

# HorariosMetaheurísticas

September 2, 2021

Resolución de una clase de problema de horario usando algoritmos genéticos

Máster en Matemáticas y Computación - Universidad de Cantabria Realizado por: Giuseppe Benito Bervis Quintero.

## 0.1 Clase “Schedule”

Para la resolución de este problema se desarrollo la clase “**Schedule**”, en Python, y esta posee los parámetros y métodos propios para resolver un problema de horario donde se desea: - Minimizar el número fechas y aulas ocupadas - Maximizar las distancias entre exámenes, para alumnos, y las holguras de aforos. En esta clase se contemplan las “hard constraints” usuales, a saber: - No es posible que un profesor o un alumno tengan examen de dos cosas a la vez, ni que un aula esté ocupada simultáneamente, ni que se supere su capacidad.

## 0.2 Input data:

- **C**: Lista de valores de ponderación de los pesos, en orden: “fechas, salones, Dist\_Ex y Holguras”
- **TE**: Matriz de profesores - exámenes (profesor  $i$  cuidará el examen  $j$ ).
- **SE**: Matriz de estudiantes - exámenes (estudiante  $i$  realizará examen  $j$ ).
- **R**: Lista de capacidades de los salones
- **days**: Cantidad de días en los que se desea acomodar el horario. (Se asumen 4 períodos por día).

## 0.3 Output:

Como salida se tiene: - Un diccionario con la mejor variable, su valor en la función objetivo, la última generación de individuos y sus valores en la función objetivo, y la cantidad de restricciones duras que viola la mejor solución. - Si la solución no viola ninguna restricción dura, genera una lista “horario”, indicando: «examen, día, periodo y salón».

## 1 Resolución de un problema

Este problema cuenta con 8 exámenes, 6 profesores, 5 estudiantes, 2 salones y 2 días para acomodar los exámenes (a linformación está contenida en los datos de entrada). La resolución se presenta a continuación, comentando los pasos entre el código.

[1]:



```

0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
0., 1., 0., 0., 0., 0., 0., 0., 0.]),
'function': 404999.1,
'last_generation': {'variables': array([[0., 0., 0., ..., 0., 0., 0.],
[0., 0., 0., ..., 0., 0., 0.],
[0., 0., 0., ..., 0., 0., 0.],
...,
[0., 0., 0., ..., 0., 0., 0.],
[0., 1., 0., ..., 0., 0., 0.],
[0., 0., 0., ..., 0., 1., 0.]])},
'scores': array([4.04999100e+05, 4.04999100e+05, 3.27725001e+08,
3.07332500e+09,
4.05004500e+09, 4.61765000e+09, 4.61765000e+09, 5.24292500e+09,
5.92965000e+09, 5.92965000e+09, 6.68172500e+09, 6.68172500e+09,
7.50317000e+09, 7.50317000e+09, 7.50317000e+09, 7.50317000e+09,
7.50317000e+09, 8.39812500e+09, 9.37085000e+09, 9.37085000e+09,
9.37085000e+09, 9.37085000e+09, 1.04257250e+10, 1.15672500e+10,
1.15672500e+10, 1.15672500e+10, 1.15672500e+10, 1.15672500e+10,
1.41288500e+10, 1.41288500e+10, 1.41288500e+10, 1.41288500e+10,
1.55585250e+10, 1.55585250e+10, 1.70940500e+10, 1.70940500e+10,
1.70940500e+10, 1.70940500e+10, 1.70940500e+10, 1.87405250e+10,
2.05031700e+10, 2.05031700e+10, 2.05031700e+10, 2.05031700e+10,
2.05031700e+10, 2.05031700e+10, 2.05031700e+10, 2.23873250e+10,
2.23873250e+10, 2.23873250e+10, 2.23873250e+10, 2.23873250e+10,
2.23873250e+10, 2.43984500e+10, 2.43984500e+10, 2.43984500e+10,
2.43984500e+10, 2.43984500e+10, 2.43984500e+10, 2.65421250e+10,
2.65421250e+10, 2.65421250e+10, 2.65421250e+10, 2.88240500e+10,
2.88240500e+10, 2.88240500e+10, 2.88240500e+10, 2.88240500e+10,
3.12500450e+10, 3.12500450e+10, 3.38260500e+10, 3.38260500e+10,
3.38260500e+10, 3.38260500e+10, 3.38260500e+10, 3.38260500e+10,
3.65581250e+10, 3.94524500e+10, 3.94524500e+10, 3.94524500e+10,
3.94524500e+10, 4.25153250e+10, 4.57531700e+10, 4.57531700e+10,
4.57531700e+10, 4.57531700e+10, 4.57531700e+10, 4.91725250e+10,
4.91725250e+10, 4.91725250e+10, 5.27800500e+10, 5.65825250e+10,
5.65825250e+10, 6.05868500e+10, 6.05868500e+10, 6.48000450e+10,
7.87648500e+10, 7.87648500e+10, 7.87648500e+10, 1.34369325e+11]))},
'hardC_violated': 3.0}

```

Puede ver que la solución encontrada viola 2 de las 4 restricciones duras, por lo que no imprime el horario.

## 2 Probando una solución al problema

Usando el código se logró encontrar una solución que no viola ninguna de las restricciones duras, por lo que podemos generar su horario respectivo.

```
[12]: # A Solution without constraint violated
Good_Sol = [0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
            0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1.,
            0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
            0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0.,
            0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
            0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0.,
            0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
            0., 0., 0., 0., 0., 1., 0., 0., 0.]

#Violación de restricciones duras
print(f'Cantidad de restricciones duras violadas por la solución: {Sch.
      ↳test(Good_Sol)}\n')

#Generando el horario
Sch.sc(Good_Sol)
```

Cantidad de restricciones duras violadas por la solución: 0.0

Examen 1 en el día 1, periodo 4 y salón 1,  
 Examen 2 en el día 1, periodo 3 y salón 1,  
 Examen 3 en el día 1, periodo 1 y salón 2,  
 Examen 4 en el día 2, periodo 1 y salón 1,  
 Examen 5 en el día 1, periodo 2 y salón 1,  
 Examen 6 en el día 2, periodo 4 y salón 1,  
 Examen 7 en el día 2, periodo 2 y salón 1,  
 Examen 8 en el día 2, periodo 3 y salón 1,

Puede verse que todos los exámenes están asignados a períodos y días diferentes, con lo que cumple las restricciones duras y los parámetros de ponderación de los pesos elegidos.

Se ha constatado que hay varios horarios para este problema particular, el presentado es solo uno de ellos.

### 3 Conclusión

Los problemas de horarios, muchos de ellos, son NP-duros, por lo que los algoritmos tradicionales son ineficientes, en este punto juegan un papel importante las estrategias heurísticas que, si bien no siempre permiten encontrar las soluciones exactas, por lo menos permiten encontrar buenas aproximaciones, y he ahí la importancia de su estudio.