

Memoria Práctica 2: Resolución de Problemas de Satisfacción de Restricciones

Fernando Luque de la Torre

5 de mayo de 2022

Índice

1	Problema de las Monedas	3
1.1	Apartado a.	3
1.2	Apartado b.	3
1.3	Apartado c.	3
1.4	Apartado d.	3
2	Problema de los Horarios	4
2.1	Horarios obtenidos	4
2.2	Preguntas	4
3	Problema Lógico	5
4	Problema de asignación de tareas	5
4.1	Preguntas	5
5	Problema de coloreado de grafos	7
5.1	Resultados	7
5.2	Preguntas	7

1. Problema de las Monedas

1.1. Apartado a.

Importe	Primera solución y número de monedas	Nº de soluciones	Runtime (s)
0,17 €	{17,0,0,0,0,0,0} (17)	28	0,29
1,43 €	{143,0,0,0,0,0,0} (143)	17952	3,06
2,35 €	{235,0,0,0,0,0,0} (235)	150824	24,86
4,99 €	{499,0,0,0,0,0,0} (499)	?	?

1.2. Apartado b.

Importe	Primera solución y número de monedas	Nº de soluciones	Runtime (s)
0,17 €	{17,0,0,0,0,0,0} (17)	28	0,29
1,43 €	{43,0,0,0,0,0,1} (44)	284	0,37
2,35 €	{35,0,0,0,0,0,1} (36)	162	0,35
4,99 €	{99,0,0,0,0,0,2} (101)	4366	0,95

1.3. Apartado c.

Importe	Solución óptima	Nº de soluciones	Runtime (s)
0,17 €	{0,1,1,1,0,0,0} (3)	15	0,28
1,43 €	{1,1,0,0,2,0,1} (5)	40	0,29
2,35 €	{0,0,1,1,1,0,1} (4)	32	0,29
4,99 €	{0,2,1,0,2,1,0} (8)	94	0,34

1.4. Apartado d.

¿Qué ocurriría si, usando la codificación (a) para encontrar todas las soluciones, el importe buscado es mucho mayor? Como podemos ver, ya para un importe de 4,99€ la codificación a) ejecutada en mi PC, no es capaz de alcanzar todas las soluciones posibles ya que se queda bloqueado el propio MiniZinc. Por tanto, es obvio pensar que para importes mayores (del orden de miles o millones de euros), la explosión combinatoria de este problema es bestial, por lo que sería inviable encontrar todas las soluciones en un tiempo razonable.

¿Se podría encontrar alguna solución (usando la codificación de (a) o cualquier otra) de este problema con un importe del orden de los millones de euros? Realmente la codificación del problema permite encontrar soluciones e incluso implementando un método de búsqueda correcto (que suponemos que hace Gecode) encontrar una solución óptima para valores grandes. El problema, como he dicho en la

primera pregunta, viene cuando se trata de explorar TODAS las soluciones posibles ya que este número de soluciones escala muy rápido en función del importe.

Sin embargo, sería una buena idea por ejemplo, la propuesta en el apartado b): asignar el importe de euros de forma automática al valor más grande de moneda, es decir, si por ejemplo tenemos 1000,32€, se asignan 1000€ en 500 monedas de 2€ y el resto ya se deja que minizinc lo minimice.

2. Problema de los Horarios

2.1. Horarios obtenidos

Primero:

Hora\Dia	L	M	X	J	V
1	A4	A4	A8	A5	A5
2	A4	A4	A8	A5	A5
3	A9	A7	A6	A2	A6
4	RECREO	RECREO	RECREO	RECREO	RECREO
5	A1	A1	A3	A3	A2
6	A1	A1	A3	A3	A7

Segundo:

Hora\Día	L	M	X	J	V
1	A4	A4	A8	A5	A5
2	A4	A4	A8	A5	A5
3	A9	A7	A6	A2	A6
4	RECREO	RECREO	RECREO	RECREO	RECREO
5	A1	A1	A3	A3	A7
6	A1	A1	A3	A3	A2

2.2. Preguntas

¿Cuál es el número de soluciones válidas obtenidas? ¿Existen soluciones simétricas? Como podemos ver, para las restricciones establecidas, son 2 las soluciones obtenidas. De hecho, son prácticamente la misma solución pero permite intercambiar las dos últimas horas del viernes, por lo que, no, las únicas dos soluciones obtenidas son diferentes, no son simétricas

¿Cómo se podrían evitar y cuál es el número de soluciones (no simétricas) obtenido? Explique cómo se consigue la rotura de simetrías (variables y/o restricciones utilizadas para ello), y entregue la solución MZN sin simetrías.

En mi caso, la rotura de simetrías no es un problema que haya tenido que abordar directamente ya que la propia codificación que he escogido desde un principio ha sido la de, en la matriz que representa el horario, representar en cada hueco el número correspondiente

a la asignatura que se imparte en ese tramo horario. De igual modo hago con el recreo, el 0 me codificaba el recreo y es por eso que no hay 5! combinaciones diferentes de recreo. Es por eso que, al no diferenciar entre diferentes bloques de asignaturas y codificar todos los bloques por igual, directamente me he deshecho de las simetrías.

3. Problema Lógico

Antes, introduzco una tabla para aclarar la codificación que he utilizado:

Característica\Codificación	1	2	3	4	5
Región	Andaluz	Vasco	Catalán	Gallego	Navarro
Profesión	Pintor	Escultor	Diplomático	Violinista	Médico
Animal	Perro	Zorro	Caballo	Caracoles	CEBRA
Bebida	Té	Café	Leche	Zumo	AGUA
Color casa	Rojo	Azul	Blanca	Verde	Amarilla

Esta tabla indica que, el valor j de la característica i se sitúa en la posición (i, j) . Es decir, si la región (característica i) Andaluz se sitúa en la columna j significa que se codifica con el entero j (en este caso con el 1).

De esta forma, la solución son 5 vectores de características donde cada posición $i \in \{1, \dots, 5\}$ indica el valor de la característica para la persona que vive en la casa i .

De esta forma la solución queda así:

Característica\Casa	Izquierda	Segunda	Centro	Cuarta	Derecha
Región	Andaluz	Navarro	Vasco	Catalán	Gallego
Profesión	Diplomático	Médico	Escultor	Violinista	Pintor
Animal	Zorro	Caballo	Caracoles	Perro	CEBRA
Bebida	AGUA	Té	Leche	Zumo	Café
Color Casa	Amarilla	Azul	Roja	Blanca	Verde

Por lo que, la cebra vive con el gallego que es pintor y bebe café en la última casa (la de la derecha) que es verde.

Por otro lado, el que bebe agua es el andaluz, que es diplomático, tiene un zorro y vive en la primera casa (la de la izquierda) que es amarilla.

4. Problema de asignación de tareas

4.1. Preguntas

¿Cuál es la duración mínima de la construcción de la casa? La construcción de la casa se realizaría como mínimo en 12 días usando los 3 trabajadores de la siguiente forma (En la fila se indica la tarea y en la columna se indican cada uno de los días del proceso):

	1	2	3	4	5	6	7	8	9	10	11	12
Tarea A	1	1	1	1								
Tarea B					1	1	1					
Tarea C								2				
Tarea D								1	1			
Tarea E										2	2	
Tarea F										3		
Tarea G										1		
Tarea H					2	2	2					
Tarea I											1	1

Considere ahora que se dispone de un cuarto trabajador de refuerzo. ¿Qué tiempo tardarán, como mínimo, los cuatro trabajadores en finalizar la construcción? Con la utilización del trabajador de refuerzo la construcción de la casa se reduciría a 8 días con la siguiente asignación. En azul el primer trabajador con el trabajador extra, en naranja el tercer trabajador con el trabajador extra. El resto de colores el trabajador indicado sin trabajador extra:

	1	2	3	4	5	6	7	8
Tarea A	1	1						
Tarea B			1					
Tarea C				2				
Tarea D				1	1			
Tarea E						2	2	
Tarea F						3		
Tarea G							3	
Tarea H						1		
Tarea I							1	1

5. Problema de coloreado de grafos

5.1. Resultados

Se incluye la tabla solicitada sobre las 3 ejecuciones de cada grafo:

Tamaño	Nº de colores mínimo medio	Runtime medio(s)	Mínimo runtime (s)
N=4, M=6	2,667	0,2793	0,277
N=6, M=15	4,667	0,293	0,292
N=8, M=28	6,333	0,331	0,327
N=10, M=45	8	0,686	0,569
N=12, M=66	9	20,491	0,867
N=14, M=91	?	?	>= 30 min

Para cada tamaño de grafo se han generado 3 grafos aleatorios con semillas del 0 al 2 y se han obtenido los siguientes resultados.

Como podemos ver, la ejecución del programa con el grafo de mayor tamaño no ha podido completarse dado su gran tamaño.

Además un hecho a destacar es la gran diferencia que se pueden dar entre grafos del mismo tamaño, tal y como ocurre en el penúltimo grafo, donde vemos que la media de runtime es de 20,49s sin embargo el tiempo de ejecución mínimo es de 0,867s. Esto se debe a que para el grafo con semilla 0, la ejecución ha tardado 55s y para el de semilla 1 ha tardado menos de 1s.

5.2. Preguntas

En base a los resultados obtenidos, ¿diría que este problema es escalable, es decir, se puede abordar su resolución en grafos de un tamaño considerable?

Razone su respuesta. Según los resultados obtenidos el problema, al menos con esta codificación, no sería escalable ya que para el último grafo de 91 aristas no es capaz de encontrar una solución en media hora de ejecución. Por tanto, si quisiéramos hacer este problema más escalable sería buena idea incluir algunas restricciones o cambiar otras presentes para que se optimice más. Otra idea, sería eliminar con algún tipo de preprocesado en otro lenguaje quizás, las aristas repetidas de tipo $(i, j) = (j, i)$, lo cual reduciría en gran medida el tiempo de ejecución.