



Técnicas de los Sistemas Inteligentes

Enunciado de la Práctica 2

En esta práctica se pide al estudiante resolver la siguiente relación de problemas usando MiniZinc. Cada ejercicio se resolverá en un **fichero MZN por separado**. La solución entregada debe ser **correcta y completa**, es decir, todas las soluciones válidas son admitidas por la codificación entregada y ésta no admite ninguna solución no válida. **El código debe estar perfectamente comentado, explicando y justificando la solución presentada, variables y restricciones usadas para codificar cada problema, y cualquier otro aspecto relativo a las decisiones tomadas durante el proceso de codificación. Un código cuyos comentarios sean, a criterio del profesor, extremadamente deficientes podrá invalidar total o parcialmente aquellos problemas a los que afecte.**

Nota: Los problemas con **errores sintácticos** en MiniZinc automáticamente califican con una puntuación de **0 (cero) puntos**.

Para la resolución de todos los problemas se recomienda usar el **solver Gecode**, que viene en el IDE de Minizinc por defecto.

Adicionalmente, se entregará una **memoria en formato PDF** donde se describa las respuestas a las preguntas planteadas en cada ejercicio de forma razonada (no se incluirá código; éste ya se entrega en el fichero MZN, el cual debe estar adecuadamente comentado). Estas respuestas deben abordar **conceptualmente** los temas de esta práctica.

Cada problema tiene una **puntuación máxima de dos puntos**, los cuales contemplan tanto la codificación MZN como la respuesta razonada en la memoria. La entrega consistirá en un **fichero ZIP** que contenga la memoria de la práctica (PDF) y los 5 ficheros MZN de cada problema.

Fecha de entrega: 08/05/2020 23:59

Modo de entrega: A través de la plataforma **Prado**, en el enlace correspondiente.



1. Problema de las monedas:

- a. Se desea encontrar un conjunto de monedas cuyo importe sea exactamente una cantidad dada. Para ello, se usarán las monedas de 1, 2, 5, 10, 20 y 50 céntimos, así como las de 1 y 2 euros. Codifique una solución en MZN que calcule cualquier solución válida, y complete la siguiente tabla usando esta codificación:

Importe	Primera solución encontrada y número de monedas en la misma	Número total de soluciones	Runtime (en segundos)
0,17 €			
1,43 €			
2,35 €			
4,99 €			

Nota (1): Para calcular todas las soluciones (en lugar de una sola), se debe desmarcar la casilla “*Stop after this many solutions (uncheck for all)*” en el menú “*Show configurator editor*”. Igualmente, para imprimir el número total de soluciones, debe activarse la opción “*Output solving statistics*” en este menú. El número de soluciones aparece en la consola en el campo “*%%%mzn-stat: solutions*” (cuando se ejecuta con el solver Gecode).

Nota (2): Para el *runtime*, se recomienda usar el valor reportado por MiniZinc en la última línea “*Finished in ...*” (ya que el valor reportado en las estadísticas como “*solveTime*” no incluye todo el proceso de resolución y, a veces, es algo más inestable por la forma en la que es medido).

- b. Modifique la versión anterior de forma que la parte entera de los importes sea asignada únicamente a monedas de uno y dos euros. Por ejemplo, para el importe de 1,43€ se deberá forzosamente asignar una moneda de 1€ y completar los 43 céntimos restantes con monedas de 1, 2, 5, 10, 20 o 50 céntimos. Complete la tabla anterior con esta nueva codificación.
- c. Realice una nueva modificación sobre la versión anterior para que el conjunto de monedas encontrados sea mínimo, es decir, que tenga el menor número de monedas posible. Para ello, se recomienda comenzar por la parte de satisfacción y guardar en una variable la “cantidad” de monedas de la solución encontrada, y tras esto resolver la parte de optimización usando la sentencia: `solve minimize <variable>`

Genere una nueva tabla de resultados con una columna de “solución óptima” donde se indique la solución óptima encontrada para los importes anteriores y la cantidad de monedas de la misma, y otra columna de runtime.

- d. Responda razonadamente a las siguientes cuestiones: ¿qué ocurriría si, usando la codificación (a) para encontrar **todas** las soluciones, el importe buscado es mucho mayor? ¿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? En caso afirmativo, ¿cuál podría ser una estrategia prometedora?

El fichero MZN que se entrega en este ejercicio es el del apartado (c).



2. Problema de los horarios. Se desea encontrar una asignación de horarios que satisfaga las siguientes condiciones:

- Existe un aula disponible en seis franjas consecutivas de 1h (por ejemplo, de 8:00 a 14:00) de lunes a viernes.
- Existen nueve asignaturas (A1..A9). El número de horas semanales de cada asignatura se detalla en la siguiente tabla:

A1	A2	A3	A4	A5	A6	A7	A8	A9
4 hs	2 hs	4 hs	4 hs	4 hs	2 hs	2 hs	2 hs	1 hr

- Las asignaturas {A1,A3,A4,A5,A8} deben impartirse en bloques de 2h consecutivas, mientras que el resto, es decir {A2,A6,A7,A9}, se imparten en bloques de 1h.
- En cada día de la semana solo se puede impartir, como máximo, un bloque de cada asignatura.
- El profesor/a de cada asignatura es el siguiente: Prof1={A1,A3}; Prof2={A4,A5}; Prof3={A6,A9}; Prof4={A2,A7,A8}.
- Cada profesor solo puede impartir un bloque de alguna de sus asignaturas cada día, excepto Prof4 (que puede impartir más de una).
- La cuarta franja horaria debe reservarse para el recreo ("NA"); es decir, no asignar ninguna asignatura.
- Varios profesores tienen ciertas restricciones horarias:
 - El Profesor 1 solo puede dar clase en las dos últimas horas de la mañana.
 - El Profesor 2 solo puede dar clase en las dos primeras horas de la mañana.
 - El Profesor 3 solo puede dar clase en la hora justo antes del recreo.
- Varias asignaturas también deben ser impartidas ciertos días concretos de la semana:
 - Asignatura 1: lunes o martes
 - Asignatura 3: miércoles o jueves
 - Asignatura 4: lunes o martes
 - Asignatura 5: jueves o viernes
 - Asignatura 6: miércoles o viernes
 - Asignatura 7: martes o viernes
 - Asignatura 8: miércoles
 - Asignatura 9: lunes

Responda a las siguientes preguntas de forma razonada: ¿Cuál es el número de soluciones válidas obtenidas? ¿Existen soluciones simétricas? Por soluciones simétricas se entienden aquellas que tienen valores distintos para las variables de la codificación CSP (por lo que MiniZinc las interpreta como soluciones diferentes), pero semánticamente representan la misma solución. En caso de que la codificación propuesta contenga soluciones 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.



3. **Problema lógico.** Cinco personas de cinco regiones diferentes viven en las primeras cinco casas contiguas de una calle. Practican cinco profesiones distintas, y cada uno tiene un animal y una bebida favoritos, todos ellos diferentes. Las casas están pintadas con diferentes colores. Además sabemos lo siguiente:

- a. El vasco vive en la casa roja.
- b. El catalán tiene un perro.
- c. El gallego es un pintor.
- d. El navarro bebe té.
- e. El andaluz vive en la primera casa de la izquierda.
- f. El de la casa verde bebe café.
- g. La casa verde está al lado de la blanca y a su derecha.
- h. El escultor cría caracoles.
- i. El diplomático vive en la casa amarilla.
- j. En la casa central se bebe leche.
- k. La casa del andaluz está al lado de la azul.
- l. El violinista bebe zumo.
- m. El zorro está en una casa al lado de la del médico.
- n. El caballo está en una casa al lado de la del diplomático.

Resolver el problema de forma que podamos responder: ¿dónde está la cebra y quién bebe agua?



4. **Problema de asignación de tareas.** En la tabla adjunta aparece la información necesaria para llevar a cabo la construcción de una casa. En la primera columna aparecen los identificadores de las tareas necesarias para construirla. En la segunda columna aparece la descripción de cada una de estas. En la tercera columna se muestra la duración (en días) de cada una de las tareas para cada uno de los trabajadores: por ejemplo, la duración de la tarea A “4,7,10” debe interpretarse como que esta tarea tardaría 4 días si la realizara el trabajador #1, 7 días si la realizara el trabajador #2, y 10 días si la realizara el trabajador #3. Y la cuarta columna muestra la relación de precedencia entre tareas: por ejemplo, si el contenido de la celda correspondiente a la tarea “F” es “C,D” se debe interpretar como que “las tareas C y D deben finalizarse antes que comience la tarea “F”. Cada tarea la realiza un único trabajador, pero los tres trabajadores pueden estar trabajando al mismo tiempo en tres tareas diferentes. Se pide encontrar una asignación de tiempos de inicio a estas tareas de forma que se pueda construir la casa en el menor tiempo posible.

Tarea	Descripción	Duración	Predecesoras
A	Levantar muros	4, 7, 10	Ninguna
B	Carpintería de tejado	3, 5, 7	A
C	Poner tejado	3, 1, 4	B
D	Instalación eléctrica	2, 5, 8	A
E	Pintado fachada	4, 2, 6	C,D
F	Ventanas	3, 2, 1	C,D
G	Jardín	1, 2, 3	C,D
H	Techado	1, 3, 5	A
I	Pintado interior	2, 3, 4	F,H

¿Cuál es la duración mínima de la construcción de la casa? Considere ahora que se dispone de un cuarto trabajador, que podrá ser asignado para apoyar a cualquier tarea/trabajador con duración mayor o igual a 3 días (es decir, si la duración de la tarea para el trabajador asignado es menos a 3 días, el cuarto trabajador de apoyo no podrá ser asignado a la misma), y consigue reducir la duración de esta tarea en dos días. Por ejemplo, si la tarea A es asignada al trabajador 1 con el trabajador de apoyo, la duración de esta tarea sería de 2 días. En ese caso, ¿qué tiempo tardarán, como mínimo, los cuatro trabajadores en finalizar la construcción? Entregue la solución MZN considerando al cuarto trabajador de apoyo. En la memoria debe responder a estas preguntas y entregar un diagrama de Gantt con la solución encontrada.



5. **Problema del coloreado de grafos.** El problema del coloreado de grafos consiste en asignar un color a cada nodo del grafo de forma que dos nodos adyacentes no tengan el mismo color, usando el mínimo número de colores. En este ejercicio se pide implementar una variante de este problema: se debe asignar un color a cada arista de forma que dos aristas que compartan algún nodo como extremo, deben tener colores diferentes.

Implemente una codificación en MiniZinc para resolver esta variante, y complete la información de la siguiente tabla, generando 3 grafos aleatorios para cada tamaño de grafo (con 3 semillas distintas) y reportando los resultados promedio de las 3 ejecuciones. Para la generación de grafos aleatorios, puede utilizar el siguiente código Python:

```
import numpy as np

# Función para generar un grafo aleatorio
# con N nodos y M aristas
def randomgraph(N, M, seed):

    print("int: NUM_NODOS =", N, ";")
    print("int: NUM_ARISTAS =", M, ";")
    print("array[1..NUM_ARISTAS,1..2] of int: aristas = [", end='')

    # Fijamos la semilla
    np.random.seed(seed)

    # Iteramos para cada arista
    for i in range(M):

        # Generamos dos nodos en [1,N]
        n1 = np.random.randint(0,N)+1
        n2 = np.random.randint(0,N)+1

        # Si son el mismo nodo, se re-genera
        while n1 == n2:
            n2 = np.random.randint(0,N)+1

        # Se imprime la arista
        print(n1, ",", n2, "|", end='')

    print("];")
    print("")
```

en el que N es el número de nodos y M es el número de aristas del grafo aleatorio generado. El tercer parámetro (seed) es la semilla aleatoria. Nótese que la generación de cada arista es independiente, por lo que pueden existir aristas repetidas (las cuales pueden – y deben— ser coloreadas con el mismo color).

Tamaño del grafo	Número de colores mínimo	Runtime (en segundos)
N=4, M=6		
N=6, M=15		
N=8, M=28		
N=10, M=45		
N=12, M=66		
N=14, M=91		

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.