Bases de l'Intelligence Artificielle

PROLOG & IA

Marie Lefevre

2023-2024

Université Claude Bernard Lyon 1

De quoi va-t-on parler ?

- Les débuts
  - Avec Eliza
- Résolution par satisfaction de contraintes (CSP)
  - Avec les N-Reines
- Recherche dans un graphe d’états
  - Avec le labyrinthe
- Décomposition de problèmes
  - Avec les Tours de Hanoi

Les débuts avec Eliza

- Classique des débuts de l’IA
- N’utilise absolument pas d’IA

- Est censé simuler une conversation avec un psychologue

Dialogue avec Eliza

?- eliza.
Bonjour. Qu est ce qui vous amene ?
|: "je suis triste".
depuis combien de temps etes vous triste ?
|: "depuis le depart de mon frere".
pouvez vous me parler de votre frere ?
|: "j aime discuter avec lui".
quelqu un d autre dans votre famille aime t il cela ?
|: "non".
continuez
|: "je me sens seule maintenant".
ressentez vous souvent cela ?
|: "oui, depuis que mon frere est parti !".
pouvez vous me parler de votre frere ?
|: "vous vous moquez de moi ?".
continuez
|: "bye".

Eliza : principe du programme

- Principe du « Stimulus – Réaction »
  - « bonjour / bienvenu »
  - « je suis X / depuis combien de temps êtes vous X ? »

- Lire l'entrée de l'utilisateur
- Tant que l'entrée n'est pas « bye »
  - Choisir une paire « stimulus-réaction »
  - Apparier l'entrée avec le stimulus
  - Générer la réponse à partir de la réaction et de l'appariement
  - Afficher la réponse
  - Lire l'entrée suivante

- Amélioration avec des mots-clés à détecter
  - pere, mere, frere….

De quoi va-t-on parler ?

- Les débuts
  - Avec Eliza
- Résolution par satisfaction de contraintes (CSP)
  - Avec les N-Reines
- Recherche dans un graphe d'états
  - Avec le labyrinthe
- Décomposition de problèmes
  - Avec les Tours de Hanoi

CSP

- $X = X_1 , X_2 , ..., X_n$ l'ensemble des variables caractérisant le problème
- $D(X_i)$ le domaine de chaque variable X_i = l'ensemble des valeurs que X_i peut prendre théoriquement
- $C = C_1 , C_2 , ..., C_k$ l'ensemble des contraintes

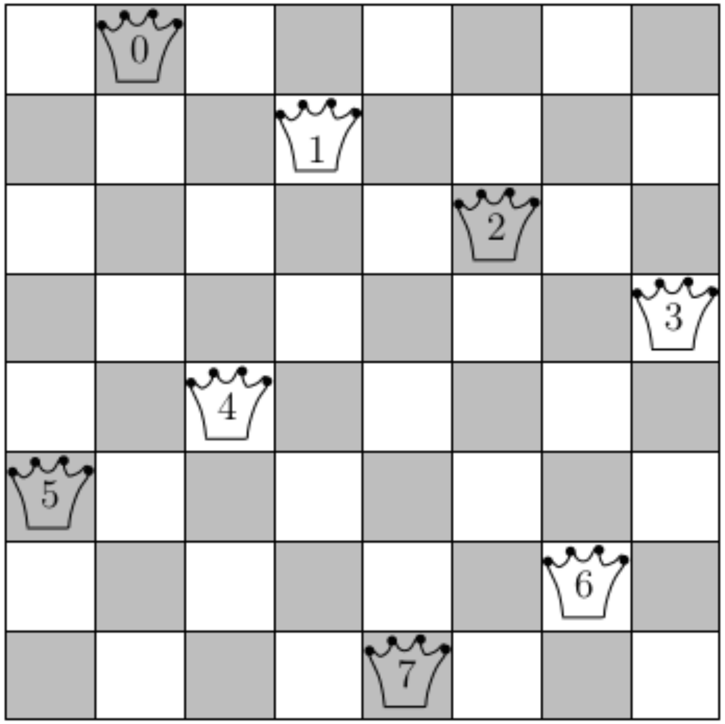
- Chaque contrainte C_j est une relation entre certaines variables de X restreignant les valeurs que peuvent prendre simultanément ces variables

Problème des N reines

Placer N reines sur un échiquier (une grille N x N)

tel qu'aucune reine attaque une autre reine

c'est-à-dire qu'il n'y a pas deux reines sur la même colonne,
la même ligne, ou sur la même diagonale



4 reines : 1ère modélisation

- Les variables
  - Associer à chaque reine i deux variables Li (sa ligne) et Ci (sa colonne)
    - X = {L1, L2, L3, L4, C1, C2, C3, C4}
- Les domaines de valeurs
    - D(L1) = D(L2) = D(L3) = D(L4) = D(C1) = D(C2) = D(C3) = D(C4) = {1,2,3,4}

- Les contraintes :
  - Les reines doivent être sur des lignes différentes.
    - Clig = {L1≠L2, L1≠L3, L1≠L4, L2≠L3, L2≠L4, L3≠L4 }
  - Les reines doivent être sur des colonnes différentes.
    - Ccol = {C1≠C2, C1≠C3, C1≠C4, C2≠C3, C2≠C4, C3≠C4 }
  - Les reines doivent être sur des diagonales montantes différentes.
    - Cdm = {C1+L1≠C2+L2, C1+L1≠C3+L3, C1+L1≠C4+L4, C2+L2≠C3+L3, C2+L2≠C4+L4, C3+L3≠C4+L4}
  - Les reines doivent être sur des diagonales descendantes différentes.
    - Cdd = {C1-L1≠C2-L2, C1-L1≠C3-L3, C1-L1≠C4-L4, C2-L2≠C3-L3, C2-L2≠C4-L4, C3-L3≠C4-L4}
  - L'ensemble des contraintes est défini par l'union de ces 4 ensembles :
    - C = Clig U Ccol U Cdm U Cdd

4 reines : 1$^{\text{ère}}$ modélisation

```
?- reines4_v1(R).

R = [1, 2, 3, 4, 2, 4, 1, 3] ;
R = [1, 2, 3, 4, 3, 1, 4, 2] ;
R = [1, 2, 4, 3, 2, 4, 3, 1] ;
R = [1, 2, 4, 3, 3, 1, 2, 4] ;
R = [1, 3, 2, 4, 2, 1, 4, 3] ;
R = [1, 3, 2, 4, 3, 4, 1, 2] ;
R = [1, 3, 4, 2, 2, 1, 3, 4] ;
R = [1, 3, 4, 2, 3, 4, 2, 1] ;
R = [1, 4, 2, 3, 2, 3, 4, 1] ;
R = [1, 4, 2, 3, 3, 2, 1, 4] ;
R = [1, 4, 3, 2, 2, 3, 1, 4] ;
R = [1, 4, 3, 2, 3, 2, 4, 1] ;
R = [2, 1, 3, 4, 1, 3, 4, 2] ;
R = [2, 1, 3, 4, 4, 2, 1, 3] ;
R = [2, 1, 4, 3, 1, 3, 2, 4] ;
R = [2, 1, 4, 3, 4, 2, 3, 1] ;
R = [2, 3, 1, 4, 1, 4, 3, 2] ;
R = [2, 3, 1, 4, 4, 1, 2, 3] ;
R = [2, 3, 4, 1, 1, 4, 2, 3] ;
R = [2, 3, 4, 1, 4, 1, 3, 2] ;
R = [2, 4, 1, 3, 1, 2, 3, 4] ;
R = [2, 4, 1, 3, 4, 3, 2, 1] ;
R = [2, 4, 3, 1, 1, 2, 4, 3] ;
R = [2, 4, 3, 1, 4, 3, 1, 2] ;
```

```
R = [3, 1, 2, 4, 1, 2, 4, 3] ;
R = [3, 1, 2, 4, 4, 3, 1, 2] ;
R = [3, 1, 4, 2, 1, 2, 3, 4] ;
R = [3, 1, 4, 2, 4, 3, 2, 1] ;
R = [3, 2, 1, 4, 1, 4, 2, 3] ;
R = [3, 2, 1, 4, 4, 1, 3, 2] ;
R = [3, 2, 4, 1, 1, 4, 3, 2] ;
R = [3, 2, 4, 1, 4, 1, 2, 3] ;
R = [3, 4, 1, 2, 1, 3, 2, 4] ;
R = [3, 4, 1, 2, 4, 2, 3, 1] ;
R = [3, 4, 2, 1, 1, 3, 4, 2] ;
R = [3, 4, 2, 1, 4, 2, 1, 3] ;
R = [4, 1, 2, 3, 2, 3, 1, 4] ;
R = [4, 1, 2, 3, 3, 2, 4, 1] ;
R = [4, 1, 3, 2, 2, 3, 4, 1] ;
R = [4, 1, 3, 2, 3, 2, 1, 4] ;
R = [4, 2, 1, 3, 2, 1, 3, 4] ;
R = [4, 2, 1, 3, 3, 4, 2, 1] ;
R = [4, 2, 3, 1, 2, 1, 4, 3] ;
R = [4, 2, 3, 1, 3, 4, 1, 2] ;
R = [4, 3, 1, 2, 2, 4, 3, 1] ;
R = [4, 3, 1, 2, 3, 1, 2, 4] ;
R = [4, 3, 2, 1, 2, 4, 1, 3] ;
R = [4, 3, 2, 1, 3, 1, 4, 2] ;
false.
```

Enorme combinatoire…

4 reines : $2^{ème}$ modélisation

- Intégrer une partie des contraintes :
  - On ne peut pas avoir deux reines sur la même colonne

- Les variables
  - Associer à chaque reine i une variable Li, sa colonne étant fixée
    - X = {L1, L2, L3, L4}
- Les domaines de valeurs
    - D(L1) = D(L2) = D(L3) = D(L4) = {1,2,3,4}

- Les contraintes :
  - les reines doivent être sur des lignes différentes
    - Clig = {Li ≠ Lj / i élément_de {1,2,3,4}, j élément_de {1,2,3,4} et i ≠ j}
  - les reines doivent être sur des diagonales montantes différentes
    - Cdm = {Li+i ≠ Lj+j / i élément_de {1,2,3,4}, j élément_de {1,2,3,4} et i ≠ j}
  - les reines doivent être sur des diagonales descendantes différentes
    - Cdd = {Li-i ≠ Lj-j / i élément_de {1,2,3,4}, j élément_de {1,2,3,4} et i ≠ j}
  - L'ensemble des contraintes est défini par l'union de ces 3 ensembles
    - C = Clig U Cdm U Cdd

4 reines : 3ème modélisation

- Les variables
  - Non pas les positions des reines
  - Mais les états des 16 cases de l'échiquier Xij (ligne i et colonne j)
    - X = { X11, X12, X13, X14, X21, X22, X23, X24, X31, X32, X33, X34, X41, X42, X43, X44}
- Les domaines de valeurs
  - chaque variable peut prendre pour valeur 0 (pas de reine sur la case) ou 1
    - D(Xij) = {0,1} pour tout i et tout j compris entre 1 et 4

- Les contraintes :
  - les reines doivent être sur des lignes différentes
    - Clig = {Xi1 + Xi2 + Xi3 + Xi4 = 1 / i élément_de {1,2,3,4}}
  - les reines doivent être sur des colonne différentes
    - Ccol = {X1i + X2i + X3i + X4i = 1 / i élément_de {1,2,3,4}}
  - les reines doivent être sur des diagonales montantes différentes
    - Ccol = pour tout couple de variables différentes Xij et Xkl, i+j=k+l => Xij + Kkl ≤ 1
  - les reines doivent être sur des diagonales descendantes différentes
    - Cdd = pour tout couple de variables différentes Xij et Xkl, i-j=k-l => Xij + Kkl ≤ 1
  - L'ensemble des contraintes est défini par l'union de ces 3 ensembles
    - C = Clig U Ccol U Cdm U Cdd

Généralisation à N reines

- Par exemple, la deuxième modélisation devient :

- Variables :
  - L = {Li / i est un entier compris entre 1 et n}
- Domaines :
  - Quelque soit Li élément de L,
    D(Li) = {j / j est un entier compris entre 1 et n}
- Contraintes :
  - les reines doivent être sur des lignes différentes
    - Clig = {Li ≠ Lj / i et j sont 2 entiers différents compris entre 1 et n}
  - les reines doivent être sur des diagonales montantes différentes
    - Cdm = {Li+i ≠ Lj+j / i et j sont 2 entiers différents compris entre 1 et n}
  - les reines doivent être sur des diagonales descendantes différentes
    - Cdd = {Li-i ≠ Lj-j / i et j sont 2 entiers différents compris entre 1 et n}
  - L'ensemble des contraintes est défini par l'union de ces 3 ensembles
    - C = Clig U Cdm U Cdd

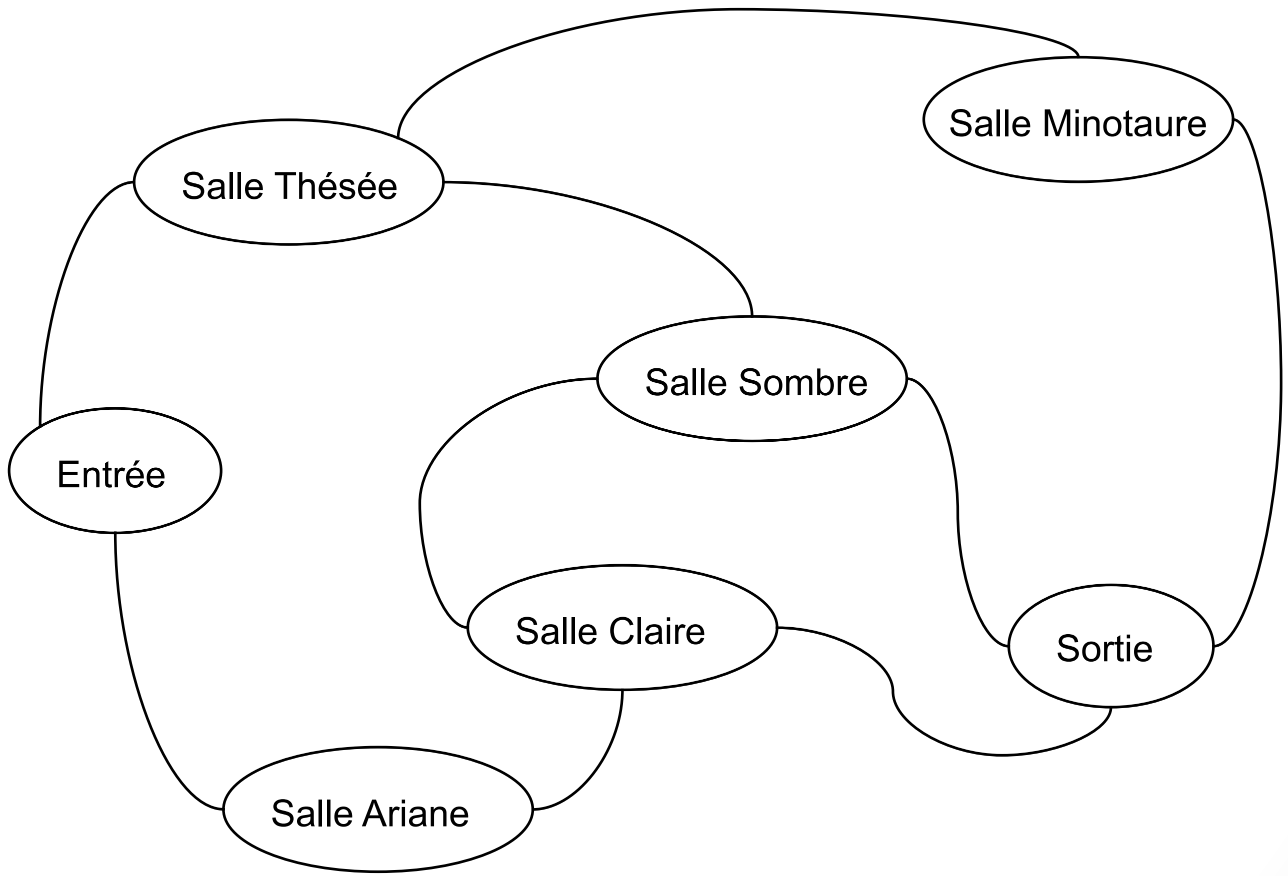
CSP : généralisation

- Prédicats de résolution génériques
- Nécessitent la définition pour chaque problème des prédicats
  - domaines(L)
    qui génère la liste L des couples
    [Variable,Domaine de la variable]
  - consistants(affectation1,affectation2)
    si les deux affectations sont compatibles

De quoi va-t-on parler ?

- Les débuts
  - Avec Eliza
- Résolution par satisfaction de contraintes (CSP)
  - Avec les N-Reines
- Recherche dans un graphe d'états
  - Avec le labyrinthe
- Décomposition de problèmes
  - Avec les Tours de Hanoi

Un problème de labyrinthe



Généralisation : recherche dans un graphe d'états

- On définit pour chaque problème :
  - Un état initial (ou plusieurs)
  - Un état final (ou plusieurs)
  - Des états interdits (éventuellement)
  - Des opérateurs de transition

- On définit un algorithme général de recherche dans le graphe ainsi construit

Algorithme de recherche

Liste des états
déjà rencontrés

État initial → État courant

Opérateur

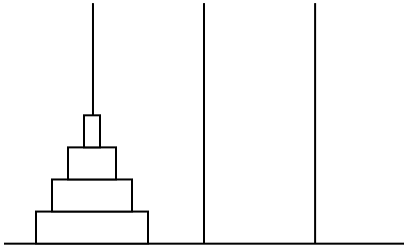
État suivant → État final

Liste des
opérateurs à
appliquer

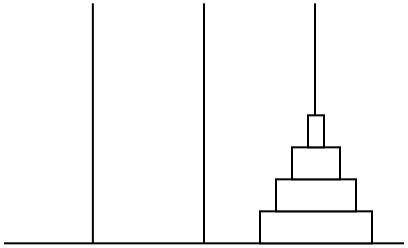
De quoi va-t-on parler ?

- Les débuts
  - Avec Eliza
- Résolution par satisfaction de contraintes (CSP)
  - Avec les N-Reines
- Recherche dans un graphe d'états
  - Avec le labyrinthe
- Décomposition de problèmes
  - Avec les Tours de Hanoi

Les tours de Hanoï

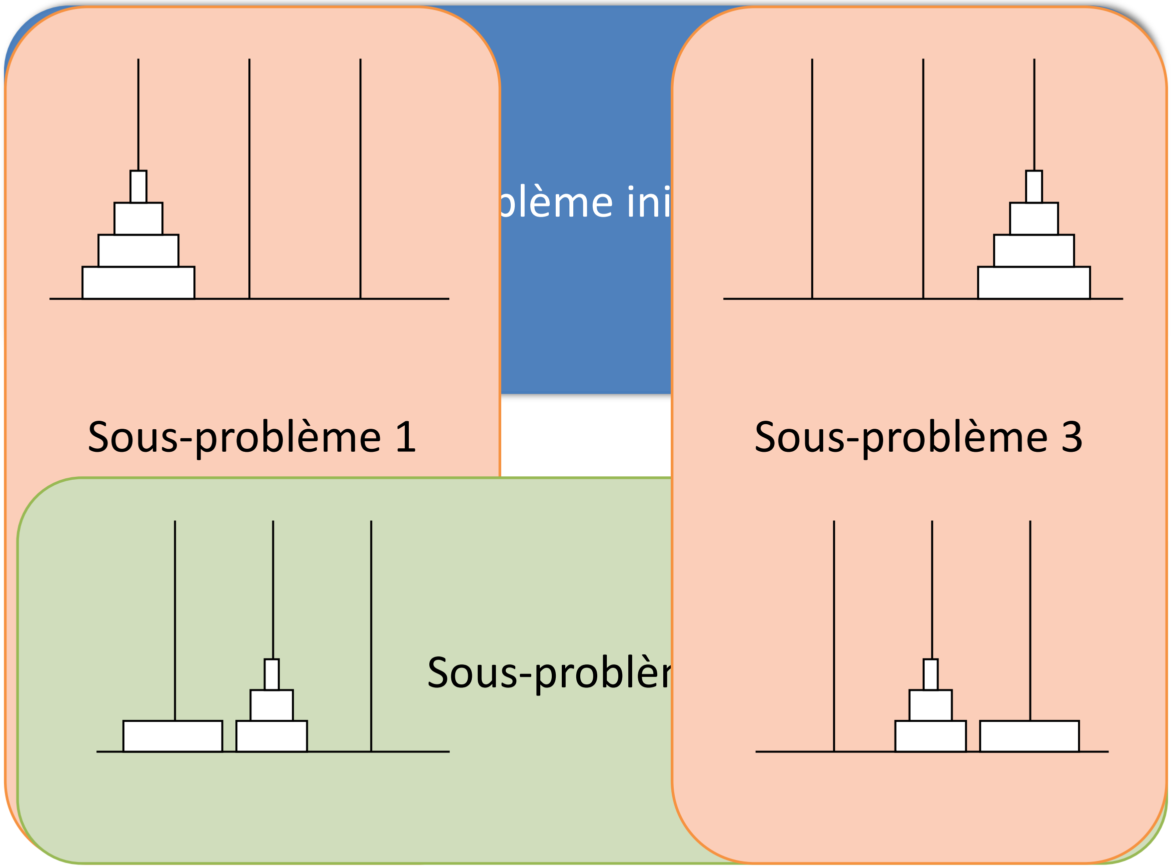


Situation initiale



Objectif

Décomposition un problème en sous-problèmes



De quoi va-t-on parler ?

- Les débuts
  - Avec Eliza
- Résolution par satisfaction de contraintes (CSP)
  - Avec les N-Reines
- Recherche dans un graphe d'états
  - Avec le labyrinthe
- Décomposition de problèmes
  - Avec les Tours de Hanoi
- Pour aller plus loin : les SBC …