Heurística y Optimización

Curso 2022-23

Práctica 1

Alumnos: **Mark Burméster Azcárraga 100451175** 100451175@alumnos.uc3m.es
David San Román Carbayo 100451246 100451246@alumnos.uc3m.es

URL Github: https://github.com/100451175/Practica_1_Heuristica

ÍNDICE

1. CONTENIDOS DEL DOCUMENTO

Para comenzar con la memoria, haremos una pequeña introducción de los contenidos que se incluyen en ella.

En el enunciado de la práctica, se nos presenta un problema de optimización. Éste consiste en determinar el número de autobuses y las rutas que deben recorrer para recoger a una serie de alumnos de una serie de paradas. El objetivo es minimizar el coste, que se calcula en función del número de autobuses y los kilómetros recorridos. La práctica se divide en dos problemas con la misma temática y ligeras diferencias y, en este documento, mostramos en detalle los modelos que hemos implementado para ambos. También realizamos un análisis de los resultados y argumentamos cada una de las decisiones tomadas. Por último, ofrecemos una conclusión sobre la práctica.

2. MODELO 1

2.1. Descripción del modelo.

<u>Variables de decisión:</u>

x_{ij} : Variable binaria que vale 1 cuando la ruta pasa por el arco que representa.

Donde i y j son cada una de las paradas: origen, s1, s2, s3, d.

f_{ij} : Cantidad de alumnos en el bus en cada arco por el que pasa la ruta. (Flujo).

Donde i y j son cada una de las paradas.

Ejemplo con datos del ejemplo de la práctica:

X	o	s1	s2	s3	d		F	o	s1	s2	s3	d
o	0	1	0	1	0		o	0	0	0	0	0
s1	0	0	0	0	1		s1	0	0	0	0	15
s2	0	0	0	0	1		s2	0	0	0	0	15
s3	0	0	1	0	0		s3	0	0	10	0	0
d	0	0	0	0	0		d	0	0	0	0	0

Se puede observar que salen dos buses del origen. Uno hace el recorrido o-s1-d y el otro o-s3-s2-d. El primer bus al principio no lleva personas y finalmente 15. El segundo comienza sin llevar a nadie, luego lleva a 10 que recoge en s3 y finalmente lleva 15 al destino.

<u>Función objetivo:</u>

$$\text{min z : precio\_autobús} * \sum_{j \in Paradas} x_{\text{oj}} + \text{coste\_km} * \sum_{i,j \in Paradas} x_{\text{ij}} * distancia_{\text{ij}}$$

En el caso del ejemplo, el coste de las rutas sería de 120 × 2 + 5 × (8 + 6 + 10 + 5 + 7) = 420€.

<u>Restricciones:</u>

1. De las paradas salen una única ruta:

$$\sum_{i \in Paradas} x_{\text{ij}} = 0 \quad \forall\ j \in \text{Paradas} \ \ /\{origen, destino\}$$

2. A las paradas llega una única ruta:

$$\sum_{j \in Paradas} x_{\text{ij}} = 0 \quad \forall\ i \in \text{Paradas} \ \ /\{origen, destino\}$$

3. Al origen no llegan rutas:

$$\sum_{i \in Paradas} x_{\text{io}} = 0$$

4. Del destino no salen rutas:

$$\sum_{j \in Paradas} x_{\mathrm{dj}} = 0$$

5. No hay más rutas distintas que buses disponibles:

$$\sum_{j \in Paradas} x_{\mathrm{oj}} \leq n\_autobuses$$

6. El número de buses que salen del origen es el mismo que llega al destino:

$$\sum_{j \in Paradas} x_{\mathrm{oj}} = \sum_{i \in Paradas} x_{\mathrm{id}}$$

7. De cada parada entran y salen el mismo número de rutas (en este caso 1):

$$\sum_{j \in Paradas} x_{\mathrm{pj}} = \sum_{i \in Paradas} x_{\mathrm{ip}} \quad \forall\ p \in \mathrm{Paradas}\ \ /\{origen,\ destino\}$$

8. El flujo que sale de cada parada es igual al que entra más la gente que estuviera esperando en ella:

$$\sum_{j \in Paradas} f_{\mathrm{pj}} = gente_p + \sum_{i \in Paradas} x_{\mathrm{ip}} \quad \forall\, p \in \mathrm{Paradas}\ \ /\{origen,\ destino\}$$

9. La suma de los flujos que llegan al origen es igual a la suma de la gente esperando:

$$\sum_{i \in Paradas} f_{\mathrm{id}} = \sum_{i \in Paradas} gente_{\mathrm{i}}$$

10. El flujo donde no pasa la ruta es igual a 0:

$$f_{\mathrm{ij}} - 99 * x_{\mathrm{ij}} \leq 0 \quad i,j \in \mathrm{Paradas}$$

11. El flujo no puede superar la capacidad máxima del autobús:

$$f_{\mathrm{ij}} \leq capacidad\_autobus \quad i,j \in \mathrm{Paradas}$$

12. Tanto x como f son enteros no negativos:

$f_{\mathrm{ij}}, x_{\mathrm{ij}} \geq 0 \quad i, j \in \text{Paradas}$

2.2 Análisis de los resultados

Tras introducir todos los datos, restricciones y variables en un documento de LibreOffice Calc y, utilizando Solver, herramienta proporcionada por Calc, obtenemos la siguiente solución:

X	o	s1	s2	s3	d		F	o	s1	s2	s3	d
o	0	1	0	1	0		o	0	0	0	0	0
s1	0	0	1	0	0		s1	0	0	15	0	0
s2	0	0	0	0	1		s2	0	0	0	0	20
s3	0	0	0	0	1		s3	0	0	0	0	10
d	0	0	0	0	0		d	0	0	0	0	0

Coste: 400€.

Es decir, la solución óptima requiere dos rutas. La primera hace el recorrido o-s1-s2-d, llevando 15 personas entre s1 y s2, y 20 entre s2 y d. La segunda hace el recorrido o-s3-d, llevando 10 personas de s3 al destino.

Mismo resultado obtenemos al utilizar GLPK.

Para comprobar la consistencia de nuestro modelo, observamos qué soluciones obtenemos al cambiar los datos propuestos (sin modificar nada del modelo).

En caso de que en cada parada hubiera 6 personas (todas caben en un solo bus):

X	o	s1	s2	s3	d		F	o	s1	s2	s3	d
o	0	1	0	0	0		o	0	0	0	0	0
s1	0	0	1	0	0		s1	0	0	6	0	0
s2	0	0	0	1	0		s2	0	0	0	12	0
s3	0	0	0	0	1		s3	0	0	0	0	18
d	0	0	0	0	0		d	0	0	0	0	0

Coste: 220€

Observamos que, al caber todos los alumnos en un bus, no es necesaria más que una ruta que hace el recorrido o-s1-s2-s3-d.

Por último, suponemos que en la parada s2 no hay personas esperando:

X	o	s1	s2	s3	d		F	o	s1	s2	s3	d
o	0	1	0	0	0		o	0	0	0	0	0
s1	0	0	0	1	0		s1	0	0	0	6	0
s2	0	0	0	0	0		s2	0	0	0	0	0
s3	0	0	0	0	1		s3	0	0	0	0	12
d	0	0	0	0	0		d	0	0	0	0	0

Coste: 215€

Observamos que la ruta ya no pasa por la parada s2.

Las pruebas que implican añadir o quitar paradas las realizamos en el análisis de resultados del modelo 2, ya que si funciona en ese modelo podemos asumir que funciona también en este, y de esa manera no sobrecargamos la práctica. Además, incorporar nuevas paradas en el archivo Calc supone una pérdida de tiempo considerable.

Complejidad: En este problema contamos con 2*|Paradas|*|Paradas| variables y 12 restricciones, lo que lo convierte en un problema relativamente complejo. Al menos comparado con los problemas con los que estamos acostumbrados a trabajar.

LibreOffice Calc vs GLPK: Mientras que en Calc es mucho más tedioso introducir los datos y las restricciones, resulta mucho más fácil de entender los resultados obtenidos y las restricciones. Sin embargo GLPK puede resultar más complejo a simple vista, pero es más potente a la hora de introducir restricciones y la modificación de datos y parámetros es mucho más sencilla, sobre todo a la hora de añadir paradas.

3. MODELO 2

3.1. Descripción del modelo.

<u>Variables de decisión:</u>

Se añaden nuevas variables de decisión que se suman a las del Modelo 1:

g_{ij} : Variable binaria que relaciona cada alumno con la parada a la que está asignado.

Donde i es el conjunto de alumnos (A1, A2…) y j es cada una de las paradas.

<u>Función objetivo:</u>

Al tratarse de una modificación del modelo 1, la función objetivo es la misma.

<u>Restricciones:</u>

En la sección de datos se incluye una matriz que relaciona a los alumnos con las paradas a las que pueden ser asignados y otra que indica si los alumnos son hermanos. Utilizando los datos del enunciado, queda así:

parada_asignable

	o	s1	s2	s3	d
A1	0	1	0	0	0
A2	0	1	0	0	0
A3	0	1	0	0	0
A4	0	1	1	1	0
A5	0	1	1	1	0
A6	0	0	0	1	0
A7	0	0	0	1	0
A8	0	0	0	1	0

hermanos

	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	0	0	0	0	0	0	0
A2	0	0	0	0	0	0	0	0
A3	0	0	0	0	0	0	0	0
A4	0	0	0	0	1	0	0	0
A5	0	0	0	1	0	0	0	0
A6	0	0	0	0	0	0	0	0
A7	0	0	0	0	0	0	0	0
A8	0	0	0	0	0	0	0	0

Se añaden las siguientes restricciones a las del Modelo 1:

13. Los alumnos sólo pueden asignarse a paradas a las que pueden ser asignados:

$$parada\_asignable_{\mathrm{ij}} - g_{\mathrm{ij}} \geq 0$$

14. Cada alumno está asignado a una sola parada:

$$\sum_{j \in Paradas} g_{\mathrm{ij}} = 1 \quad \forall\, i \in \text{Alumnos}$$

15. Aquellos alumnos que sean hermanos son asignados a la misma parada:

$$hermanos_{\mathrm{ij}} * \left(g_{\mathrm{ip}} - g_{\mathrm{jp}}\right) = 0 \quad \forall\, i, j \in \text{Alumnos};\ \forall\, p \in \text{Paradas}$$

16. La variable de decisión g es no negativa y entera:

$$g_{\mathrm{ij}} \geq 0 \quad \forall\, \mathrm{i} \in \text{Alumnos};\ \forall\, j \in \text{Paradas}$$

3.2. Análisis de los resultados

Tras implementar el modelo en GLPK obtenemos el siguiente resultado:

X	o	s1	s2	s3	d		F	o	s1	s2	s3	d
o	0	1	1	1	0		o	0	0	0	0	0
s1	0	0	0	0	1		s1	0	0	0	0	3
s2	0	0	0	0	1		s2	0	0	0	0	2
s3	0	0	0	0	1		s3	0	0	0	0	3
d	0	0	0	0	0		d	0	0	0	0	0
A1	0	1	0	0	0							
A2	0	1	0	0	0							
A3	0	1	0	0	0							
A4	0	0	1	0	0							
A5	0	0	1	0	0							
A6	0	0	0	1	0							
A7	0	0	0	1	0							
A8	0	0	0	1	0							

Coste: 585€

Es decir, debido a que son hermanos, A4 y A5 se ubican en s2 y para recoger a todos los alumnos es necesario usar 3 autobuses. Veamos qué sucedería si no fueran hermanos:

X	o	s1	s2	s3	d		F	o	s1	s2	s3	d
o	0	1	0	1	0		o	0	0	0	0	0
s1	0	0	0	0	1		s1	0	0	0	0	4
s2	0	0	0	0	0		s2	0	0	0	0	0
s3	0	0	0	0	1		s3	0	0	0	0	4
d	0	0	0	0	0		d	0	0	0	0	0
A1	0	1	0	0	0							
A2	0	1	0	0	0							
A3	0	1	0	0	0							
A4	0	1	0	0	0							
A5	0	0	0	1	0							
A6	0	0	0	1	0							
A7	0	0	0	1	0							
A8	0	0	0	1	0							

Coste: 380€

Se observa que A4 y A5 son asignados cada uno a una parada distinta y se deja s2 vacía, lo que reduce bastante el coste.

Por último, en esta prueba se añadirán 2 paradas, con distancias inventadas y dos alumnos nuevos que serán hermanos de otros de los ya existentes. A continuación se exponen los datos:

Matriz de costes:

X	o	s1	s2	s3	s4	s5	d
o	99	8	10	10	7	3	99
s1	8	99	3	7	5	2	6
s2	10	3	99	5	6	3	7
s3	10	7	5	99	10	5	4
s4	7	5	6	10	99	12	4
s5	3	2	3	5	12	99	5
d	99	6	7	4	4	5	99

parada_asignable *hermanos*

	o	s1	s2	s3	s4	s5	d			A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	0	1	0	0	0	1	0		A1	0	0	0	0	0	0	0	0	0	0
A2	0	1	0	0	0	1	0		A2	0	0	0	0	0	0	0	1	0	0
A3	0	1	0	0	0	0	0		A3	0	0	0	0	0	0	0	0	0	1
A4	0	1	1	1	0	0	0		A4	0	0	0	0	0	0	0	0	1	0
A5	0	1	1	1	0	1	0		A5	0	0	0	0	0	0	0	0	0	0
A6	0	0	0	1	0	0	0		A6	0	0	0	0	0	0	0	0	0	0
A7	0	0	0	1	0	0	0		A7	0	0	0	0	0	0	0	0	0	0
A8	0	0	0	1	0	1	0		A8	0	1	0	0	0	0	0	0	0	0
A9	0	1	1	0	0	0	0		A9	0	0	0	1	0	0	0	0	0	0
A10	0	1	0	0	1	0	0		A10	0	0	1	0	0	0	0	0	0	0

Resultado:

X	o	s1	s2	s3	s4	s5	d		F	o	s1	s2	s3	s4	s5	d
o	0	1	0	1	0	1	0		o	0	0	0	0	0	0	0
s1	0	0	0	0	0	0	1		s1	0	0	0	0	0	0	4
s2	0	0	0	0	0	0	0		s2	0	0	0	0	0	0	0
s3	0	0	0	0	0	0	1		s3	0	0	0	0	0	0	2
s4	0	0	0	0	0	0	0		s4	0	0	0	0	0	0	0
s5	0	0	0	0	0	0	1		s5	0	0	0	0	0	0	4
d	0	0	0	0	0	0	0		d	0	0	0	0	0	0	0
A1	0	0	0	0	0	1	0									
A2	0	0	0	0	0	1	0									
A3	0	1	0	0	0	0	0									
A4	0	1	0	0	0	0	0									
A5	0	0	0	0	0	1	0									
A6	0	0	0	1	0	0	0									
A7	0	0	0	1	0	0	0									
A8	0	0	0	0	0	1	0									
A9	0	1	0	0	0	0	0									
A10	0	1	0	0	0	0	0									

Coste: 540€

Como se puede observar, quedan vacías las paradas s2 y s4. Los alumnos que son hermanos van a la misma parada: A2 y A8 a s5, A3 y A10 a s1, y A9 y A3 a s1.

La solución consiste en tres rutas: o-s1-d, o-s3-d y o-s5-d.

Complejidad: En este caso, contamos con más variable de decisión que en el anterior modelo: 2*|Paradas|*|Paradas| + |Alumnos|*|Alumnos| variables. Y, además, debe almacenar más información en memoria: matriz de hermanos y de paradas asignables.

4. CONCLUSIÓN

Como conclusión, creemos que es una práctica muy completa. Hemos quedado sorprendidos con la potencia de GLPK y todo lo que permite implementar. Hemos aprendido muy bien lo básico tanto de GLPK y Calc en cuanto a la Programación Lineal. Ha supuesto un verdadero reto realizar esta práctica, pero creemos que hemos conseguido un muy buen resultado.