Técnicas de los Sistemas Inteligentes

Práctica 2: Satisfacción de restricciones con MiniZinc

Curso 2021-2022

Pedro Bedmar López - 75935296Z
pedrobedmar@correo.ugr.es
Grado en Ingeniería Informática

---

1. Problema de las monedas

En este ejercicio se pretende resolver el problema de que dada una cantidad de dinero, se devuelve en monedas de 1, 2, 5, 10, 20 y 50 céntimos y en monedas de 1 y 2 euros.

1.1. Apartado a)

En este apartado calculamos cualquier solución válida, o sea, cualquier combinación de esas monedas que sume exactamente el importe introducido. Al no existir ninguna restricción aparte del uso de esas monedas específicas, el número de soluciones es muy alto y crece exponencialmente (como también lo hace el tiempo de ejecución).

En la siguiente tabla se muestran los resultados de ejecución en 4 situaciones. La solución encontrada se expresa como un vector [c1,c2,c5,c10,c20,c50,e1,e2], donde cx representa la cantidad de monedas de x céntimos y ex la cantidad de monedas de x euros.

Importe	Primera solución encontrada y número de monedas de la misma	Número total de soluciones	Runtime (en segundos)
0.17€	[17,0,0,0,0,0,0,0] $->$ 17 monedas	28	0.286
1.43€	[143,0,0,0,0,0,0,0] $->$ 143 monedas	17952	2.692
2.35€	[235,0,0,0,0,0,0,0] $->$ 235 monedas	150824	25.764
4.99€	[499,0,0,0,0,0,0,0] $->$ 499 monedas	6224452	2002

Tabla 1: Resultados del apartado a) del problema de las monedas.

1.2. Apartado b)

En este caso, añadimos restricciones extra que impiden que con monedas de céntimo podamos representar un euro o más. Debido a esto el espacio de búsqueda se va a reducir y el tiempo de ejecución por tanto también disminuye.

Importe	Primera solución encontrada y número de monedas de la misma	Número total de soluciones	Runtime (en segundos)
0.17€	[17,0,0,0,0,0,0,0] ->17 monedas	28	0.342
1.43€	[43,0,0,0,0,0,1,0] ->143 monedas	284	0.311
2.35€	[35,0,0,0,0,0,2,0] ->235 monedas	324	0.412
4.99€	[99,0,0,0,0,0,4,0] ->499 monedas	13098	2.200

Tabla 2: Resultados del apartado b) del problema de las monedas

1.3. Apartado c)

En este caso observamos que los tiempos de ejecución son menores o iguales que en el caso anterior.

Importe	Primera solución encontrada y número de monedas de la misma	Runtime (en segundos)
0.17€	[0,1,1,1,0,0,0,0] ->3 monedas	0.283
1.43€	[1,1,0,0,2,0,1,0] ->5 monedas	0.327
2.35€	[0,0,1,1,1,0,0,1] ->4 monedas	0.338
4.99€	[0,2,1,0,2,1,0,2] ->8 monedas	0.376

Tabla 3: Resultados del apartado c) del problema de las monedas

1.4. Apartado d)

¿Qué ocurriría si, usando la codificación (a) para encontrar todas las soluciones, el importe buscado es mucho mayor?

Ocurre que el espacio de búsqueda crece exponencialmente, y de la misma forma lo hace el tiempo de ejecución, por lo tanto llega un punto en el que el problema es inabarcable.

¿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?

Sí, una codificación donde únicamente existan monedas de un céntimo sería más eficiente y podría resolver el problema con tiempos de ejecución menores.

2. **Problema de los horarios**

3. **Problema lógico**

4. **Problema de la asignación de tareas**

5. **Problema del coloreado de grafos**