

# Modelos Matemáticos Aplicados: Tribunales-Tesis

Brian Ameht Inclan Quesada  
Leonardo Ramirez Calatayud  
Eric Lopez Tornas  
Nahomi Lahera Champage  
Loitzel Morales Santiesteban

March 19, 2024

## Introducción

Este informe describe el código proporcionado que utiliza la biblioteca PuLP para resolver un problema de asignación de tribunales. PuLP es una biblioteca de modelado de programación lineal en Python que permite generar archivos MPS o LP y llamar a solvers como GLPK, COIN-OR CLP/CBC, CPLEX, GUROBI, MOSEK, XPRESS, CHOCO, MIPCL, HiGHS, SCIP/FSCIP para resolver problemas lineales.

## Modelo de Asignación de Tribunales

### Variables de decisión

$$x_{t,d,l,h} \in \{0, 1\}$$

donde  $x_{t,d,l,h}$  indica si el tribunal  $t$  está asignado al día  $d$ , en el lugar  $l$ , y en el horario  $h$ .

### Función objetivo

Minimizar la cantidad total de asignaciones:

$$\min \sum_{t=1}^T \sum_{d=1}^D \sum_{l=1}^L \sum_{h=1}^H x_{t,d,l,h}$$

## Restricciones

### 1. Asignación de tribunales

A lo sumo, un tribunal por día, lugar y hora:

$$\sum_{t=1}^T x_{t,d,l,h} \leq 1, \quad \forall d, l, h$$

### 2. Cumplimiento de tesis asignadas por tribunal

Cada tribunal tiene asignada exactamente una tesis:

$$\sum_{d=1}^D \sum_{l=1}^L \sum_{h=1}^H x_{t,d,l,h} = 1, \quad \forall t$$

### 3. Restricciones específicas de personas

Para cada persona  $i$ , con restricciones por horario  $H_i$ , lugar  $L_i$ , y día  $D_i$ , y perteneciente a tribunales  $P_i$ :

Restricciones por horario:

$$\sum_{d=1}^D \sum_{l=1}^L x_{t,d,l,h} = 0, \quad \forall t \in P_i, \forall h \in H_i$$

Restricciones por lugar:

$$\sum_{d=1}^D \sum_{h=1}^H x_{t,d,l,h} = 0, \quad \forall t \in P_i, \forall l \in L_i$$

Restricciones por día:

$$\sum_{l=1}^L \sum_{h=1}^H x_{t,d,l,h} = 0, \quad \forall t \in P_i, \forall d \in D_i$$

### 4. Restricciones de traslado entre lugares

Una persona no puede estar asignada a lugares diferentes en horarios consecutivos:

$$\sum_{l1 \neq l2} (x_{t,d,l1,h} + x_{t',d,l2,h+1}) \leq 1, \quad \forall t, t', d, h$$

## Descripción General

El código define dos funciones principales:

## Funciones Principales

- **creando\_T(Tribunales):** Esta función crea una matriz  $T$  que representa la asignación de personas a tribunales. La matriz  $T$  es una matriz binaria donde  $T[i][j]$  es 1 si la persona  $j + 1$  está asignada al tribunal  $i + 1$ , y 0 en caso contrario. La función también devuelve el número total de personas.
- **resolver\_asignacion\_tribunales(Tribunales, dias, lugares, horarios, restricciones):** Esta función resuelve el problema de asignación de tribunales utilizando PuLP. Define un problema de optimización donde el objetivo es minimizar el número total de asignaciones de tribunales a días, lugares y horarios, sujeto a varias restricciones.

## Detalles de la Implementación

La creación de la matriz de asignación y la definición del problema de optimización se realizan de manera eficiente, implementando restricciones específicas para asegurar la factibilidad de la asignación de tribunales.

## Utilización del Programa

Para utilizar correctamente el programa de asignación de tribunales, es necesario proporcionar los datos necesarios en el formato adecuado.

## Definición de Datos

Los tribunales se representan como una lista de listas, donde cada sublista contiene los números de las personas asignadas a ese tribunal. Los parámetros de días, lugares y horarios determinan las opciones disponibles para asignar los tribunales. Las restricciones específicas de personas pueden incluir restricciones de día, lugar y hora.

## Instalación de Software Necesario

Para ejecutar el código y trabajar en este proyecto, necesitarás instalar lo siguiente en tu sistema:

1. Python: Visita <https://www.python.org/downloads/> y descarga la última versión para tu sistema operativo. Sigue las instrucciones de instalación proporcionadas en el sitio web.
2. PuLP: Una vez que tengas Python instalado, puedes instalar PuLP usando pip, el gestor de paquetes de Python. Abre una terminal o línea de comandos y ejecuta `pip install pulp`.

## Descripción del Código

El script de Python se basa en dos funciones principales para resolver el problema de asignación de tribunales:

- `creando_T(Tribunales)`: Genera una matriz binaria que indica la asignación de personas a tribunales, junto con el total de personas.
- `resolver_asignacion_tribunales(Tribunales, dias, lugares, horarios, restricciones)`: Utiliza PuLP para definir y resolver el problema de asignación de tribunales dadas ciertas restricciones y parámetros.

## Cómo Utilizar el Programa

Para utilizar el programa, necesitas preparar los datos de entrada:

1. Define los tribunales y las personas asignadas a cada uno en la lista `Tribunales`.
2. Establece los parámetros del problema (`dias`, `lugares`, `horarios`).
3. Define las restricciones específicas para algunas personas en el diccionario `restricciones`.

## Ejecución del Código

Para ejecutar el script:

1. Abre una terminal o línea de comandos.
2. Navega al directorio donde está guardado el script.
3. Ejecuta el script con Python mediante el comando `python test.py`.
4. Observa las asignaciones de tribunales generadas por el programa.

## Utilización del programa

Para utilizar correctamente el programa de asignación de tribunales, es necesario proporcionar los datos necesarios en el formato adecuado. A continuación, se detallan los pasos para introducir los datos:

1. **Definir los Tribunales:** Los tribunales se representan como una lista de listas, donde cada sublista contiene los números de las personas asignadas a ese tribunal. Por ejemplo:

```
Tribunales = [[1, 2, 3], [4, 5], [6, 7, 8, 9]]
```

2. **Definir los parámetros de días, lugares y horarios:** Estos parámetros representan las opciones disponibles para asignar los tribunales. Por ejemplo:

```
d = 5 # Días disponibles
l = 3 # Lugares disponibles
h = 8 # Horarios disponibles
```

3. **Definir las restricciones:** Las restricciones se aplican a ciertas personas y pueden incluir restricciones de día, lugar y hora. Por ejemplo:

```
restricciones = {
    1: {'d': [1, 2], 'l': [0], 'h': []},
    3: {'d': [3, 4], 'l': [1], 'h': []}
}
```

Una vez que se han definido estos datos, se puede llamar a la función `resolver_asignacion_tribunales` con los parámetros adecuados para resolver el problema de asignación. A continuación, se muestra cómo llamar a esta función y cómo imprimir el resultado:

```
resultado = resolver_asignacion_tribunales(Tribunales, d, l,
                                           h, restricciones)
print(resultado)
```

La función `resolver_asignacion_tribunales` devuelve un diccionario con las asignaciones de tribunales, que luego se pueden imprimir o guardar utilizando la función `save`.

Es *importante* asegurarse de que los datos proporcionados sean coherentes con las restricciones y los parámetros definidos. Por ejemplo, el número de tribunales en `Tribunales` debe coincidir con la longitud de `Ct`, y los números de personas en cada tribunal deben estar dentro del rango de `total_personas` definido por el programa.

## Representación de Resultados

El programa imprime las asignaciones de tribunales y, utilizando la función `save`, puedes guardar estas asignaciones en un formato estructurado, como un archivo Excel, para su análisis y revisión posteriores. El archivo se guarda en la ruta `data/horarios_matriz.xlsx` dentro del directorio del proyecto.

## Conclusión

Este proyecto demuestra cómo se puede utilizar la programación lineal y PuLP para resolver problemas complejos de asignación en entornos académicos. Las mejoras futuras podrían incluir la optimización de restricciones adicionales o la integración de una interfaz gráfica para facilitar la entrada de datos y la visualización de resultados.