

# Modélisation en PPC

## Projet

Master Informatique 1st year  
Université Nice-Sophia Antipolis

## 1 Création de Sudoku

Le but de ce projet est d'être capable d'engendrer des problèmes Sudoku automatiquement et de leur associer une difficulté.

Le problème Sudoku est bien connu :

<https://fr.wikipedia.org/wiki/Sudoku>

Nous considérerons ici la formulation classique du sudoku. Il s'agit de remplir un carré comportant 9x9 cellules avec des chiffres de 1 à 9 tel que les chiffres d'une même ligne sont tous différents, les chiffres d'une même colonne sont tous différents et les chiffres des sous carrés 3x3 sont tous différents.

Pour se familiariser avec le problème le mieux est d'écrire un modèle en Choco qui remplit une grille vide. Ensuite, on adaptera ce premier modèle afin qu'il puisse tenir compte de cellules déjà définies.

### 1.1 Génération de Sudoku

Le problème qu'on essaie de résoudre correspond à celui que l'on trouve dans les journaux ou les livres spécialisés : on veut définir des grilles de sudoku qui comportent soit des cases vides, soit des cases affectées avec une unique valeur et surtout qui sont bien formées. Autrement dit ils n'ont qu'une et une seule solution.

Pour obtenir ce résultat, on peut procéder comme suit :

- On part d'une grille vide et on calcule une solution
- Une fois qu'on a cette solution on remplace petit à petit des affectations de la solution par des cases vides (on dé-affecte les variables) et pour chaque remplacement on calcule s'il existe une UNIQUE solution. Si c'est le cas on continue sinon on essaie à la place de la cellule considérée une autre cellule. On s'arrête quand on n'arrive plus à remplacer des affectations par des cellules vides sans que cela conduise à deux solutions.

Cet algorithme est donné à titre indicatif et vous pouvez en proposer un autre.

Vous devriez être capables d'engendrer des Sudoku ayant assez peu de cellules affectées.

### 1.2 Niveau de difficulté

Une fois qu'on a créé un Sudoku encore faut-il déterminer sa difficulté

C'est là que la programmation par contraintes entre en jeu.

Plus un sudoku nécessite de règles de déduction complexes et plus il est considéré comme difficile. Les problèmes les plus difficiles sont ceux qui nécessitent de faire au moins un backtrack.

Pour déterminer la difficulté d'un Sudoku vous devez donc essayer de le résoudre en utilisant des modèles impliquant des contraintes associées à des règles de déduction de plus en plus complexes. Par exemple, remplacer la contrainte alldiff par des contraintes binaires de différence est un modèle beaucoup moins puissant que le modèle qui utilise des contraintes alldiff associées à un algorithme de filtrage effectuant l'arc consistence.

Vous devez proposer plusieurs modèles, chacun associé avec un niveau de difficulté, pour résoudre le Sudoku. Donc plus le modèle sera "simple" et plus le niveau du Sudoku sera faible s'il est résolu avec ce modèle sans faire de backtrack.

## 2 Rendu du projet

Le projet doit être rendu avant le 30 Mai 2020.

Un dossier zip à votre nom doit être envoyé par email à [jcregin@gmail.com](mailto:jcregin@gmail.com). En cas de problème un message associé à un dépôt oneDrive de l'Université est accepté. Le dossier zip doit contenir :

1. Le code source de votre projet
2. Une explication sur ce qui est nécessaire pour le compiler et l'exécuter. Eventuellement un fichier pom.xml
3. Un rapport expliquant votre projet et détaillant l'algorithme de génération que vous avez implémenté et les différents modèles que vous avez utilisés