{3,1} - y_3 \leq 0 \\
& x_{3,2} - y_3 \leq 0 \\
& x_{3,3} - y_3 \leq 0 \\
& x_{i,j} \geq 0 \quad \forall i \in \{1,2,3\}, j \in \{1,2,3\} \\
& y_i \in \{0,1\} \quad \forall i \in \{1,2,3\}
\end{aligned}
$$

In [4]:
```
solve(mlp)
```

Out[4]:
```
:Optimal
```

In [5]:
```
typeof(getInternalModel(mlp))
```

Out[5]:
```
Cbc.CbcMathProgSolverInterface.CbcMathProgModel
```

In [6]:
```
getObjectiveValue(mlp)
```

Out[6]:
```
15.0
```

In [7]:
```
getValue(y)
```

Out[7]:
```
3-element Array{Float64,1}:
 1.0
 0.0
 1.0
```

In [8]:
```
getValue(x)
```

Out[8]:
```
3x3 Array{Float64,2}:
 1.0  1.0  0.0
 0.0  0.0  0.0
 0.0  0.0  1.0
```