

APLICACIONES COMPUTACIONALES EN NEGOCIOS

Informe – TP2

Mariño Martina, Martinez Kiara y Taié Colette

Índice

1. Consigna 1: Asignación de parciales	2
1.1. Definición de conjuntos, parámetros y variables	2
1.2. Razonamiento detrás de la función objetivo	2
2. Consigna 2: Extensión con restricción adicional	3
2.1. Razonamiento detrás del modelo	3
3. Consigna 3: Maximización de la separación temporal	4
4. Resultados generales y discusión	4
5. Conclusiones	4

Índice de figuras

Índice de cuadros

1. Consigna 1: Asignación de parciales

1.1. Definición de conjuntos, parámetros y variables

Para comenzar, definiremos los conjuntos con los que vamos a trabajar a lo largo del proyecto (algunos de ellos mencionados en el enunciado):

- P : Conjunto de parciales (índices p), que interpretaremos como un equivalente a conjunto de cursos (dado que solo hay un parcial por curso).
- D : Conjunto de días disponibles para asignar parciales, donde $D = (1, 2, 3, \dots, 12) - (6, 7, 8)$
- T : Horarios disponibles por días, donde $T = (9, 12, 15, 18)$ (índices t)
- $a_p \in Z_+$: Número de aulas que requiere el parcial $p \quad \forall p \in P$
- A : Aulas totales disponibles en cada slot ($A = 75$)
- E : Conjunto de pares incompatibles (p, q) . Si $(p, q) \in E \rightarrow$ no pueden coincidir en el mismo slot.

Ahora, tambien definimos la variable indicadora $x_{p,d,t} \in \{0, 1\}$ que nos señala si un parcial p está asignado al slot d, t . $x_{p,d,t} = 1$ si el parcial se programa en el slot (d, t) , y 0 en caso contrario.

1.2. Razonamiento detrás de la función objetivo

El objetivo principal del problema planteado es determinar en qué día y horario tomar cada uno de los parciales, respetando restricciones de incompatibilidad y disponibilidad de aulas. El enunciado establece explícitamente que:

“No siempre será posible programar todos los parciales cumpliendo las restricciones, y en ese caso buscamos maximizar la cantidad de parciales programados.”

A partir de esta indicación, la función objetivo elegida es:

$$\max \sum_{p \in P} \sum_{d \in D} \sum_{t \in T} x_{p,d,t}$$

Sujeto a:

$$\sum_{p \in P} x_{p,d,t} \leq 1 \quad \forall p \in P \quad (1)$$

Que los parciales asignados solo estén asignados a un slot.

$$\sum_{p \in P} a_p x_{p,d,t} \leq 75 \quad \forall d \in D, \forall t \in T \quad (2)$$

Que todos los parciales asignados no superen la cantidad de aulas disponibles por día.

$$x_{p,d,t} + x_{q,d,t} \leq 1 \quad \forall (p, q) \in E, \forall d \in D, \forall t \in T \quad (3)$$

No pueden haber dos parciales asignados al mismo slot si tienen alumnos que cursan ambas materias.

2. Consigna 2: Extensión con restricción adicional

Si bien el enunciado no nos provee información individual sobre qué estudiantes cursan qué materias, sí contamos con el grafo de incompatibilidad $G = (P, E)$, donde cada arista (p, q) indica que los cursos p y q tienen estudiantes en común.

Esta limitación es importante: **no podemos modelar explícitamente estudiantes individuales**, sino únicamente relaciones entre cursos. Por lo tanto, debemos buscar una forma de asegurar que no existan “tripletes” de parciales mutuamente relacionados que caigan el mismo día.

2.1. Razonamiento detrás del modelo

El enunciado establece que queremos evitar que “haya tres vértices vecinos en G programados un mismo día”. Esto implica que si tres materias p , q y r tienen estudiantes en común (es decir, si forman un triángulo en el grafo de incompatibilidad), entonces no pueden ser asignadas todas al mismo día.

Sin embargo, dado que la estructura del grafo no necesariamente es simple, el problema consiste en traducir esa idea a una restricción operativa que use las herramientas disponibles en el modelo.

La clave del razonamiento es:

- Para cada parcial p , la suma $\sum_{t \in T} x_{p,(d,t)}$ indica si el parcial fue programado en el día d .
- Si dos cursos p y q son adyacentes en G , ya hay una restricción que impide que estén en el mismo *slot*, pero no impide que estén el mismo día en horarios distintos.
- Para evitar que un estudiante tenga **tres parciales el mismo día**, no es suficiente controlar pares: necesitamos controlar **conjuntos de tres cursos que comparten estudiantes**.

Pero como no conocemos explícitamente esos tripletes, debemos inferirlos del grafo.

La aproximación práctica es:

Si un curso p tiene varios vecinos en el grafo, y tres de ellos se planifican el mismo día, entonces podría existir un estudiante en común afectado.

Por lo tanto, no queremos permitir que un curso p tenga dos vecinos en su vecindad programados el mismo día que él.

- **La restricción captura bien el problema.** Si p tiene dos vecinos q y r programados el mismo día, y p también está ese día, entonces existe un alto riesgo de que un estudiante tenga tres parciales.
- **Es computacionalmente bastante eficiente.** No requiere enumerar todos los posibles triangulos, lo cual sería costoso; sólo examina la vecindad de cada nodo.

Ahora ¿Cómo lo resolvimos?

- 3. Consigna 3: Maximización de la separación temporal**
- 4. Resultados generales y discusión**
- 5. Conclusiones**