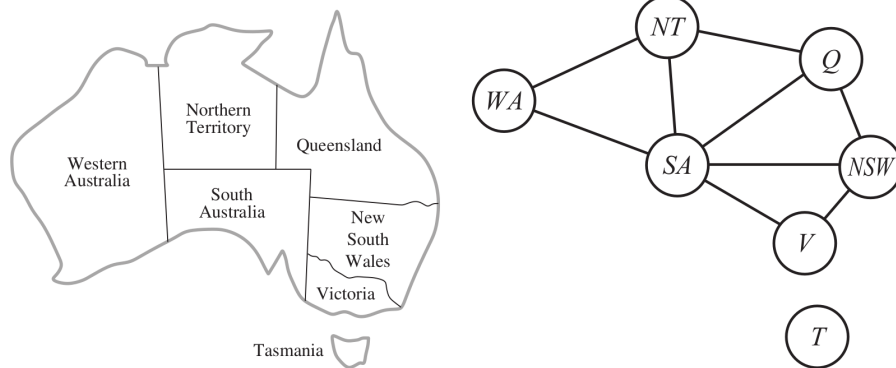


Introduction à l'IA - Programmation par contraintes

Christophe Rodrigues

1 Coloration



Le but est de colorier la carte de l'Australie sans que deux états contigus n'utilisent la même couleur. On suppose que seulement 3 couleurs différentes sont disponibles. Le graphe représente le problème où chaque état est un noeud. Deux noeuds sont connectés s'ils sont contigus.

Question 1 : Définir les variables nécessaires.

Question 2 : Définir les contraintes sur les variables afin de réussir à colorier au mieux la carte.

2 Message crypté

Vous devez trouver des valeurs entières associables à chacune des lettres de l'équation suivante en respectant toutes les contraintes:

$$\begin{array}{rcccc} & S & E & N & D \\ + & M & O & R & E \\ \hline M & O & N & E & Y \end{array}$$

Par exemple si $E=1$ et $D=2$, alors $Y=3$ (car $E+D=Y$)

Question 1 : Définir les variables nécessaires.

Question 2 : Définir les contraintes sur les variables.

3 Le problème des 8 reines(*bis*)

Reprendre le problème des 8 reines vu au Tp 1 et le résoudre à l'aide de la programmation par contrainte.

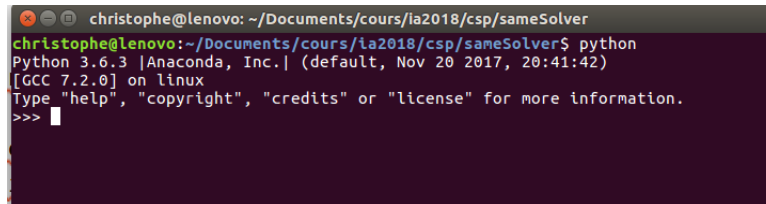
Question 1 : Définir les variables nécessaires.

Question 2 : Définir les contraintes sur les variables.

4 Utilisation d'un solveur

Afin de tester nos problèmes à l'aide d'un solveur, nous allons utiliser la librairie OR-tools de google. Cette librairie est multi-plateforme(Windows/Linux/Mac) mais aussi multi-langage (Java/C++/C#/Python). Mais pour ceux qui ne sont pas familiers avec Python c'est l'occasion d'essayer.

Si vous désirez faire le Tp en Python vous devez l'installer avant la librairie OR-tools sur votre machine. Python est disponible à l'adresse suivante (bien télécharger la version 64 bits): <https://www.python.org/downloads/>
Sous windows, souvent plusieurs versions de Python cohabitent. Afin de vérifier si votre système appelle bien la dernière version de Python, lancer-le en ligne de commande (taper *cmd* dans la barre de lancement). La commande *python* affichera la version de celui-ci :



```
christophe@lenovo: ~/Documents/cours/ia2018/csp/sameSolver
christophe@lenovo:~/Documents/cours/ia2018/csp/sameSolver$ python
Python 3.6.3 [Anaconda, Inc.] (default, Nov 20 2017, 20:41:42)
[GCC 7.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Pour sortir taper : *exit()* . Si la version affichée n'est pas la dernière installée, il faut modifier votre variable d'environnement PATH afin de lui indiquer le chemin de votre dernière version de Python.

Afin d'installer le solveur de la librairie OR-tools vous devez suivre la procédure décrite à la page suivante : <https://developers.google.com/optimization/install/> et suivre les instructions.

Vous pouvez également installer un environnement de développement comme PyCharm : <https://www.jetbrains.com/pycharm/>
Créer un nouveau projet, ensuite faire File → Setting → Project Interpreter → "+" on se retrouve dans l'ajout de package et choisir ORTOOLS.

5 Sudoku

Traduire le problème du Sudoku en terme de variables et de contraintes. Le coder et le tester. Dans un premier temps il sera préférable d'avoir une taille de

Sudoku plus petite (4 carrés de 9 cases au lieu de 9 carrés de 9 cases classiques). De plus vous commencerez à l'aide d'une grille initialisée à la main.

Dans un deuxième temps vous devrez créer automatiquement la grille du Sudoku en respectant les contraintes du jeu. Pour un Sudoku classique de 9 carrées de 9 cases chacun, on considère un jeu comme:

- très difficile si seulement 17 cases sont données
- difficile si seulement 26 cases sont données
- moyen si 33 cases sont données
- facile si 40 cases sont données
- débutant si 50 cases sont données

L'affichage se fera sous la forme d'une matrice en mode texte. Ce travail peut être réalisé en binôme et sera à remettre sur le dépôt brightspace dans les 2 semaines.