

Práctica Programación Lineal

David Rodríguez Bacelar (david.rbacelar@udc.es)

Lucas Campos Camiña (lucas.campos@udc.es)

Diego Vázquez Fernández (diego.vazquez.fernandez@udc.es)

Problema 1

- Planteamiento:

```
- Variables:
  • Tomarán el valor 1 si el candidato forma parte de la comisión
  • 0 en caso contrario

- Objetivo:
Minimizar      a + b + c + d + e + f + g + h + i + j

- Restricciones:
S.A.           a + b + c + d + e >= 1 (al menos una mujer)
                f + g + h + i + j >= 1 (al menos un hombre)
                a + b + c + j >= 1 (al menos un estudiante)
                e + f >= 1 (al menos un administrativo)
                d + g + h + i >= 1 (al menos un profesor)
                d + g + h + i >= e + f (mayor o igual número de
profesores que administrativos)
                a + b + c + d + e = f + g + h + i + j (mismo número de
hombres y mujeres)
```

- Codificación:

```
from pyomo.environ import *
from pyomo.opt import SolverFactory

model= ConcreteModel()

model.a=Var(domain=Binary)
model.b=Var(domain=Binary)
model.c=Var(domain=Binary)
model.d=Var(domain=Binary)
model.e=Var(domain=Binary)
model.f=Var(domain=Binary)
model.g=Var(domain=Binary)
model.h=Var(domain=Binary)
model.i=Var(domain=Binary)
model.j=Var(domain=Binary)

model.profit=Objective(expr=model.a+model.b+model.c+model.d+model.e+model.f+model.g+model.h+model.i+model.j, sense=minimize)

model.r1=Constraint(expr=model.a+model.b+model.c+model.d+model.e >= 1)
model.r2=Constraint(expr=model.f+model.g+model.h+model.i+model.j >= 1)
model.r3=Constraint(expr=model.a+model.b+model.c+model.j >= 1)
model.r4=Constraint(expr=model.e+model.f >= 1)
model.r5=Constraint(expr=model.d+model.g+model.h+model.i >= 1)
model.r6=Constraint(expr=model.a+model.b+model.c+model.d+model.e == model.f+model.g+model.h+model.i+model.j)
model.r7=Constraint(expr=model.d+model.g+model.h+model.i >= model.e+model.f)

results=SolverFactory('glpk').solve(model)
results.write()

if results.solver.status=='ok':
    print('Objetivo=', model.profit)
    for v in model.component_data_objects(Var):
        print(str(v), v.value)
```

- Resultados:

```
Objetivo = 4.0  
a 1.0  
b 0.0  
c 0.0  
d 0.0  
e 1.0  
f 0.0  
g 0.0  
h 0.0  
i 1.0  
j 1.0
```

- Interpretación:

La solución que nos ofrece el solver es óptima (el mínimo número de personas de la comisión tiene que ser 4, dadas las restricciones), pero no es la única, ya que por ejemplo los candidatos C,E,I,J también cumplirían las restricciones.

Problema 2

- Planteamiento:

- Variables (t_{xy}):

- Tomarán el valor 1 si la tarea x la realiza el trabajador y
- 0 en caso contrario

- Objetivo:

Minimizar $65 * t_{11} + 67 * t_{12} + 68 * t_{13} + 67 * t_{14} + 71 * t_{15} + 69 * t_{16} + 73 * t_{21} + 70 * t_{22} + 72 * t_{23} + 75 * t_{24} + 69 * t_{25} + 71 * t_{26} + 63 * t_{31} + 65 * t_{32} + 69 * t_{33} + 70 * t_{34} + 75 * t_{35} + 66 * t_{36} + 57 * t_{41} + 58 * t_{42} + 55 * t_{43} + 59 * t_{44} + 57 * t_{45} + 59 * t_{46}$

- Restricciones:

S.A.

(Cada tarea solo la realiza un trabajador)

$$t_{11} + t_{12} + t_{13} + t_{14} + t_{15} + t_{16} = 1$$

$$t_{21} + t_{22} + t_{23} + t_{24} + t_{25} + t_{26} = 1$$

$$t_{31} + t_{32} + t_{33} + t_{34} + t_{35} + t_{36} = 1$$

$$t_{41} + t_{42} + t_{43} + t_{44} + t_{45} + t_{46} = 1$$

(Cada trabajador solo realiza una tarea)

$$t_{11} + t_{21} + t_{31} + t_{41} \leq 1$$

$$t_{12} + t_{22} + t_{32} + t_{42} \leq 1$$

$$t_{13} + t_{23} + t_{33} + t_{43} \leq 1$$

$$t_{14} + t_{24} + t_{34} + t_{44} \leq 1$$

$$t_{15} + t_{25} + t_{35} + t_{45} \leq 1$$

$$t_{16} + t_{26} + t_{36} + t_{46} \leq 1$$

- Codificación:

```
f.obj <-
c(65, 67, 68, 67, 71, 69, 73, 70, 72, 75, 69, 71, 63, 65, 69, 70, 75, 66, 57, 58, 55, 59, 57, 59);

f.const <- rbind(c(1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
                 c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
                 c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0),
                 c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1),
                 c(1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0),
                 c(0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0),
                 c(0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0),
                 c(0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0),
                 c(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0),
                 c(0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1));

f.cond <- c("=", "=", "=", "=", "<=", "<=", "<=", "<=", "<=", "<=");

f.rhs <- c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1);

# Solve
solution <- lp("min", f.obj, f.const, f.cond, f.rhs)

solution$objval

solution$solution
```

- Resultados:

```
> solution$objval
[1] 254
> solution$solution
[1] 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0
# (Tarea 1 - Trabajador 1, Tarea 2 - Trabajador 5, Tarea 3 -
Trabajador 2, Tarea 4 - Trabajador 3)
```

- Interpretación:

El mínimo número de horas es 254 con la plantilla de trabajadores anterior.