UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Universidad Técnica Federico Santa María

Departamento de Informática
Universidad Técnica Federico Santa María

Optimización

Proyecto semestral

Paralelo: 1

Matias Lara	ROL: 202204030-0
José Meza	ROL: 202204041-6
Franco Cerda	ROL: 202204018-1
Matias Barraza	ROL: 202204030-3

Profesores: Sergio Campos -Yoslandy Lazo

Índice

1. Definición de conjuntos y parámetros

- I: Conjunto de asignaturas ($i \in I$)
- S: Conjunto de salas ($s \in S$)
- T: Conjunto de bloques horarios disponibles por día ($t \in T = \{1, 2, 3, 4, 5, 6, 7\}$)
- D: Días de la semana ($d \in D = \{1, 2, 3, 4, 5\}$).
- C_s: Capacidad de la sala s, que varía entre 20 y 45 estudiantes.
- E_i: Estudiantes interesados en la asignatura i, valor entre 10 y 40.
- $$A_i = \begin{cases} 1 \text{ Si la asignatura } i \text{ es indispensable.} \\ 0 \text{ en caso contrario.} \end{cases}$$
- P_i: Prioridad de la asignatura i, valor aleatorio entre 6 y 10 si es indispensable, y entre 1 y 5 si no lo es.
- $$B_{itd} = \begin{cases} 1 \text{ si el bloque } t \text{ del dia } d \text{ está bloqueado para el profesor que dicta la asignatura } i. \\ 0 \text{ caso contrario.} \end{cases}$$
- $$R_i = \begin{cases} 1 \text{ Si la asignatura } i \text{ requiere 1 bloque a la semana.} \\ 2 \text{ Si la asignatura } i \text{ requiere 2 bloques a la semana.} \end{cases}$$

2. Variables de decisión

$$x_{i,s,t,d} = \begin{cases} 1 \text{ si la asignatura } i \text{ se asigna a la sala } s \text{ en el bloque } t \text{ del día } d. \\ 0 \text{ en caso contrario.} \end{cases}$$

3. Función Objetivo

Maximizar la prioridad total de las asignaturas asignadas a las salas:

$$\text{Maximizar} \quad z = \sum_{i \in I} \sum_{s \in S} \sum_{t \in T} \sum_{d \in D} P_i x_{i,s,t,d}$$

4. Restricciones

- **Evitar solapamientos en una sala:** Cada sala solo puede tener una asignatura en un horario especifico.

$$\sum_{i \in I} x_{i,s,t,d} \leq 1 \quad \forall s \in S, \quad \forall t \in T, \quad \forall d \in D$$

La suma de todas las asignaturas en una sala s durante un bloque t en un día d debe ser 1 o menor, es decir, la sala **puede** estar asignada a una **única asignatura** o permanecer vacía.

- **Asignación de bloques:** Cada asignatura debe ser asignada a un número de bloques que coincide con su requerimiento, ya sea uno o dos bloques.

$$\sum_{s \in S} \sum_{t \in T} \sum_{d \in D} x_{i,s,t,d} = R_i \quad \forall i \in I$$

Es decir, si la asignatura requiere un bloque ($R_i = 1$), se asignará a un solo bloque. Si la asignatura requiere dos bloques ($R_i = 2$), se asignarán dos bloques consecutivos en el mismo día.

- **Bloques consecutivos:** Si una asignatura requiere dos bloques, estos deben ser asignados consecutivamente en el mismo día y en la misma sala.

$$X_{i,s,t,d} - X_{i,s,t+1,d} = 0 \quad \forall i \in I, \forall s \in S, \forall t \in \{0, \ldots, T-1\}, \forall d \in D$$

Notar que esta restricción evita que el bloque 7 sea elegido para asignaturas que requieren dos bloques consecutivos porque cuando el modelo intenta asignar dos bloques consecutivos a una asignatura, lo hace asegurando que si la asignatura i está asignada al bloque t en el día d, entonces también debe estar asignada al bloque inmediatamente siguiente $t+1$. Si $t = 7$ (el último bloque del día), la restricción no puede cumplirse porque no hay un bloque 8 en el mismo día ($8 \notin T$). Esto fuerza al modelo a no considerar el bloque 7 como una opción válida para las asignaturas que requieren dos bloques consecutivos.

- **Exclusión del último bloque del día**

$$x_{i,s,T,d} = 0 \quad \forall i \in I \text{ tal que } R_i = 2, \quad \forall s \in S, \quad \forall d \in D$$

Esta restricción asegura que si una asignatura i requiere dos bloques consecutivos ($R_i = 2$), dicha asignatura no podrá ser asignada al último bloque T del día d. Esto es necesario porque no existe un bloque consecutivo disponible para el último bloque del día, lo que haría imposible cumplir con la restricción de consecutividad de bloques en el mismo día. De esta forma, evitamos que se asignen dos bloques que crucen entre días.

- **Capacidad de la sala:** El número de estudiantes interesados en una asignatura no debe exceder la capacidad de la sala asignada.

$$E_i \times x_{istd} \leq C_s, \quad \forall i \in I, \quad \forall s \in S, \quad \forall t \in T, \quad \forall d \in D$$

Si $x_{istd} = 1$, es decir, si la asignatura i se ha asignado a la sala s en el bloque t del día d, entonces se cumple que $E_i \leq C_s$, lo que implica que la sala s tiene capacidad suficiente para albergar a todos los estudiantes interesados en la asignatura i.

- **Asignación de asignaturas indispensables:** Las asignaturas indispensables **deben** asignarse al menos una vez.

$$\sum_{s\in S}\sum_{t\in T}\sum_{d\in D} x_{istd} \geq A_i, \quad \forall i \in I$$

Si la asignatura es indispensable ($A_i = 1$), esta debe asignarse al menos una vez durante la semana. Si no es indispensable ($A_i = 0$), puede asignarse opcionalmente, permitiendo que sea asignada 0 o más veces según la disponibilidad de bloques.

- **Restricciones de horario de los profesores:** Los profesores tienen restricciones sobre en qué bloques pueden dictar clases.

$$x_{istd} \leq 1 - B_{itd}, \quad \forall i \in I, \quad \forall s \in S, \quad \forall t \in T, \quad \forall d \in D$$

De esta forma, si $B_{itd} = 1$, es decir, la asignatura i no se puede dictar en el bloque t del dia d debido al horario personal del profesor, la restricción se convierte en $x_{istd} \leq 0$, lo que implica que el bloque no se puede asignar a esa asignatura. En caso contrario, si $B_{itd} = 0$, la restricción se relaja y el modelo permite $x_{istd} \leq 1$, es decir, la asignatura i puede asignarse o no en ese bloque.

5. Descripción de la Función Objetivo y Restricciones

La función objetivo busca maximizar la prioridad de las asignaturas asignadas, priorizando aquellas que son indispensables. Las restricciones aseguran que se cumplan las limitaciones de capacidad de salas, disponibilidad de bloques horarios, y las necesidades específicas de cada asignatura, tanto en términos de bloques consecutivos como en la asignación de uno o dos bloques por semana, según sea necesario. Además, se da prioridad a las asignaturas indispensables, que deben ser asignadas obligatoriamente.

6. Análisis de la Función Objetivo

La función objetivo se genera a través de un generador de instancias, ya que involucra un número de variables igual a $I \times S \times T \times D$. La magnitud de la función objetivo depende directamente del tamaño de la instancia, ya que tanto I como S varían considerablemente entre instancias pequeñas, medianas y grandes.

A medida que crece el tamaño de la instancia, particularmente en el número de asignaturas (I) y salas (S), la función objetivo se vuelve más compleja debido al incremento exponencial en el número de combinaciones posibles de asignaturas y bloques horarios. Esto implica que, en instancias más grandes, el número de variables en el modelo aumenta significativamente, lo que puede complicar el proceso de optimización.

En instancias pequeñas, el número de variables es manejable y la solución se obtiene de forma eficiente. Sin embargo, en instancias medianas y grandes, el número de combinaciones posibles aumenta de manera exponencial, haciendo que el modelo de programación lineal requiera más recursos computacionales y tiempo para encontrar una solución óptima o factible. Esto puede provocar que, en instancias grandes, el tiempo de cómputo se incremente y la función objetivo tarde más en converger a un valor óptimo.

7. Análisis de Infactibilidad

La infactibilidad en el modelo puede ocurrir debido a varias razones, principalmente por la interacción entre las restricciones impuestas y la cantidad de parámetros disponibles. Una de las causas más comunes de infactibilidad surge cuando el modelo no logra asignar todas las asignaturas indispensables a los bloques horarios y salas disponibles. Esto puede ocurrir si la demanda de bloques horarios o de capacidad de las salas excede la oferta disponible, especialmente cuando hay más asignaturas que bloques o cuando los aforos de las salas son insuficientes para acomodar a todos los estudiantes interesados.

Otro factor relevante es la restricción de bloques consecutivos para ciertas asignaturas. Si la asignatura requiere dos bloques consecutivos en el mismo día, pero las salas o los bloques ya están ocupados por otras asignaciones, puede no ser posible encontrar una solución factible para esa asignatura. Además, las restricciones impuestas por los horarios bloqueados de los profesores pueden limitar aún más la disponibilidad de bloques, creando situaciones en las que no sea posible satisfacer todas las condiciones simultáneamente.

Es importante destacar que las restricciones relacionadas con la capacidad de las salas también juegan un rol crucial. Si la cantidad de estudiantes inscritos en una asignatura excede el aforo de la sala asignada, el modelo debe encontrar otra sala o bloque, lo cual puede no ser factible si las opciones son limitadas.

Finalmente, estas situaciones de infactibilidad se vuelven más comunes y difíciles de evitar en instancias grandes del problema. A medida que aumentan el número de asignaturas, salas, días y bloques, las combinaciones de restricciones se vuelven más complejas, lo que incrementa la probabilidad de que no exista una solución que satisfaga todas las condiciones impuestas de manera simultánea. Esto pone de relieve la importancia de generar instancias donde los parámetros estén balanceados y las restricciones sean manejables para garantizar la factibilidad del modelo.

8. Generación de Instancias y Uso de Aleatoriedad

El proceso de generación de instancias en este modelo se basa en la creación aleatoria de los parámetros clave que definen las restricciones y condiciones del problema. Para cada instancia, se generan aleatoriamente los siguientes parámetros:

- **Asignaturas (I):** Se crea un conjunto de 25 asignaturas en total. Cada asignatura puede tener un número aleatorio de estudiantes interesados, entre 10 y 40 estudiantes.

- **Salas (S):** Solo se cuenta con una sala en esta instancia, y su capacidad se genera de forma aleatoria entre 20 y 45 estudiantes.

- **Días y bloques horarios (D y T):** Se consideran 5 días de la semana, con 7 bloques horarios disponibles por día.

- **Asignaturas indispensables (A_i):** Se genera aleatoriamente si una asignatura es indispensable con una probabilidad del 20 % (es decir, 1 de cada 5 asignaturas será indispensable).
- **Prioridades (P_i):** Las asignaturas indispensables tienen una prioridad asignada entre 6 y 10, mientras que las no indispensables tienen una prioridad entre 1 y 5.
- **Requerimiento de bloques (R_i):** Se determina aleatoriamente si cada asignatura requiere uno o dos bloques semanales. El 65 % de las asignaturas solo requiere un bloque, mientras que el 35 % requiere dos bloques consecutivos.
- **Bloques bloqueados por los profesores (B_{itd}):** Los profesores tienen entre 7 y 21 bloques bloqueados semanalmente. Estos bloques se seleccionan de forma aleatoria para cada asignatura, garantizando que los profesores no estarán disponibles en esos horarios.

La función objetivo maximiza la asignación de asignaturas a las salas, priorizando las asignaturas indispensables según las prioridades definidas.

En cuanto a las restricciones, el modelo asegura que:

- No se produzcan solapamientos en las salas (es decir, solo se puede asignar una asignatura a la vez a una sala en un mismo horario).
- Las asignaturas se asignen el número correcto de bloques, según su requerimiento de uno o dos bloques semanales.
- La capacidad de las salas no sea excedida.
- Las asignaturas indispensables sean asignadas al menos una vez.
- Los horarios bloqueados por los profesores sean respetados.

Nota: La restricción que asegura que los bloques sean consecutivos para las asignaturas que requieren dos bloques aún no funciona correctamente. Esta restricción genera infactibilidad en todo el modelo, lo que provoca que el solver no pueda encontrar una solución factible en las instancias generadas.

En resumen, el modelo utiliza la aleatoriedad para generar diferentes parámetros en cada instancia y busca optimizar la asignación de asignaturas a las salas, respetando las restricciones impuestas por el problema.

9. Enlace GitHub

El código fuente y la documentación del proyecto están disponibles en el repositorio de GitHub:

`https://github.com/Matias-Lara/Proyecto1-OPTIMIZACION-INF292`