



# Constraint (AI Space) : un outil de résolution de CSPs

Julien Alexandre dit Sandretto

Department U2IS  
ENSTA Paris  
IA302-2020-2021



Introduction

L'outil

Le problème

La modélisation sous forme de CSP

Création et résolution du CSP dans l'outil

## Vu précédemment



Comment résoudre un CSP, comment écrire un algorithme simple en utilisant le simple retour arrière et la consistance pour résoudre le problème du Sudoku (certains auront peut être implémenté un début de solveur).

Dans ce cours, nous allons étudier plus précisément comment se déroule la résolution d'un CSP au moyen d'un solveur graphique.

# L'outil

Télécharger Consistency based CSP solver :

<http://aispace.org/constraint/version4.6.1signed/constraint.jar>

Approche graphique pour la résolution de CSPs en utilisant la consistance.

Il se lance depuis un terminal avec la commande “java -jar constraint.jar”.

# Auto-apprentissage



Des tutoriaux ainsi qu'une aide sont disponibles :  
<http://www.aisspace.org/constraint/help/general.shtml>

# Le problème



Nous allons résoudre le problème très connu des reines (ici 5).

Placer 5 reines sur un échiquier  $5 \times 5$  de telle sorte qu'aucune d'entre elles ne soit en prise. Prises deux à deux, elles ne peuvent donc pas s'attaquer. Ceci implique qu'elles se trouvent sur des lignes, colonnes et diagonales différentes.

# La modélisation sous forme de CSP



Ce problème a été beaucoup étudié et plusieurs modélisations ont déjà été proposées.

La plus “évidente”, celle qui vient en premier à l’esprit est de considérer comme variables les positions données par les couples (ligne,colonne) pour chaque reine. Une deuxième un peu plus avancée consiste à se dire que comme elles ne peuvent être sur une même ligne, les seules vraies variables sont les colonnes.

## solution

Voici comment on peut modéliser le problème généralisé :

- ▶ Variables :  $\mathcal{X} = \{x_i; i \text{ est un entier compris entre } 1 \text{ et } n\}$
- ▶ Domaines : quelque soit  $x_i$  élément de  $\mathcal{X}$ ,  
 $D(x_i) = \{j; j \text{ est un entier compris entre } 1 \text{ et } n\}$
- ▶ Contraintes :
  - ▶ les reines doivent être sur des lignes différentes  
 $C_{lig} = \{x_i \neq x_j; i, j \text{ sont } 2 \text{ entiers différents compris entre } 1 \text{ et } n\}$
  - ▶ les reines doivent être sur des diagonales montantes différentes  
 $C_{dm} = \{x_i + i \neq x_j + j; i, j \text{ sont } 2 \text{ entiers différents compris entre } 1 \text{ et } n\}$
  - ▶ les reines doivent être sur des diagonales descendantes différentes  
 $C_{dd} = \{x_i - i \neq x_j - j; i, j \text{ sont } 2 \text{ entiers différents compris entre } 1 \text{ et } n\}$



## solution



Une autre modélisation qui peut être utilisée consiste à prendre comme variable non plus la position des reines, mais l'occupation des cases. En effet, toutes les reines sont identiques, et peu importe que ce soit la reine A ou la reine B qui occupe une case.

# Création et résolution du CSP dans l'outil



Créez le CSP dans l'outil graphique et résolvez le en comprenant bien le déroulement.

Pour cela, l'outil propose des boutons permettant de résoudre un CSP étapes par étapes, choix de l'arc, vérifier la consistance, revenir en arrière, sélectionner les valeurs à tester, etc.

Remarque :

Le problème des 5 reines est déjà dans les exemples de l'outil, cependant il ne vous servira pas à comprendre comment on construit un CSP (il utilise des contraintes particulières), donc jouez le jeu. Par contre, si vous n'arrivez pas à le créer, alors utiliser l'exemple pour comprendre les étapes de la résolution.