Choco Solver
Programmation par contrainte
Exercice M6 : Nombre de Schur
Master Bases de données et intelligence artificielle
2018-2019

Julien HALLE

November 4, 2018

Part I
Raisonnement

Résumé :
Pour tout triplet (a, b, c) avec c = a + b et a, b, c appartenant à $\mathbb{N}^\star$ et un nombre de couleur défini n, il est possible de colorier chacune des variables a,b,c de tel façon que le triplet ne soit jamais monochrome jusqu'à une certaine valeur de c. On appelle ce nombre le nombre de Schur. On le note : S(n) = max(c).

1 Représentation

Le nombre de Schur est la valeur maximale de c pour laquelle on peut colorier de facon non monochrome un triplet.
On connait ses bornes minimale et maximale :

- $\min(S(n)) = \frac{(3^n-1)}{2}$
- $\max(S(n)) = (3 \times n!) - 1$

On représente les nombres à colorier et leurs couleurs associées par un tableau où l'index représente le nombre et la valeur représente la couleur.
Cependant :

- L'index 0 n'est jamais utilisé (pour éviter de devoir décaler tous les résultats de 1).
- Les index utilisables sont compris entre min(S(n)) et max(S(n)).
- La valeur 0 indique les variables non utilisées lors de la résolution.
- Les valeurs sont comprises entre 1 et n.

Les triplets sont représentés dans une classe Triplet avec 3 variables a, b et c.

2 Contraintes

On génère tous les triplets possibles pour les valeurs allant de min(S(n)) à max(S(n)). Il faut éviter de générer les triplets mirroir qui génèreraient des doublons dans les contraintes.
Exemple : $(1, 2, 3)$ et $(2, 1, 3)$

La contrainte principal étant pour un triplet généré (a, b, c) :
Si a=b alors colors[a] != colors[c] avec colors le tableau contenant les couleurs associés à un nombre.

Le raisonnement est le même que si l'on essayait de résoudre de façon manuel le problème. Une fois les triplets de la forme $(a, a, 2a)$ résolu, il est plus aisé de continuer à attribuer les couleurs.

On forcera le chiffre 1 à avoir la couleur 1, pour éviter la génération de noeuds supplémentaires.
Exemple : Pour S(3) en fixant 1 = couleur(1) on trouve 70 noeuds et sans on trouve 249 noeuds, soit 179 noeuds évités.

3 Problèmes du modèle

Pour que le solveur trouve une solution, il faut un domaine borné, or cela créé des contraintes inutiles pour certaines valeurs :
Exemple : Pour trouver S(3) avec min(S(3))=13 et max(S(3))=17, si on cherche à évaluer tous les nombres strictement inférieurs à 14 alors les triplets $(1, 14, 15)$, $(2, 13, 15)$ jusqu'à $(7, 8, 15)$ génèrent des contraintes inutiles (idem pour les triplets de 16 jusqu'à 17).

Il faut préciser au solver de pas prendre en considération ces contraintes (indiquée par la couleur 0).

Part II
Images

Figure 1 : On fixe n = 1

```
run:
** Choco 4.0.8 (2018-07) : Constraint Programming Solver, Copyright (c) 2010-2018
- Model[Borne inférieur de Schur S(1){1;2}] features:
        Variables : 26
        Constraints : 19
        Building time : 0,083s
        User-defined search strategy : yes
        Complementary search strategy : no
- Complete search - 1 solution found.
        Model[Borne inférieur de Schur S(1){1;2}]
        Solutions: 1
        MAXIMIZE borne = 1,
        Building time : 0,083s
        Resolution time : 0,045s
        Nodes: 2 (44,5 n/s)
        Backtracks: 3
        Fails: 1
        Restarts: 0
Solution: borne=1, c=0=0, c=1=1, c=2=0, REIF_1=0, not(REIF_1)=1, REIF_2=1, not(REI
Non_utilisé:   0   2
Couleur_n°1:   1

Temps écoulé : hour:00 - min:00 - sec:00 - millis:0262
BUILD SUCCESSFUL (total time: 0 seconds)
```

figure 1 : Calcul pour S(1)

Figure 2 : On fixe n = 2

```
run:
** Choco 4.0.8 (2018-07) : Constraint Programming Solver, Copyright (c) 2010-2018
- Model[Borne inférieur de Schur S(2){4;5}] features:
        Variables : 92
        Constraints : 64
        Building time : 0,097s
        User-defined search strategy : yes
        Complementary search strategy : no
- Complete search - 1 solution found.
        Model[Borne inférieur de Schur S(2){4;5}]
        Solutions: 1
        MAXIMIZE borne = 4,
        Building time : 0,097s
        Resolution time : 0,047s
        Nodes: 2 (42,7 n/s)
        Backtracks: 3
        Fails: 1
        Restarts: 0
Solution: borne=4, used=2, c=0=0, c=1=1, c=2=2, c=3=2, c=4=1, c=5=0, REIF_1=1, not
Non_utilisé:    0   5
Couleur_n°1:    1   4
Couleur_n°2:    2   3

Temps écoulé : hour:00 - min:00 - sec:00 - millis:0263
BUILD SUCCESSFUL (total time: 0 seconds)
```

figure 2 : Calcul pour S(2)

Figure 3 : On fixe n = 3

```
run:
** Choco 4.0.8 (2018-07) : Constraint Programming Solver, Copyright (c) 2010-2018
- Model[Borne inférieur de Schur S(3){13;17}] features:
        Variables : 770
        Constraints : 523
        Building time : 0,212s
        User-defined search strategy : yes
        Complementary search strategy : no
- Complete search - 1 solution found.
        Model[Borne inférieur de Schur S(3){13;17}]
        Solutions: 1
        MAXIMIZE borne = 13,
        Building time : 0,212s
        Resolution time : 0,150s
        Nodes: 70 (467,8 n/s)
        Backtracks: 139
        Fails: 69
        Restarts: 0
Solution: borne=13, used=3, c=0=0, c=1=1, c=2=2, c=3=2, c=4=1, c=5=3, c=6=3, c=7=2,
Non_utilisé:   0  14  15  16  17
Couleur_n°1:   1   4  10  13
Couleur_n°2:   2   3   7  11  12
Couleur_n°3:   5   6   8   9

Temps écoulé : hour:00 - min:00 - sec:00 - millis:0424
BUILD SUCCESSFUL (total time: 0 seconds)
```

figure 3 : Calcul de S(3)

Figure 4 : On fixe n = 4

```
run:
** Choco 4.0.8 (2018-07) : Constraint Programming Solver, Copyright (c) 2010-2018
- Model[Borne inférieur de Schur S(4){40;71}] features:
        Variables : 11840
        Constraints : 7933
        Building time : 44,222s
        User-defined search strategy : yes
        Complementary search strategy : no
- Complete search - 3 solution(s) found.
        Model[Borne inférieur de Schur S(4){40;71}]
        Solutions: 3
        MAXIMIZE borne = 44,
        Building time : 44,222s
        Resolution time : 44,005s
        Nodes: 13 621 (309,5 n/s)
        Backtracks: 27 237
        Fails: 13 616
        Restarts: 0
Solution: borne=44, used=4, c=0=0, c=1=1, c=2=2, c=3=2, c=4=1, c=5=3, c=6=3, c=7=3, c=8=3, c=9=1, c=10=2, c=11=2, c=12=1, c=13
Non_utilisé:   0  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71
Couleur_n°1:   1   4   9  12  19  26  33  36  44
Couleur_n°2:   2   3  10  11  16  29  30  34  35  42  43
Couleur_n°3:   5   6   7   8  17  18  27  28  37  38  39  40  41
Couleur_n°4:  13  14  15  20  21  22  23  24  25  31  32

Temps écoulé : hour:00 - min:00 - sec:44 - millis:0577
BUILD SUCCESSFUL (total time: 44 seconds)
```

figure 4 : Calcul de S(4)