

Choco Solver
Programmation par contrainte
Exercice M4 : Tomologie
Master Bases de données et intelligence
artificielle
2018-2019

Julien HALLE

November 4, 2018

Part I

Raisonnement

Résumé : Le raisonneur doit trouver une image en noir et blanc en ne connaissant uniquement que la somme des pixels noir des lignes, colonnes, diagonales montantes et descendantes.

Une image en noir et blanc est une matrice de pixel qui ne peut prendre que les valeurs de l'ensemble suivant : $\{0, 1\}$.

Les diagonales sont lues de gauche à droite.

Il existe une fonction permettant de déclarer des contraintes par des sommes : `model.sum(tableau,"=",somme)`

Il faudra un tableau contenant les valeurs que l'on veut additionner, ainsi que naviguer dans les valeurs en entrée (ligne, colonne, diag.montante, diag.descendante) pour connaître la valeur que doit faire cette somme.

4 grands axes à suivre pour trouver les tableaux à référencer en tant que contrainte :

1. Ligne : Une matrice est un tableau 1D qui contient un sous-tableau 1D par valeur : `tableau[0]` = première ligne
2. Colonne : Il faut récupérer les valeurs des colonnes dans un tableau 1D temporaire, pour cela on utilisera une fonction existante : `org.chocosolver.util.tools.ArrayUtils.getColumn(tableau,numero_colonne)`
3. Diag. montante : Il faut créer un tableau temporaire afin de déclarer la contrainte. A noter que chaque diagonale ne contient pas le même nombre de valeurs.
4. Diag. descendante : Idem que 3.

Part II

Images

La solution est exprimée sous forme de triplets donnant les coordonnées des points noir $x=y=1$

Figure 1 :

- ligne = (2,1,3)
- colonne = (3,1,2)
- diag.mont = (1,0,2,2,1)
- diag.desc = (1,1,2,1,1)

```
run:
** Choco 4.0.8 (2018-07) : Constraint Programming Solver, Copyright (c) 2010-2018
- Model[Tomologie n = 3] features:
    Variables : 13
    Constraints : 16
    Building time : 0,063s
    User-defined search strategy : no
    Complementary search strategy : no
- Complete search - 1 solution found.
    Model[Tomologie n = 3]
    Solutions: 1
    Building time : 0,089s
    Resolution time : 0,016s
    Nodes: 1 (62,0 n/s)
    Backtracks: 0
    Fails: 0
    Restarts: 0
Solution: 0=0=1, 1=0=1, 2=0=1, 0=1=0, 1=1=0, 2=1=1, 0=2=1, 1=2=0, 2=2=1,
  0|0|0|
  0|0|0|
  0|0|0|
BUILD SUCCESSFUL (total time: 0 seconds)
```

figure 1 : Affichage d'une image d'un coin

Figure 2 :

- ligne = (4,2,2,2)
- colonne = (3,2,3,2)
- diag.mont = (1,1,2,3,1,2,0)
- diag.desc = (1,2,2,2,2,0,1)

```
run:
** Choco 4.0.8 (2018-07) : Constraint Programming Solver, Copyright (c) 2010-2018
- Model[Tomologie n = 4] features:
    Variables : 21
    Constraints : 22
    Building time : 0,063s
    User-defined search strategy : no
    Complementary search strategy : no
- Complete search - 1 solution found.
    Model[Tomologie n = 4]
    Solutions: 1
    Building time : 0,088s
    Resolution time : 0,015s
    Nodes: 1 (64,5 n/s)
    Backtracks: 0
    Fails: 0
    Restarts: 0
Solution: 0=0=1, 1=0=1, 2=0=1, 3=0=0, 0=1=1, 1=1=0, 2=1=0, 3=1=1, 0=2=1, 1=2=1, 2=2=1, 3=
■|■|■|■|
■|■|■|■|
■|■|■|■|
■|■|■|■|
BUILD SUCCESSFUL (total time: 0 seconds)
```

figure 2 : Affichage d'une image représentant un R majuscule

Figure 3 :

- ligne = (1,1,6,3,3, 6,1,1)
- colonne = (2,2,2,6,4, 2,2,2)
- diag.mont = (0,0,1,2,1, 2,3,4,2,1, 4,2,0,0,0)
- diag.desc = (0,0,0,2,4, 1,2,4,3,2, 1,2,1,0,0)

[illegible]

figure 3 : Affichage d'une image représentant un personnage