

Réunion du 14 juin

Tests solveur n-reines

- Nombre de reines testées :
 - 10
 - 50
 - 100
 - 110
 - 300
 - 674
 - 677

Listes me permettant de nommer les variables

```
#ces deux listes me permettent de nommer les variables  
alpha = "abcdefghijklmnopqrstuvwxyz"  
Alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

Ma nouvelle manière de nommer les variables

```
tab = []
for k in range(num_reines):
    #une manière de nommer les variables permettant en théorie de mettre le nombre de variables qu'on veut
    name = ""
    n = k//26
    tab2 = [n]
    while(n>25):
        n = n//26
        tab2.append(n)
    name = name + alpha[tab2[len(tab2)-1]]
    for l in range(len(tab2)-1):
        name = name + alpha[tab2[len(tab2)-1-l]//(26*(len(tab2)-1-l))]
    name = Alpha[k%26] + name
    tab.append(model.new_int_var(1, num_reines, name))
```

Test 10 reines

```
Duree de la recherche d'une solution = 0.0377114460006851 s
-----
| | | X | | | | | | | |
-----
| | | | | | | X | | |
-----
| | | | | X | | | | |
-----
| | X | | | | | | | |
-----
| | | | | | | | | X |
-----
| | | | X | | | | | |
-----
| X | | | | | | | | |
-----
| | | X | | | | | | |
-----
| | | | | | | X | | |
-----
| | | | | X | | | | |
-----
```

Nouvelle manière d'afficher solution de n-reines

```
if status == cp_model.OPTIMAL or status == cp_model.FEASIBLE: #s'il existe une solution
    print(affichage_echiquier())
else: #sinon
    print("Pas de solution.")

print("Duree de la recherche d'une solution = ",(fin-debut),"s") #affichage de la durée de recherche d'une solution
```

Nouvelle manière d'afficher solution de n-reines

```
def affichage_echiquier(): #affichage des coordonnées des reines
    ch = "Affichage des coordonnees (x,y) des reines telles que 1 <= x,y <= " + str(num_reines) + " :\n"
    for i in range(num_reines):
        ch += "Reine n°" + str(i+1) + " : (" + str(i+1) + "," + str(solver.value(tab[i])) + ")\n"
    return ch
```

Fin du test de 300 reines

```
Reine n°275 : (275,227)
Reine n°276 : (276,225)
Reine n°277 : (277,233)
Reine n°278 : (278,231)
Reine n°279 : (279,239)
Reine n°280 : (280,237)
Reine n°281 : (281,245)
Reine n°282 : (282,243)
Reine n°283 : (283,251)
Reine n°284 : (284,249)
Reine n°285 : (285,257)
Reine n°286 : (286,255)
Reine n°287 : (287,263)
Reine n°288 : (288,261)
Reine n°289 : (289,269)
Reine n°290 : (290,267)
Reine n°291 : (291,275)
Reine n°292 : (292,273)
Reine n°293 : (293,281)
Reine n°294 : (294,279)
Reine n°295 : (295,287)
Reine n°296 : (296,285)
Reine n°297 : (297,293)
Reine n°298 : (298,291)
Reine n°299 : (299,299)
Reine n°300 : (300,297)
```

```
Duree de la recherche d'une solution = 795.6583561090001 s
```


Plantage dès 674 reines

```
aclotild@aclotild-OptiPlex-7040:~/Documents/Codes$ python3 solveur_nreines2.py  
Nombre de reines : 674  
  
Processus arrêté
```

```
aclotild@aclotild-OptiPlex-7040:~/Documents/Codes$ python3 solveur_nreines2.py  
Nombre de reines : 677  
  
Processus arrêté
```

Hypothèses possibles

- Limite de variables / contraintes d'OR-Tools atteinte
- Limite de variables / contraintes de ce solveur atteinte
- Problème de l'ordinateur de l'IRIT

Questions

- Est-ce que la limite d'OR-Tools a été atteinte ?
- Est-ce que d'autres solveurs pourraient traiter 674 variables ?
- Est-ce que l'ordinateur de l'IRIT a un problème ?

Coloration de graphe

- Fichiers testés du dossier GraphColoring :
 - Dans le sous-dossier « GraphColoring-m1-mono » :
 - GraphColoring-1-fullins-3.xml (30 sommets)
 - GraphColoring-1-fullins-4.xml (93 sommets)
 - GraphColoring-zeroin-i-3.xml (206 sommets)

Example : GraphColoring-1-fullins-3.xml

```
1 <instance format="XCSP3" type="COP">  
2   <variables>  
3     <array id="x" size="[30]"> 0..29 </array>  
4   </variables>  
5   <constraints>
```

Example : GraphColoring-1-fullins-3.xml

```
6      <group>
7          <intension> ne(%0,%1) </intension>
8          <args> x[0] x[1] </args>
9          <args> x[0] x[3] </args>
10         <args> x[0] x[10] </args>
11         <args> x[0] x[12] </args>
12         <args> x[1] x[2] </args>
13         <args> x[1] x[9] </args>
14         <args> x[1] x[11] </args>
15         <args> x[2] x[5] </args>
16         <args> x[2] x[7] </args>
17         <args> x[2] x[10] </args>
18         <args> x[2] x[14] </args>
19         <args> x[2] x[16] </args>
20         <args> x[3] x[4] </args>
```

Example : GraphColoring-1-fullins-3.xml

```
<objectives>  
  <minimize type="maximum"> x[] </minimize>  
</objectives>  
</instance>
```

Exemple : GraphColoring-1-fullins-3.xml

```
arcs = [[0 for j in range(s)] for i in range(s)] #liste de nombres binaires representants les arcs
```

```
if ((len(debut)>0)&(len(fin)>0)): #si les sommets de debut et de fin sont effectivement releves
    D = int(debut) #somet de debut
    F = int(fin) #somet de fin
    arcs[D][F]=1 #formation de l'arc DF
    arcs[F][D]=1 #formation de l'arc FD (note : DF = FD)
```


Exemple : GraphColoring-1-fullins-3.xml

```
tab.append(model.new_int_var(min, max, name))

#initialisation des contraintes
for i in range(s):
    for j in range(s):
        if ((arcs[i][j]==1)&(i!=j)): #si 2 sommets différents sont reliés
            model.add(tab[i]!=tab[j]) #leur couleur doit être différente
```

Exemple : GraphColoring-1-fullins-3.xml

Duree de la recherche d'une solution = 0.026088685000104306 s

Z = 2

A = 3

B = 4

C = 1

D = 0

E = 2

F = 15

G = 0

H = 8

I = 2

J = 1

K = 1

L = 1

M = 3

N = 1

O = 1

P = 3

Q = 1

R = 22

S = 0

T = 0

U = 0

V = 0

W = 0

X = 0

Y = 0

Za = 0

Aa = 1

Ba = 0

Ca = 16

Remarque sur GraphColoring-m1-fixed

```
110      <extension>
111        <list> %0 </list>
112        <supports> 0..4 </supports>
113      </extension>
114      <args> x[2] </args>
115      <args> x[4] </args>
116      <args> x[5] </args>
117      <args> x[11] </args>
118      <args> x[12] </args>
119      <args> x[13] </args>
120      <args> x[16] </args>
121      <args> x[21] </args>
122      <args> x[22] </args>
123      <args> x[23] </args>
124    </group>
125    <instantiation>
126      <list> x[0..1] x[3] x[6..10] x[14..15] x[17..20] x[24] </list>
127      <values> 0 1 3 4 1 2 0 1 3 2 0 1 4 4 1 </values>
```

Pour GraphColoring-qwhdec-o5-h10-1.xml (24 sommets)