Anexo 1: Resolución de los problemas con SAGE

Problema de optimización

Trabajo de Curso

Universidad de Sevilla

Ingeniería Informática Tecnologías Informáticas

Matemática aplicada a Sistemas de Información - Tercer curso

Juan Arteaga Carmona
Herrera, Sevilla, España
JuanArteaga@andalu30.me
Participación: 50%

Enrique Ramos Miró
Sanlúcar la Mayor, Sevilla, España
kikeramos9@gmail.com
Participación: 50%

1. Resolución del problema considerando que no hay un mínimo de vacantes en cada departamento.

1.1 Modelado mediante Programación lineal del problema propuesto.

```
p=MixedIntegerLinearProgram(maximization=False)
x=p.new_variable(nonnegative=True,binary=true)
```

```
p.add_constraint(x[11]+x[12]+x[13]+x[14]==1)
p.add_constraint(x[21]+x[22]+x[23]+x[24]==1)
p.add_constraint(x[31]+x[32]+x[33]+x[34]==1)
p.add_constraint(x[41]+x[42]+x[43]+x[44]==1)
```

```
p.set_objective(3*x[11]+0*x[12]+2*x[13]+6*x[14]+2*x[21]+x[22]+4*x[23]+5*x[24]+4*x[31]+2*x[32]+5*x[33]+7*x[34]+2*x[41]+0*x[42]+2*x[43]+4*x[44])
```

1.2 Aplicación de un metodo de resolución para obtener el resultado al primer apartado

```
p.solve()
```

```
    3.0
```

```
p.get_values(x)
```

```
    {11: 0.0,
     12: 1.0,
     13: 0.0,
     14: 0.0,
     21: 0.0,
     22: 1.0,
     23: 0.0,
     24: 0.0,
     31: 0.0,
     32: 1.0,
     33: 0.0,
     34: 0.0,
     41: 0.0,
     42: 1.0,
     43: 0.0,
     44: 0.0}
```

Como podemos ver en el resultado anterior las variables seleccionadas han sido 12, 22, 32, 42, lo que indica que todos los estudiantes han sido seleccionado en el departamento de maternidad ya que esta es la mejor forma de minimizar los errores que se producirán.

1.3 Aplicación de un metodo de resolución para obtener el resultado al segundo apartado

```
p=MixedIntegerLinearProgram(maximization=False)
```

```
x=p.new_variable(nonnegative=True,binary=true)
```

```
p.add_constraint(x[11]+x[12]+x[13]+x[14]==1)
p.add_constraint(x[21]+x[22]+x[23]+x[24]==1)
p.add_constraint(x[31]+x[32]+x[33]+x[34]==1)
p.add_constraint(x[41]+x[42]+x[43]+x[44]==1)
```

```
bigM = 10000000000
```

```
p.set_objective(
3*x[11]+bigM*x[12]+2*x[13]+6*x[14]+
2*x[21]+bigM*x[22]+4*x[23]+5*x[24]+
4*x[31]+bigM*x[32]+5*x[33]+7*x[34]+
2*x[41]+bigM*x[42]+2*x[43]+4*x[44])
```

```
p.solve()
```

```
    10.0
```

```
p.get_values(x)
```

```
    {11: 0.0,
     12: 0.0,
```

```
 13: 1.0,
 14: 0.0,
 21: 1.0,
 22: 0.0,
 23: 0.0,
 24: 0.0,
 31: 1.0,
 32: 0.0,
 33: 0.0,
 34: 0.0,
 41: 0.0,
 42: 0.0,
 43: 1.0,
 44: 0.0}
```

En este caso obtenemos un resultado más variado:

- El estudiante 1 ha sido asignado al departamento 3 (Medicina)
- El estudiante 2 ha sido asignado al departamento 1 (Urgencias)
- El estudiante 3 ha sido asignado al departamento 1 (Urgencias)
- El estudiante 4 puede ser asignado tanto al departamento 1 como al 3 (Urgencias y Medicina), ya que tendrían el mismo número de errores fatales (2) en ambos departamentos. De modo que cada vez que ejecutemos la función 'p.get_values(x)' unas veces saldrá que se le ha sido asignado el departamento 1 y otras el 3

2. Resolución del problema considerando que solo hay una vacante en cada departamento.

La solución será la misma que se optiene por el metodo húngaro.

2.1 Modelado mediante Programación lineal del problema propuesto.

```
p=MixedIntegerLinearProgram(maximization=False)
x=p.new_variable(nonnegative=True,binary=true)
```

```
# Todos los alumnos tienen que ser destinados a un departamento
p.add_constraint(x[11]+x[12]+x[13]+x[14]==1)
p.add_constraint(x[21]+x[22]+x[23]+x[24]==1)
p.add_constraint(x[31]+x[32]+x[33]+x[34]==1)
p.add_constraint(x[41]+x[42]+x[43]+x[44]==1)

# Todos los departamentos tienen que recibir algún alumno
p.add_constraint(x[11]+x[21]+x[31]+x[41]==1)
p.add_constraint(x[12]+x[22]+x[32]+x[42]==1)
p.add_constraint(x[13]+x[23]+x[33]+x[43]==1)
p.add_constraint(x[14]+x[24]+x[34]+x[44]==1)
```

```
p.set_objective(3*x[11]+0*x[12]+2*x[13]+6*x[14]+2*x[21]+x[22]+4*x[23]+5*x[24]+4*x[31]+2*x[32]+5*x[33]+7*x[34]+2*x[41]+0*x[42]+2*x[43]+4*x[44])
```

2.2 Aplicación de un metodo de resolución para obtener el resultado al primer apartado

```
p.solve()
```

```
10.0
```

```
p.get_values(x)
```

```
{11: 0.0,
 12: 0.0,
 13: 1.0,
 14: 0.0,
 21: 1.0,
 22: 0.0,
 23: 0.0,
 24: 0.0,
 31: 0.0,
 32: 1.0,
 33: 0.0,
 34: 0.0,
 41: 0.0,
 42: 0.0,
 43: 0.0,
 44: 1.0}
```

Como podemos observar, en la solucion:

- El alumno 1 será destinado al departamento 3 (Medico)
- El alumno 2 será destinado al departamento 1 (Urgencias)
- El alumno 3 será destinado al departamento 2 (Maternidad)
- El alumno 4 será destinado al departamento 4 (Cirugía)

2.3 Aplicación de un metodo de resolución para obtener el resultado al segundo apartado

```
p=MixedIntegerLinearProgram(maximization=False)
x=p.new_variable(nonnegative=True,binary=true)
```

```
# Todos los alumnos tienen que ser destinados a un departamento
p.add_constraint(x[11]+x[12]+x[13]+x[14]==1)
p.add_constraint(x[21]+x[22]+x[23]+x[24]==1)
p.add_constraint(x[31]+x[32]+x[33]+x[34]==1)
p.add_constraint(x[41]+x[42]+x[43]+x[44]==1)

# Todos los departamentos tienen que recibir algún alumno
p.add_constraint(x[11]+x[21]+x[31]+x[41]==1)
p.add_constraint(x[12]+x[22]+x[32]+x[42]==1)
p.add_constraint(x[13]+x[23]+x[33]+x[43]==1)
```

```
p.add_constraint(x[14]+x[24]+x[34]+x[44]==1)
```

```
bigM = 10000000000
```

```
p.set_objective(
3*x[11]+       0*x[12]+ bigM*x[13]+    6*x[14]+
bigM*x[21]+      x[22]+    4*x[23]+    5*x[24]+
4*x[31]+    bigM*x[32]+    5*x[33]+    7*x[34]+
2*x[41]+       0*x[42]+    2*x[43]+ bigM*x[44])
```

```
p.solve()
```

```
11.0
```

```
p.get_values(x)
```

```
{11: 0.0,
 12: 1.0,
 13: 0.0,
 14: 0.0,
 21: 0.0,
 22: 0.0,
 23: 0.0,
 24: 1.0,
 31: 1.0,
 32: 0.0,
 33: 0.0,
 34: 0.0,
 41: 0.0,
 42: 0.0,
 43: 1.0,
 44: 0.0}
```

Por lo tanto, la solución al segundo apartado en este caso será:

- El estudiante 1 será destinado al departamento 2 (Maternidad)
- El estudiante 2 será destinado al departamento 4 (Cirugía)
- El estudiante 3 será destinado al departamento 1 (Urgencias)
- El estudiante 4 será destinado al departamento 3 (Medicina)

```
```