Trabajo Individual de Optimización

Laboratorio de Modelización

Jose Antonio Lorencio Abril

31 de marzo de 2022

1. Problema

Un grupo de 9 músicos, que llamaremos, a riesgo de ser originales, A,B,C,D,E,F,G,H e I, debe tocar cada tarde un repertorio de 7 melodías, que siguiendo con nuestra originalidad denotaremos por 1,2,3,4,5,6 y 7. No todas las melodías necesitan de todos los músicos (véase la tabla adjunta). Por otro lado, cada músico recibe un salario proporcional al número de melodías en las que está presente (tocando o no su instrumento) desde la primera hasta la última en la que interviene. Ningún músico recibe sueldo alguno por las melodías en las que esté presente antes de la primera en la que sea necesario ni después de la última en la que sea necesario, pero sí por todas las demás. La siguiente tabla muestra las melodías en las que cada músico debe tocar su instrumento, así como el suelo que recibe por cada sinfonía en la que tenga necesariamente que estar presente.

Músico	A	B	C	D	E	F	G	H	I
Melodías	1,7	2,4,7	1,2,5,7	1,3,5	2,3,5,6	1,2,4,6,7	3,5,7	4,6	1,2,3
coste/melod	2	3	3	2	1	2	2	1	2

¿En qué orden debe el director tocar las melodías para minimizar el coste total de los salarios?

2. Solución 1

Inicialmente propuse una solución distinta, pero el profesor José Fernández me dijo que no era válida por usar restricciones de otros paradigmas de optimización. Esta solución, ya que está hecha, no la voy a deshechar, puede verse en la sección 3.

Entonces, pasemos a la resolución del problema.

Sea $X \in \mathcal{M}_{9\times 7}$ tal que

$$X_{i,j} = \begin{cases} 0, & \text{si el músico } i \text{ no toca en el momento } j \\ 1, & \text{si el músico } i \text{ toca en el momento } j \end{cases}$$

y sea $C_i, i = A, ..., I$ el vector de costes de los músicos, así como $Y \in \mathcal{M}_{7\times 7}$ tal que

$$Y_{i,j} = \begin{cases} 0, & \text{si la melodía } i \text{ no se toca en el momento } j \\ 1, & \text{si la melodía } i \text{ se toca en el momento } j \end{cases}.$$

Entonces, el problema de optimización es:

$$\begin{array}{lllr}
\text{mín} & \sum_{i=A}^{I} \sum_{j=1}^{7} C_i \cdot X_{i,j} & & \text{(OBJ)} \\
\text{s.a.} & \sum_{j=1}^{7} Y_{i,j} = 1 & \forall i = 1, ..., 7 & \text{(R1)} \\
& \sum_{i=1}^{7} Y_{i,j} = 1 & \forall j = 1, ..., 7 & \text{(R2)} \\
& X_{A,j} \geqslant Y_{1,j}, Y_{7,j} & \forall j = 1, ..., 7 & \text{(R3)} \\
& X_{B,j} \geqslant Y_{2,j}, Y_{4,j}, Y_{7,j} & \forall j = 1, ..., 7 & \text{(R4)} \\
& X_{C,j} \geqslant Y_{1,j}, Y_{2,j}, Y_{5,j}, Y_{7,j} & \forall j = 1, ..., 7 & \text{(R5)} \\
& X_{D,j} \geqslant Y_{1,j}, Y_{3,j}, Y_{5,j} & \forall j = 1, ..., 7 & \text{(R6)} \\
& X_{E,j} \geqslant Y_{2,j}, Y_{3,j}, Y_{5,j}, Y_{6,j} & \forall j = 1, ..., 7 & \text{(R7)} \\
& X_{F,j} \geqslant Y_{1,j}, Y_{2,j}, Y_{4,j}, Y_{6,j}, Y_{7,j} & \forall j = 1, ..., 7 & \text{(R8)} \\
& X_{G,j} \geqslant Y_{3,j}, Y_{5,j}, Y_{7,j} & \forall j = 1, ..., 7 & \text{(R9)} \\
& X_{H,j} \geqslant Y_{4,j}, Y_{6,j} & \forall j = 1, ..., 7 & \text{(R10)} \\
& X_{I,j} \geqslant Y_{1,j}, Y_{2,j}, Y_{3,j} & \forall j = 1, ..., 7 & \text{(R11)} \\
& X_{i,j} \geqslant X_{i,j-1} - (1 - X_{i,j+k}) & \forall i = A, ..., I, \forall j = 2, ..., 7-k, \forall k = 1, ..., 5 & \text{(R12)}
\end{array}$$

(OBJ) La función objetivo consiste en minimizar coste total que implica el orden para tocar las sinfonías elegido.

(R1) En cada momento solo se toca una sinfonía.

(R2) Cada sinfonía solo se toca una vez.

(R3)-(R11) Las restricciones de qué canción toca cada músico: si la canción i se toca en el momento j, entonces el músico que la toca debe estar en ese momento.

(R12) Restricción para que los músicos estén desde que se les necesita por primera vez, hasta que dejan de ser necesitados. La lógica es que si estamos en el momento j, si en el momento anterior el músico i debe estar, y en algún momento posterior también, entonces el músico i también debe estar en este momento. Podemos comprobar solo el momento anterior porque se van completando en cascada desde el momento 2 en adelante (el 1 no puede estar entre dos momentos).

2.1. Implementación en AMPL

2.1.1. .mod

```
set Musicos;
set Melodias;

param Coste{Musicos};
param Asign{Musicos,Melodias};

var X{Musicos, 1..7} binary;
var Y{Melodias, 1..7} binary;


minimize CosteTotal:
        sum {i in Musicos, j in 1..7} Coste[i]*X[i,j];

subject to R1{i in Melodias}:
        sum {j in 1..7} Y[i,j] = 1;

subject to R2{j in 1..7}:
        sum {i in Melodias} Y[i,j] = 1;

subject to R3a11{i in Musicos, j in Melodias, k in 1..7}:
        X[i,k]>=Asign[i,j]*Y[j,k];

subject to R12{i in Musicos, k in 1..5, j in 2..7-k}:
        X[i,j] >= X[i,j-1] - (1 - X[i,j+k]);
```

Como vemos, **CosteTotal** se corresponde con (OBJ), y las demás restricciones con las correspondientes a la formulación.

2.1.2. .dat

```
set Musicos := A B C D E F G H I;
set Melodias := 1 2 3 4 5 6 7;

param Coste := A 2
           B 3
           C 3
           D 2
           E 1
           F 2
           G 2
           H 1
           I 2;
param Asign :   1    2    3    4    5    6    7 :=
        A       1    0    0    0    0    0    1
```

B	0	1	0	1	0	0	1
C	1	1	0	0	1	0	1
D	1	0	1	0	1	0	0
E	0	1	1	0	1	1	0
F	1	1	0	1	0	1	1
G	0	0	1	0	1	0	1
H	0	0	0	1	0	1	0
I	1	1	1	0	0	0	0;

Aquí no hay mucho que comentar, solo que Asign indica las melodías que toca cada músico.

2.1.3. .run

```
#RESET THE AMPL ENVIROMENT

reset;

model 3-26-sol1.mod;

data 3-26-sol1.dat;

option solver ilogcp;

#SOLVE
solve;

#SHOW RESULTS
display X;
display Y;

for{i in 1..7}{
        for{j in 1..7}{
                if Y[i,j]=1 then{
                        printf "%d %d \n", i, j;
                }
        }
}

display CosteTotal;
```

3. Solución 2

Definamos las variables X_i, Y_i para $i \in A, ..., I$ como

$$X_i = j, \text{ si } i \text{ comienza a ser necesario en la melodía } j\text{-ésima},$$

$$Y_i = j, \text{ si } i \text{ deja de ser necesario en la melodía } j\text{-ésima.}$$

Definimos también Z_k, para $k \in 1, ..., 7$ mediante

$$Z_k = j, \text{ si la melodía } k \text{ se toca en } j\text{-ésimo lugar.}$$

Llamamos, además, C_i al coste por sinfonía en la que debe estar presente el músico i.

La formulación del problema es, entonces:

$$
\begin{array}{llll}
\text{mín} & \sum_{i=A}^{I} C_i \cdot (Y_i - X_i + 1) & & \text{(OBJ)} \\
\text{s.a.} & X_A \leqslant Z_1, Z_2 \leqslant Y_A & & \text{(R1)} \\
& X_B \leqslant Z_2, Z_4, Z_7 \leqslant Y_B & & \text{(R2)} \\
& X_C \leqslant Z_1, Z_2, Z_5, Z_7 \leqslant Y_C & & \text{(R3)} \\
& X_D \leqslant Z_1, Z_3, Z_5 \leqslant Y_D & & \text{(R4)} \\
& X_E \leqslant Z_2, Z_3, Z_5, Z_6 \leqslant Y_E & & \text{(R5)} \\
& X_F \leqslant Z_1, Z_2, Z_4, Z_6, Z_7 \leqslant Y_F & & \text{(R6)} \\
& X_G \leqslant Z_3, Z_5, Z_7 \leqslant Y_G & & \text{(R7)} \\
& X_H \leqslant Z_4, Z_6 \leqslant Y_H & & \text{(R8)} \\
& X_I \leqslant Z_1, Z_2, Z_3 \leqslant Y_I & & \text{(R9)} \\
& Z_k \neq Z_l & \forall k \neq l \in 1, ..., 7 & \text{(R10)} \\
& X_i, Y_i, Z_k \in 1, ..., 7 & \forall i = A, ..., I, \forall k = 1, ..., 7 & \text{(R11)}
\end{array}
$$

(OBJ) La función objetivo consiste en minimizar coste total que implica el orden para tocar las sinfonías elegido.

(R1) Las sinfonías 1 y 2 deben hacer después de que llegue A y antes de que se vaya. Las demás restricciones hasta la 9 siguen esta misma lógica.

(R10) Deben tocarse las 7 sinfonías, y se tocan en momentos distintos, por lo que no pueden repetirse valores.

(R11) Restricción del dominio de las variables.

3.1. Implementación en AMPL

3.1.1. .mod

```
set Musicos;
set Melodias;
```

```
param Coste{Musicos};
param Asign{Musicos,Melodias};

var X{i in Musicos} integer >= 1, <= 7;
var Y{i in Musicos} integer >= 1, <= 7;

var Z{j in Melodias} integer >= 1, <= 7;

minimize CosteTotal:
        sum {i in Musicos} Coste[i]*(Y[i]-X[i]+1);

subject to R19X{i in Musicos, j in Melodias}:
        X[i] <= Z[j] or Asign[i,j]<=0;

subject to R19Y{i in Musicos, j in Melodias}:
        Y[i] >= Z[j] or Asign[i,j]<=0;

subject to R10:
        alldiff {k in 1..7} Z[k];
```

Como vemos, **CosteTotal** se corresponde con (OBJ), **R19X** se corresponde con la parte izquierda de las restricciones (R1)-(R9) y **R19Y** con la derecha. Observemos el uso del **or**, que asegura que si i toca la melodía j, lo que equivale a que $asign[i,j] = 1$, entonces debe verificar la desigualdad, en caso contrario no nos importa si se satisface o no.

Por último, **R10** hace exactamente lo mismo que nuestra (R10), y (R11) se modela en la definición de las variables.

3.1.2. .dat

```
set Musicos := A B C D E F G H I;
set Melodias := 1 2 3 4 5 6 7;

param Coste := A 2
            B 3
            C 3
            D 2
            E 1
            F 2
            G 2
            H 1
            I 2;
param Asign :   1   2   3   4   5   6   7 :=
        A       1   0   0   0   0   0   1
        B       0   1   0   1   0   0   1
        C       1   1   0   0   1   0   1
```

D	1	0	1	0	1	0	0
E	0	1	1	0	1	1	0
F	1	1	0	1	0	1	1
G	0	0	1	0	1	0	1
H	0	0	0	1	0	1	0
I	1	1	1	0	0	0	0;

Aquí no hay mucho que comentar, solo que Asign indica las melodías que toca cada músico.

3.1.3. .run

```
#RESET THE AMPL ENVIROMENT

reset;

model 3-26.mod;

data 3-26.dat;

option solver ilogcp;

#SOLVE
solve;

#SHOW RESULTS
display X;
display Y;
display Z;
display CosteTotal;
```

Lo interesante que encontramos aquí es que usamos el solver **ilogcp**, que permite utilizar operaciones lógicas en las restricciones, como el **or**, y restricciones de unicidad de valores, como el **alldiff**.

4. Resultados

4.1. Solución 1

Obtenemos el resultado (la salida imprimida con el for):

```
1 3
2 5
3 1
4 6
```

```
5  2
6  7
7  4
CosteTotal = 68
```

Por tanto, las canciones se tocarían en el orden: 3,5,1,6,2,7,4; obteniendo un coste de 68.

4.2. Solución 2

Obtenemos la ordenación:

```
Z [*] :=
1   3
2   5
3   1
4   6
5   2
6   7
7   4
;

CosteTotal = 68
```

Es decir, que el orden de interpretación de las melodías debe ser 3,5,1,6,2,7,4; y el coste mínimo es de 68.

5. Conclusiones

Como vemos, las dos formulaciones proporcionan las mismas ordenaciones. No obstante, la primera formulación es preferible, pues da uso únicamente a restricciones estrictamente de optimización lineal, mientras que el segundo enfoque usa restricciones lógicas y restrictivas. Esta segunda formulación quizás sea mejor desde el punto de vista de la comunicación del modelo a un interesado menos familiarizado con las matemáticas, pues las restricciones son muy semánticas y es sencillo comprender el modelo.