

MA4702. Programación Lineal Mixta : Teoría y Laboratorio. 2018.

Profesor: José Soto



Tarea 1: Modelando variantes del sudoku.

Instrucciones.

- Usted debe entregar en ucursos antes de la hora de entrada al primer laboratorio los siguientes archivos con las instrucciones pedidas `sudoku-apellido.mod`, `sudoku-apellido.run`, `jigsaw-apellido.mod`, `jigsaw-apellido.run`, donde **apellido** es su apellido sin tilde y en minúsculas.
- Cada archivo **debe** comenzar con una línea de comentarios
`## Archivo (nombrearchivo) ##`
 donde **(nombrearchivo)** es el nombre del archivo incluyendo extensión.
- Realice esta primera tarea de manera **individual**. Bajo ningún motivo comparta o deje a disposición los archivos con sus compañeros. Se realizará un estudio de similitudes entre los archivos resultantes para detectar posibles plagios.
- **EXTRA: GUARDE, PERO NO ENTREGUE LOS ARCHIVOS GENERADOS POR SUS SCRIPTS.**
 Para efectos de evaluación, sus scripts deben funcionar abriendo `ampl` y ejecutando **include sudoku-apellido.run**, e **include jigsaw-apellido.run**.

1. SUDOKU

Un tablero M es una matriz en $[N]^{N \times N}$. Un conjunto de índices $C \subseteq [N] \times [N]$ es un **bloque** si todos los elementos $\{M_{ij} : ij \in C\}$ son diferentes. El tablero es un **sudoku** si $N = n^2$ es cuadrado perfecto y además, cada fila, cada columna y cada cuadrante son bloques. Aquí, un cuadrante es un elemento de $\{q^{ij} : i, j \in \{0, 1, \dots, n-1\}\}$ donde $q^{ij} = \{(n \cdot i + a, n \cdot j + b) : a, b \in [n]\}$. En otras palabras, los cuadrantes se obtienen al dividir el tablero de n^2 por n^2 en subtableros de n por n de manera natural.

En el **juego del sudoku** recibimos como entrada un sudoku incompleto, es decir un tablero Q que podría tener algunos casilleros vacíos, representados con un 0. Podemos escribir un PLE que complete Q a un sudoku M introduciendo variables binarias $\{x_{ijk} : i, j, k \in [N]\}$, tal que $x_{ijk} = 1$ si y solo si $M[i, j] = k$ y revisando factibilidad del conjunto:

$$\begin{aligned} \sum_{k \in [N]} x_{ijk} &= 1, & \forall (i, j) \in [N] \times [N]. \\ \sum_{(i, j) \in B} x_{ijk} &= 1, & \forall k \in [N], \forall B \text{ bloque.} \\ x_{ijk} &= 1, & \forall (i, j, k) \in [N]^3 : Q[i, j] = k. \\ x_{ijk} &\in \{0, 1\}, & \forall (i, j, k) \in [N]^3. \end{aligned}$$

Parte 1: Escriba el conjunto lineal entero anterior en un modelo en AMPL. Para esto complete el siguiente archivo y grábelo como `sudoku-apellido.mod`.

```
## Archivo (sudoku-apellido.mod) ##
param n;
param N:=n*n;
param Q {1..N, 1..N}; #datos preescritos
```

```
/* escribir variables y restricciones (escriba por separado las
condiciones para filas, columnas y cuadrantes)*/
```

Indicación: Recuerde que puede comprobar todas las declaraciones y conjuntos creados por `ampl` usando **show;** y las ecuaciones generadas usando **expand;**

Los archivos `sudoku-nn.txt` contienen sudokus incompletos de distintas dificultades donde el símbolo 0 es usado para representar un casillero vacío. Por ejemplo el archivo `sudoku-01.txt` luce como:

```
3
5 7 0 0 0 4 3 1 0
1 3 0 5 0 0 0 2 0
4 0 2 3 0 6 0 0 0
9 0 7 0 6 0 1 0 0
0 0 0 2 0 7 0 0 0
0 0 3 0 8 0 7 0 4
0 0 0 7 0 2 9 0 1
0 4 0 0 0 1 0 7 8
0 2 1 8 0 0 0 4 5
```

La primera línea del archivo tiene el valor n tal que $N = n^2$. Llamaremos a N el tamaño del sudoku. El sudoku anterior tiene tamaño 9.

Usted puede leer el archivo anterior y guardar sus datos en n y en Q declaradas antes, escribiendo **read n, {(i,j) in 1..N,1..N} Q[i,j] <sudoku-01.txt; close sudoku-01.txt;**

Al resolver la instancia anterior y reinterpretar los datos, encontrará que su solución puede ser escrita como:

```
salida [*,*]
: 1 2 3 4 5 6 7 8 9 :=
1 5 7 8 9 2 4 3 1 6
2 1 3 6 5 7 8 4 2 9
3 4 9 2 3 1 6 8 5 7
4 9 5 7 4 6 3 1 8 2
5 8 1 4 2 9 7 5 6 3
6 2 6 3 1 8 5 7 9 4
7 6 8 5 7 4 2 9 3 1
8 3 4 9 6 5 1 2 7 8
9 7 2 1 8 3 9 6 4 5
;
```

Parte 2: Escriba un script llamado `sudoku-apellido.run`, donde `apellido` es su apellido sin tilde y en minúsculas. Este script debe

1. Cargar el modelo `sudoku-apellido.mod`.
2. Ejecutar los comandos siguientes:

```
option presolve 0;
option solver cplex;
option display_lcol 0;
```

3. Leer y resolver los sudokus de cada archivo `sudoku-nn.txt` y entregar su solución como matriz en el archivo `sudoku-apellido-nn.txt`. En el caso que el sudoku no sea factible, el archivo debe contener una línea con la palabra `infactible`.
4. Escribir un archivo `sudoku-reporte.txt` como sigue:

```

SUDOKU.
Autor: (nombre-y-apellido-autor)

Instancia      - N - factible? - tiempo
sudoku-01.txt  - 9 - si         - 0.578125
sudoku-02.txt  - 4 - no         -
(...)

```

Indicaciones: En cada fila de su salida debe indicar el nombre del archivo, el tamaño del sudoku, si el sudoku tiene solución, y el tiempo de ejecución.

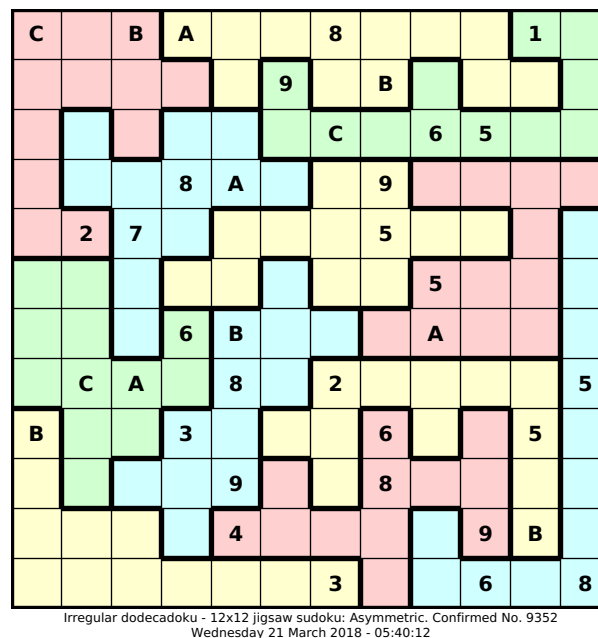
Recuerde que puede ejecutar su script usando el comando **include**, y que en cualquier momento puede eliminar los datos de un objeto **OBJ** cualquiera escribiendo **reset data OBJ**; (esto es útil para iterar y reescribir solo ciertos valores).

Para determinar si el solver entrega una solución factible, investigue la variable **solve_result_num**.

El último tiempo de ejecución se guarda en la variable **_total_solve_system_time**.

2. Jigsaw-Sudoku

Una variante interesante de sudoku es aquella donde los **cuadrantes** son reemplazados por otros **bloques**. Por ejemplo, en el siguiente puzzle (extraído de sudoku-puzzles-online.com) los bloques son (1) todas las filas, (2) todas las columnas y (3) todas las áreas contiguas del mismo color y separadas por líneas gruesas. Notamos en particular que N ya no necesita ser cuadrado perfecto. En la imagen los símbolos usados son 1, 2, 3, ..., 9, A, B, C.



The solution corresponding to this puzzle is available on your site.
sudoku-puzzles-online.com

Los archivos **jigsaw-nn.txt** contienen jigsaw-sudokus incompletos de distintas dificultades. Estos archivos consisten en una línea que indica el tamaño N del tablero. Luego una matriz con las entradas preescritas (usamos números consecutivos en vez del formato usado por la imagen anterior), donde el símbolo 0 es usado para representar un casillero vacío, una

línea en blanco, y finalmente un N -tablero que representa los N bloques que reemplazan a los cuadrantes. por ejemplo, el archivo `jigsaw-01.txt` que representa el puzzle anterior se ve como sigue:

```

12
12 0 11 10 0 0 8 0 0 0 1 0
0 0 0 0 0 9 0 11 0 0 0 0
0 0 0 0 0 0 12 0 6 5 0 0
0 0 0 8 10 0 0 9 0 0 0 0
0 2 7 0 0 0 0 5 0 0 0 0
0 0 0 0 0 0 0 0 5 0 0 0
0 0 0 6 11 0 0 0 10 0 0 0
0 12 10 0 8 0 2 0 0 0 0 5
11 0 0 3 0 0 0 6 0 0 5 0
0 0 0 0 9 0 0 8 0 0 0 0
0 0 0 0 4 0 0 0 0 9 11 0
0 0 0 0 0 0 3 0 0 6 0 8

1 1 1 2 2 2 2 2 2 2 3 3
1 1 1 1 2 3 2 2 3 2 2 3
1 4 1 4 4 3 3 3 3 3 3 3
1 4 4 4 4 4 5 5 6 6 6 6
1 1 4 4 5 5 5 5 5 5 6 7
8 8 4 5 5 9 5 5 6 6 6 7
8 8 4 8 9 9 9 6 6 6 6 7
8 8 8 8 9 9 10 10 10 10 10 7
11 8 8 9 9 10 10 12 10 12 10 7
11 8 9 9 9 12 10 12 12 12 10 7
11 11 11 9 12 12 12 12 7 12 10 7
11 11 11 11 11 11 11 12 7 7 7 7

```

Parte 3: Escriba un modelo en AMPL para representar el conjunto lineal entero y grábalo como `jigsaw-apellido.mod`, donde `apellido` es su apellido sin tilde y en minúsculas. Básele en la parte 1 de esta tarea. (no olvide que la primera línea de su archivo debe tener un comentario indicando el nombre del archivo)

Parte 4: Escriba un script llamado `jigsaw-apellido.run`, donde `apellido` es su apellido sin tilde y en minúsculas que realice el trabajo análogo al script de la parte 2 de esta tarea pero reemplazando en todas partes *sudoku* por *jigsaw*. En particular, su script debe generar archivos soluciones `jigsaw-apellido-nn.txt` y un reporte `jigsaw-reporte.txt`.

Nota: Las instancias factibles fueron extraídas de diversas fuentes. En particular, se usó sudoku-puzzles-online.com y www.websudoku.com.