

Projet programmation par contrainte : Elisabeth

T.Béziers La fosse, D.Bordet, J. Clayton, A. Giraudet, B. Moreau

29 février 2016



Sommaire

1	Introduction	3
1.1	Problématique et but du projet	3
1.2	le problème des n reines et les contraintes	3
1.3	Composition de notre groupe	4
2	Partie 1 : Complete Search	4
2.1	Définition	4
2.2	4
3	Partie 2 : Local Search	4
3.1	Définition	4
3.2	5
4	Conclusion	5

1 Introduction

1.1 Problématique et but du projet

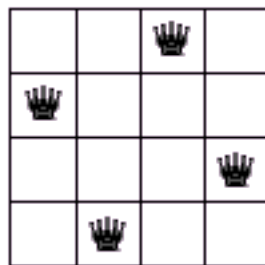
Le but de ce projet en programmation par contrainte est d'implémenter un mini solveur pour des domaines CPSs finis. La conception de ce projet s'étend sur trois séances de TP et doit nous permettre de proposer un solveur pour la recherche complète et la recherche local et comparer les solutions pour chaque méthode du problème dit "des n reines".

Nous avons décidé d'utiliser python 3.5.1 comme support de programmation et nous traiterons dans une première partie du **Complete Search** et dans la seconde du **Local Search**. Enfin, nous discuterons des résultats et de la comparaison de ceux ci dans la conclusion.

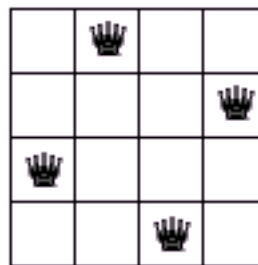
1.2 le problème des n reines et les contraintes

Le problème des n reines correspond à la mise en place de n reines sur un échiquier. Chaque reine ne doit pas être en position d'en attaquer une autre. Il faut ainsi donc placer les reines d'une façon définie et respectant les contraintes suivante :

- Il doit y avoir qu'une reine par ligne.
- Il doit y avoir qu'une reine par colonne.
- Il ne doit pas y avoir plus d'une reine par diagonal ;



(2413)



(3142)

1.3 Composition de notre groupe

Thibault	Béziers La fosse	M1ALMA
Dennis	Bordet	M1ALMA
Joachim	Clayton	M1ALMA
Alexis	Giraudet	M1ALMA
Benjamin	Moreau	M1ALMA

2 Partie 1 : Complete Search

2.1 Définition

Le Complete Search est une méthode qui vise à rechercher une solution au problème en parcourant un arbre de possibilité progressivement remplie. Celui-ci utilise un algorithme de backtracking.

2.2

3 Partie 2 : Local Search

3.1 Définition

Le Local Search elle est une méthode qui

3.2

4 Conclusion