

Técnicas de los Sistemas Inteligentes

Práctica 2: Resolución de problemas con MiniZinc

David Muñoz Sánchez
07256819C - Grupo A1

14 de mayo de 2023

Índice

1. Ejercicio 1	2
2. Ejercicio 2	2
3. Ejercicio 3	3
4. Ejercicio 4	4
5. Ejercicio 5	5

1. Ejercicio 1

Antes de mostrar los distintos resultados notar, que si un conjunto contiene, por ejemplo, dos 1's, son distintos elementos dentro del conjunto, para seguir la definición matemática de conjunto (no hay elementos repetidos).

Para la tercera tabla, el primer resultado es aparentemente correcto, pero con la ejecución de MiniZinc hay un problema. No termina en un tiempo razonable como lo hacen los otros dos resultados. No obstante, me dio un resultado con diferencia 1 y se sabe que es el mínimo porque no se puede alcanzar la igualdad en suma entre ambos conjuntos.

Estas son las tablas que se piden completar:

Conjunto S	Conjunto S1	Conjunto S2	Suma de S1 (y de S2)
{1,1,2}	{1,1}	{2}	2
{1,2,3,4,5,6}	UNS.	UNS.	UNS.
{1,1,2,4,4,5,6,7,8}	{1,1,2,4,5,6}	{4,7,8}	19

Tabla 1: Ejercicio 1.a

Conjunto S	Conjunto S1	Conjunto S2	Suma de S1 (y de S2)	Número de soluciones
{1,1,2}	{1,1}	{2}	2	2
{1,2,3,4,5,6}	UNS.	UNS.	UNS.	UNS.
{1,1,2,4,4,5,6,7,8}	{1,1,2,4,5,6}	{4,7,8}	19	14

Tabla 2: Ejercicio 1.b

CONJUNTO S	S1	S2	SUM S1	SUM S2	DIFF
{99,14,82,47,82,50,77,45,23,92,52,90,46,57,29,25,74,25,30,70}	{99,82,47,82,77,23,92,52}	{14,50,45,90,46,57,29,25,74,25,30,70}	554	555	1
{35,82,97,30,38,78,18,39,35,93,49,47,30,57,43,21,92,51,89,80}	{82,97,30,38,78,18,39,93,47,30}	{35,35,49,57,43,21,92,51,89,80}	552	552	0
{60,8,11,2,49,69,87,15,62,72,89,14,76,63,45,92,74,80,11,2}	{60,2,49,69,87,62,72,89}	{8,11,15,14,76,63,45,92,74,80,11,2}	490	491	1

Tabla 3: Ejercicio 1.c

2. Ejercicio 2

Estas son las tablas pedidas:

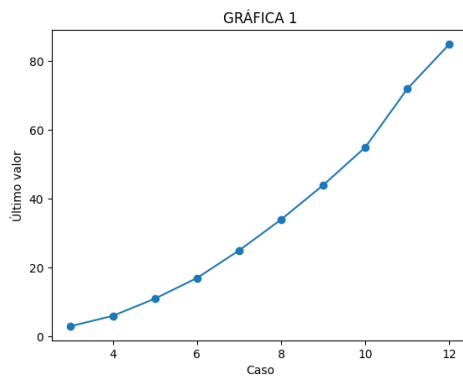
#marcas	Posición de las marcas	Última marca x_m	Runtime
3	{0,1,3}	3	357ms
5	{0,2,7,8,11}	11	243ms
7	{0,2,3,10,16,21,25}	25	237ms
10	{0,1,6,10,23,26,34,41,53,55}	55	5s 463ms
12	{0,2,6,24,29,40,43,55,68,75,76,85}	85	8min 33s

Tabla 4: Ejercicio 2

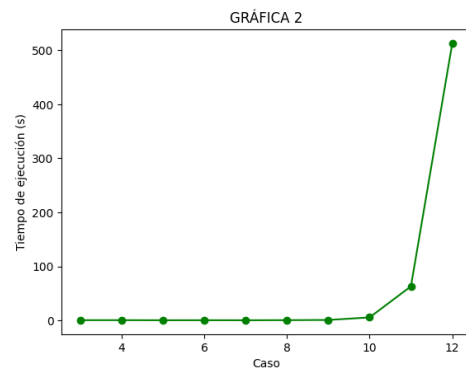
DATOS GRÁFICA		
Caso	Última marca	Tiempo ejecución (s)
m = 3	3	0.357
m = 4	6	0.394
m = 5	11	0.243
m = 6	17	0.310
m = 7	25	0.237
m = 8	34	0.414
m = 9	44	0.758
m = 10	55	5.463
m = 11	72	63
m = 12	85	513

Tabla 5: Datos para la gráfica del ejercicio 2.

Estas son las gráficas que se piden:



(a) Caso frente a último valor.



(b) Caso frente a tiempo de ejecución.

Figura 1: Gráficas ejercicio 2.

3. Ejercicio 3

En este ejercicio se obtienen 8 soluciones válidas que satisfacen todas las restricciones. Por la forma de implementarlo, no se han obtenido soluciones simétricas, dado que las 8 soluciones que se obtienen son horarios diferentes. Se entiende por soluciones simétricas aquellas que semánticamente son iguales pero, MiniZinc, por la forma de implementación, las considera distintas. Además, se ha modificado la salida para hacer más comprensible el horario, mostrando por cada día de la semana y hora de la mañana, la asignatura que se imparte y el profesor que lo hace. No he visto necesario imprimir el bloque de la asignatura, dado que a lo sumo tienen dos bloques de 1 o 2 horas, y bastaría con comprobar que si una asignatura, por ejemplo, tiene 2 bloques de 2 horas, no hay más de 2 pares de esa asignatura en el horario.

No obstante, una fuente clara de simetría sería la hora de tutoría/recreo. En mi caso se ha modelado asignando el número 10 independientemente del día de la semana. Si por ejemplo, hubiésemos asignado el 10 a la del lunes, el 11 a la del martes y así sucesivamente, hubiera habido soluciones simétricas en el sentido de que el 11 por ejemplo, podría ir en martes pero también en los demás días.

También podría ser una fuente de simetría el hecho de que hay varias asignaturas que se dan en bloques de 2 horas seguidas. En mi caso, esto se ha modelado con un array que indica, por asignatura, de cuantas horas son sus bloques, y a la hora de asignar al horario, se tiene en cuenta y si el bloque es de 2 horas, la siguiente directamente se asigna a la misma asignatura. Si en vez de así, se hubiera hecho modelando bloque 1 y bloque 2 (para este caso), en una asignatura que, por ejemplo, se de en las dos primeras horas, cabría poner bloque 1 primero y bloque 2 después, y viceversa, siendo en esencia, lo mismo, ya que es la misma asignatura.

4. Ejercicio 4

Los diagrama de Gantt se han creado a partir de un código Python usando Pandas y Matplotlib. Se adjunta dicho código en un fichero externo. Para realizar el DataFrame de Pandas necesario, se tienen en cuenta todas las variables que se muestran al minimizar:

1. Un array con los días de inicio.
2. Un array con los días de finalización.
3. Un array con el trabajador que efectúa cada tarea.
4. Una variable donde se almacena el tiempo total.
5. Un array de booleanos para indicar si una tarea se hace con auxiliar o no (para el caso de la versión con auxiliar).

Sin trabajador de apoyo, la obra tarda como mínimo 17 días en construirse. El diagrama de Gantt es el siguiente:

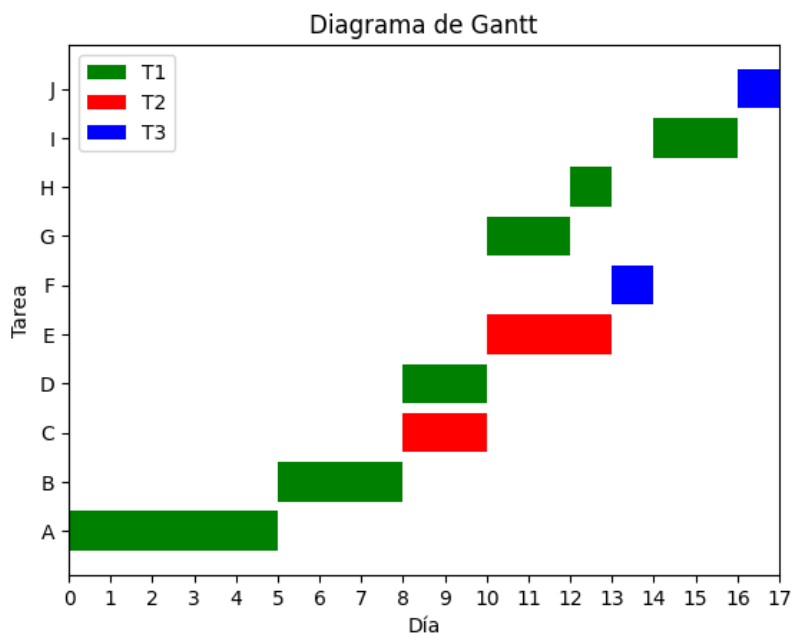


Figura 2: Diagrama de Gantt primer apartado.

Con trabajador de apoyo, la obra tarda como mínimo 13 días en construirse. El diagrama de Gantt es el siguiente (siendo T4 el trabajador de apoyo):

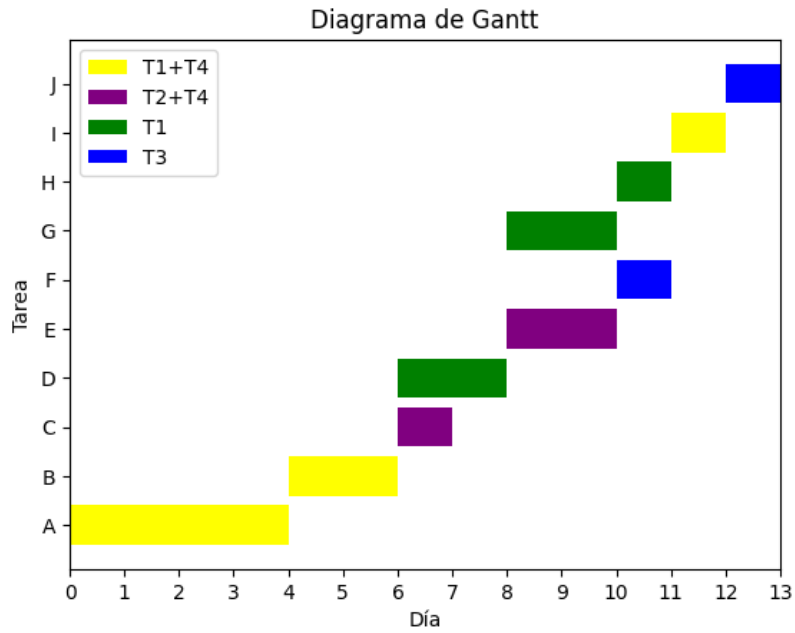


Figura 3: Diagrama de Gantt segundo apartado.

5. Ejercicio 5

Para la generación de la matriz de adyacencia, se ha usado el código Python proporcionado junto con las semillas 0, 1 y 2. Las sumas de la tabla aparecen en orden para dichas semillas. Se aporta un txt de apoyo con todas las matrices generadas por el código Python para su prueba, a fin de no hacer el código demasiado extenso.

Nº DE NODOS	MÍNIMO Nº DE NODOS EN EL CONJUNTO	RUNTIME(MS)
10	$\frac{2+2+2}{3} = 2$	$\frac{104+105+203}{3} \approx 137,33$
20	$\frac{3+3+4}{3} \approx 3,33$	$\frac{108+101+102}{3} \approx 103,66$
30	$\frac{4+5+5}{3} \approx 4,67$	$\frac{118+132+122}{3} = 124$
40	$\frac{6+6+6}{3} = 6$	$\frac{334+437+321}{3} = 364$
50	$\frac{8+9+8}{3} \approx 8,33$	$\frac{2845+4435+2983}{3} = 3421$
60	$\frac{9+10+9}{3} \approx 9,33$	$\frac{48590+60002+42384}{3} \approx 50325,33$

Tabla 6: Tabla ejercicio 5.

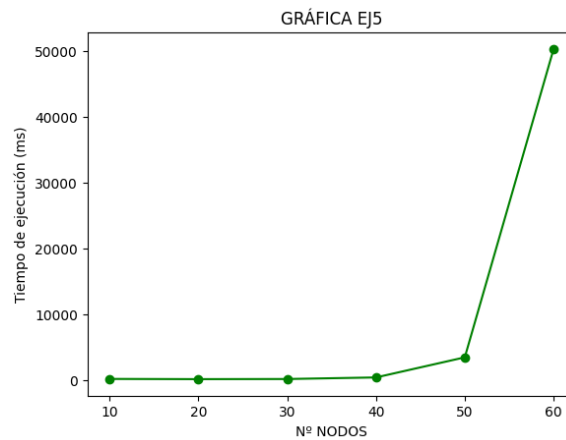


Figura 4: Gráfica para mostrar la no escalabilidad.

Como se puede observar en la gráfica, la tendencia del tiempo de ejecución hace ver que para un número de

nodos amplio, el problema tendrá un tiempo de ejecución bastante alto, por lo tanto, no es escalable. La matriz de adyacencia conforme sube el número de nodos se va haciendo cada vez más y más grande.