07/01/2018

Adrien Bonfils ; Clément Braillon

M2 iia / M2 BDIA

OIA - PPC

Queens Completion

Table des matières

[Introduction 3](#_Toc503100891)

[Rappel du problème 3](#_Toc503100892)

[Présentation de la solution 3](#_Toc503100893)

[Solution 3](#_Toc503100894)

# Introduction

## Rappel du problème

Une instance du problème est un échiquier de taille N \* N, sur lequel doivent être placées K reines. Ces K reines doivent être placées de façon à ne pas être en conflit avec les autres (au plus une sur chaque ligne, colonne et diagonale). On appellera résolution d’une instance, le fait de déterminer s’il existe un moyen de placer les N-K reines restantes sans créer de conflits (voir problème N-Queens).

Le but de ce projet est d’analyser la difficulté, en fonction de N et K, de la résolution d’instances produites aléatoirement.

## Présentation de la solution

Pour ce projet, nous avons découper le travail en deux tâches distinctes. Ces deux tâches n’ont pas de grosses dépendances entre elles, ce qui a permis de les réaliser plus ou moins en même temps. Une des deux tâches consistait à créer la génération aléatoire d’une « pré-solution », c’est-à-dire de placer k reines aléatoirement sur un échiquier de taille n\*n (en prenant ces attributs en paramètres). La deuxième tâche consistait à faire marcher le solveur Choco (application des contraintes), traitement de la sortie (prettyOut) ainsi que gérer les multiplications d’instances d’échiquier pour chaque essais (30 instances par test par exemple).

# Solution

### Structure mise en place :

Pour générer nos instances nous avons utilisés un tableau à deux dimensions qui représente l’échiquier sur lesquelles sont placées les reines. Ainsi que deux tableaux à une dimension qui représente les lignes et les colonnes encore utilisable de l’échiquier.

### Placement des k reines :

Pour placer une reine on sélectionne de façon aléatoire une ligne et une colonne parmi nos tableaux de lignes et colonnes. Une fois la ligne et colonne sélectionné on teste les diagonales pour s’assurer que l’on peut placer la reine. Si on peut placer la reine on l’ajoute dans le grille et on supprime la ligne et la colonne correspondante dans nos tableaux de ligne et colonne restantes. Si on ne peut pas placer la reine, on essaye de décaler la reine le long de ligne en faisant attention a ce que la nouvelle colonne soit dans notre tableau de colonnes restantes et on reteste les diagonales. Si on ne peut toujours pas placer la reine on suppose que ce n’est pas possible et on relance une génération entière.

### Echec de génération :

Lors de la réalisation de notre échantillon de test nous avons remarqué de pour des valeurs de n et k supérieur à 20 nous pouvions avoir un temps de génération pour une instance très longue c’est pour cela que nous avons décidé de mettre en place une limite de temps pour la génération des instances. Dans nos test la limite de temps est de 1 seconde et une fois ce temps dépasser la génération de l’instance est marquer comme un échec de génération, cependant cela ne veut pas dire que notre solution ne permet pas de générée des instances pour ces valeurs de n et k mais que cela prendras probablement beaucoup de temps car notre solution n’est pas optimale.

Exemple pour n=20 et k = 18-19 avec temps de 1 seconde et avec un temps de 10 secondes :

|  |  |  |
| --- | --- | --- |
| N\K | 18 | 19 |
| 20(1s) | 1 | 18 |
| 20(10s) | 0 | 7 |

Résultat :

Si l’on prend les valeurs de n pour lesquels le nombre d’échec de génération est nulles, on remarque que les nombres de solutions trouvées chute quand k est supérieur ou égale à la moitié de n sauf pour k égale à n-1 ou le nombre de solutions trouvées remontent légèrement probablement car les chances de placer une reine qui empêcherais de résoudre l’instance sont plus faible qu’avec 2 reine restantes.