

# Informe de Solución: Planificación de Horarios

Jackson C. Vera Pineda - estudiante de 4<sup>to</sup> año, Ciencia de la Computación,  
facultad de Matemática y Computación, Universidad de La Habana

Massiel Paz Otaño - estudiante de 4<sup>to</sup> año, Ciencia de la Computación,  
facultad de Matemática y Computación, Universidad de La Habana

1 de febrero de 2025

## 1. Introducción

Este informe aborda el problema de la planificación de horarios, un problema clásico de optimización combinatoria con múltiples restricciones. El objetivo es diseñar un horario que minimice conflictos, maximice el uso eficiente de recursos y satisfaga las preferencias de los estudiantes y profesores. A continuación, se presenta la formalización del problema, los subproblemas clave, estrategias de solución, análisis de complejidad y demostraciones de correctitud.

## 2. Formalización del Problema

El problema se modela como un problema de optimización combinatoria con restricciones duras y blandas. A continuación, se definen los conjuntos, variables, restricciones y función objetivo.

### 2.1. Conjuntos y Variables

- $C = \{c_1, c_2, \dots, c_n\}$ : Conjunto de cursos/clases.
- $P = \{p_1, p_2, \dots, p_m\}$ : Conjunto de profesores.
- $S = \{s_1, s_2, \dots, s_k\}$ : Conjunto de estudiantes.
- $R = \{r_1, r_2, \dots, r_l\}$ : Conjunto de aulas/recursos.
- $T = \{t_1, t_2, \dots, t_q\}$ : Conjunto de franjas horarias.
- $x_{c,r,t} \in \{0, 1\}$ : Variable binaria que indica si la clase  $c$  se asigna al aula  $r$  en el tiempo  $t$ .
- $y_{p,c} \in \{0, 1\}$ : Variable binaria que indica si el profesor  $p$  está asignado a la clase  $c$ .

### 2.2. Restricciones

- No superposición:

$$\sum_{c \in C} x_{c,r,t} \leq 1 \quad \forall r \in R, t \in T$$

- Disponibilidad de profesores:

$$\sum_{c \in C} y_{p,c} \cdot x_{c,r,t} \leq 1 \quad \forall p \in P, t \in T$$

- Aforo:

$$\text{Capacidad}(r) \geq \text{Estudiantes}(c) \quad \text{si } x_{c,r,t} = 1$$

## 2.3. Función Objetivo

Minimizar  $\alpha \cdot (\text{Conflictos}) + \beta \cdot (\text{Subutilización de aulas}) + \gamma \cdot (\text{Preferencias insatisfechas})$

## 3. Subproblemas Clave y Estrategias de Solución

### 3.1. Subproblema 1: Asignación de Profesores sin Conflictos

- **Formalización:** Asignar cada clase a un profesor sin superposiciones.
- **Estrategia:** Modelar como un problema de emparejamiento en grafos bipartitos.
- **Algoritmo:** Algoritmo húngaro.
- **Correctitud:** Garantiza una asignación óptima en tiempo polinomial.
- **Complejidad:**  $O(m^3)$ .

### 3.2. Subproblema 2: Programación de Clases en Aulas y Horarios

- **Formalización:** Asignar  $(c, r, t)$  sin violar restricciones duras.
- **Estrategia:** Modelar como un problema de coloración de grafos.
- **Algoritmo:** Heurística greedy con ordenamiento por grado de conflicto.
- **Correctitud:** Encuentra una coloración con  $\leq k + 1$  colores.
- **Complejidad:**  $O(n^2)$ .

### 3.3. Subproblema 3: Manejo de Preferencias de Estudiantes

- **Formalización:** Optimizar restricciones blandas.
- **Estrategia:** Programación lineal entera (ILP) con relajación Lagrangiana.
- **Complejidad:** NP-Hard, pero relajaciones aproximadas en  $O(n^3)$ .

### 3.4. Subproblema 4: Condiciones Aleatorias (Eventos Imprevistos)

- **Formalización:** Replanificar ante cancelaciones de clases.
- **Estrategia:** Algoritmos online con ventanas deslizantes.
- **Análisis competitivo:** Cota de rendimiento frente al óptimo offline.

## 4. Demostración de NP-Hardness

El problema se reduce al problema de coloración de grafos (Graph Coloring), que es NP-Completo.

### Reducción

- Construir un grafo  $G$  donde cada nodo representa una clase  $c$ .
- Crear una arista entre dos nodos si comparten: mismo profesor, estudiantes en común (evitar superposición), y/o requisitos de aula especializada.
- Asignar colores (pares  $(r, t)$ ) a los nodos

Por lo tanto, Si podemos resolver la planificación de horarios en tiempo polinomial, entonces podemos resolver Graph Coloring en tiempo polinomial, lo cual es imposible a menos que  $P = NP$ .

## 5. Cotas Mínimas y Análisis de Complejidad

- **Cota inferior para el problema general:** cualquier algoritmo exacto requiere  $\Omega(2^n)$  en el peor caso (por la reducción a Graph Coloring).
- **Algoritmos de aproximación:** No se puede aproximar mejor que  $O(n^{1-\epsilon})$  a menos que  $P = NP$ .

## 6. Estrategias de Solución Parcial/Total

- **Algoritmos Greedy:** Asignar primero las clases con mayor restricciones.
- **Algoritmos de Aproximación:** ILP con relajación lineal.
- **Metaheurísticas:** Algoritmos genéticos o simulated annealing.

## 7. Conclusión

Este informe ha presentado un enfoque estructurado para resolver el problema de planificación de horarios, demostrando su naturaleza NP-Hard y proponiendo estrategias para abordar subproblemas con análisis de complejidad y correctitud. Se recomienda implementar soluciones parciales usando enfoques híbridos o metaheurísticas para instancias grandes.