

Programación con restricciones- Práctica 7

Una compañía de transportes dispone de N camiones, con los que tiene realizar D viajes. Llamamos **DESTINOS** al conjunto con el rango 1..D. Disponemos además de un array **d** de tamaño D x 4. Cada fila tiene los datos de un viaje: un índice de fila, el número de camiones necesarios, y el tiempo en días (tiempo total, ida+vueltas).

Queremos organizar en qué día debe realizarse cada viaje, para que se acabe lo antes posible con todos los destinos, asegurando siempre que hay caminos suficientes para el viaje requerido.

Vamos a utilizar la restricción *cumulative* para lograr este objetivo.

- 1) Vamos a declarar el vector **t** de tamaño **DESTINOS** que contenga el tiempo requerido cada viaje. Este vector será una copia exacta de la tercera columna de **d**. Declarar **t** y asegurarse de que, en efecto, corresponde con la tercera columna de **d**.
- 2) Ahora declaramos el array **s** de tamaño **DESTINOS**, que tiene el día en el que partirá cada viaje
 - a. Declarar el array
 - b. La restricción que asegura que son todos valores ≥ 0
- 3) Finalmente tenemos que declarar el vector **r** de tamaño **DESTINOS** con los camiones que necesita cada viaje
 - a. Declarar el array
 - b. La restricción que hace que contenga la segunda columna de **d**
- 4) Añadir la restricción *cumulative* correspondiente y el *solve satisfy*. La salida será del estilo (puede variar, según las restricciones):

```
t= array1d(1..13 ,[4, 3, 4, 5, 3, 2, 2, 2, 6, 2, 2, 2, 2]);
s = array1d(1..13 ,[11, 8, 0, 0, 8, 13, 17, 6, 0, 11, 15, 15, 4]);
r = array1d(1..13 ,[1, 2, 3, 2, 4, 5, 2, 6, 1, 4, 3, 2, 3]);
```

- 5) [2] Ahora queremos minimizar el tiempo. Para esto declararemos:
 - a. Un vector de **DESTINOS** con variables de 1 a SUM_T de nombre **a**, con SUM_T la suma de lo que tardan todos los viajes
 - b. Una restricción que asegura que **a[i]** es el momento en que acaba la tarea i

La salida será de la forma:

```
a = array1d(1..13 ,[15, 11, 4, 5, 11, 15, 19, 8, 6, 13, 17, 17, 6]);
```

- 6) Ahora :
 - a. Declaramos una variable entera **fin**
 - b. Una restricción que asegura que **fin** es el máximo de **a**
 - c. Comentamos el *solve satisfy*, y añadimos la restricción que hace que el tiempo que se tarda en completar todos los viajes es mínimo.

Posible salida:

```
t = array1d(1..13,[4, 3, 4, 5, 3, 2, 2, 2, 6, 2, 2, 2, 2]);
s = array1d(1..13,[13, 8, 0, 0, 8, 13, 15, 6, 0, 11, 15, 11, 4]);
r = array1d(1..13,[1, 2, 3, 2, 4, 5, 2, 6, 1, 4, 3, 2, 3]);
a = array1d(1..13,[17, 11, 4, 5, 11, 15, 17, 8, 6, 13, 17, 13, 6]);
fin = 17;
```

- 7) Añadir un constraint que asegure que los destinos (cuarta columna de d) forman un circuito hamiltoniano.
- 8) [2] Queremos mostrar la salida de forma que indique para cada viaje el mes de comienzo y los meses que ocupa en forma de cronograma.

Fin: 17

																		*	*	*	*
										*	*	*									
*	*	*	*																		
*	*	*	*	*																	
										*	*	*									
																		*	*		
																				*	*
								*	*												
*	*	*	*	*	*																
													*	*							
																		*	*		
				*	*								*	*							