

IA41

**Concepts fondamentaux en Intelligence Artificielle
et langages dédiés**

CM #7

**Problèmes de satisfaction
de contraintes**

Fabrice LAURI

Problèmes de satisfaction de contraintes

- Introduction
- Définitions
- Exemples de CSP : fabrication sous contraintes d'un modèle de voiture, les N dames
- Algorithmes de résolution de CSP :
 - *generate & test*,
 - *test & generate (Backtrack)*
- Présentation et utilisation de la librairie PROLOG *clp/bounds* pour la résolution d'un CSP

Problèmes de satisfaction de contraintes

- **Introduction**

- **Définitions**

- **Exemples de CSP : fabrication sous contraintes d'un modèle de voiture, les N dames**

- **Algorithmes de résolution de CSP :**

 - *generate & test*,

 - *test & generate (Backtrack)*

- **Présentation et utilisation de la librairie PROLOG *clp/bounds* pour la résolution d'un CSP**

Introduction

Les CSP (*Constraint Satisfaction Problems*) regroupent :

- une partie des problèmes d'Intelligence Artificielle (satisfiabilité d'un ensemble de clauses...)
- une partie des problèmes de Recherche Opérationnelle (problèmes d'ordonnancement...)

Définition d'une contrainte

Enoncé d'une propriété relative à un ou plusieurs objets.

Exemple : « $x = y+z$ »

Résolution d'un CSP

Trouver l'ensemble des valeurs à affecter aux variables du problème, ces valeurs devant satisfaire chacune des contraintes.

Problèmes de satisfaction de contraintes

- Introduction
- Définitions
- Exemples de CSP : fabrication sous contraintes d'un modèle de voiture, les N dames
- Algorithmes de résolution de CSP :
 - *generate & test*,
 - *test & generate (Backtrack)*
- Présentation et utilisation de la librairie PROLOG *clp/bounds* pour la résolution d'un CSP

Définition d'un CSP

Un CSP $P = \{ X, D, C \}$ est défini par :

- une séquence $X = (x_1, x_2, \dots, x_n)$ de n variables
- une séquence $D = (d_1, d_2, \dots, d_n)$ de n domaines finis. Chaque d_i est associé à x_i .
- une séquence $C = (c_1, c_2, \dots, c_m)$ de m contraintes. Chaque contrainte c_i est définie par un couple (v_i, r_i) tel que :
 - v_i est une séquence de variables tel que $v_i \subset X$ sur lesquelles porte la contrainte c_i . Elle est exprimée en extension.
 - r_i est un sous-ensemble du produit cartésien des domaines associés aux variables de v_i . Il est exprimé en extension.

Exemple

Soit le CSP $P = \{ X, D, C \}$ défini par :

| Variables | Domaines |
|-----------|-----------------|
| x_1 | $\{ a, b, c \}$ |
| x_2 | $\{ a, b, c \}$ |
| x_3 | $\{ a, b \}$ |
| x_4 | $\{ a, b \}$ |
| x_5 | $\{ a, b \}$ |

| Contrainte | Variables | Relation |
|--------------------|--------------------|---------------------------------|
| $c_1 = (v_1, r_1)$ | $v_1 = (x_3, x_1)$ | $r_1 = \{(a,b), (a,c), (b,c)\}$ |
| $c_2 = (v_2, r_2)$ | $v_2 = (x_3, x_2)$ | $r_2 = \{(a,b), (a,c), (b,c)\}$ |
| $c_3 = (v_3, r_3)$ | $v_3 = (x_3, x_4)$ | $r_3 = \{(a,b)\}$ |
| $c_4 = (v_4, r_4)$ | $v_4 = (x_3, x_5)$ | $r_4 = \{(a,b)\}$ |
| $c_5 = (v_5, r_5)$ | $v_5 = (x_4, x_5)$ | $r_5 = \{(a,b)\}$ |

Définitions (1/3)

Instanciation

Soit un CSP $P = \{ X, D, C \}$. Une instanciation A de $Y \subset X$ est une application qui associe à chaque variable de Y une valeur de son domaine.

On notera $A(V)$ la séquence des valeurs affectées aux variables de la séquence $V \subset Y$ par A .

Si $X = Y$, on dira que A est complète.

Sinon, on dira que A est partielle.

Satisfaction de contraintes

Une instanciation A de Y satisfait la contrainte $c_i = (v_i, r_i)$ de C , noté $A \models c_i$, ssi $v_i \subset Y$ et $A(v_i) \in r_i$.

Taux de satisfiabilité d'une contrainte c_i

Rapport entre le cardinal de r_i et le cardinal du produit cartésien des domaines des variables de v_i .

Définitions (2/3)

Instanciation consistante

Soient un CSP $P = \{ X, D, C \}$ et une instanciation A de $Y \subset X$.
 A est dite consistante ssi elle ne viole aucune contrainte, c-à-d :

$$\forall c_i = (v_i, r_i) \in C \text{ telle que } v_i \in Y, A \models c_i.$$

Solution d'un CSP

Une solution S de $P = \{ X, D, C \}$ est une instanciation complète consistante. On dit que l'instanciation S satisfait P , ou $S \models P$.
L'ensemble des solutions de P sera noté S_P .

Consistance d'un CSP

$P = \{ X, D, C \}$ est consistant ssi $S_P \neq \emptyset$.

Définitions (3/3)

Instanciation globalement consistante

Soit $P = \{ X, D, C \}$. Une instanciation A de $Y \subset X$ est globalement consistante ssi $\exists S \in SP$ telle que $A \subset S$.

Equivalence de CSP

Les CSP $P = \{ X, D, C \}$ et $P' = \{ X', D', C' \}$ sont équivalents ssi $S_P = S_{P'}$. On note alors $P \equiv P'$.

Problèmes de satisfaction de contraintes

- Introduction
- Définitions
- Exemples de CSP : fabrication sous contraintes d'un modèle de voiture, les N dames
- Algorithmes de résolution de CSP :
 - *generate & test*,
 - *test & generate (Backtrack)*
- Présentation et utilisation de la librairie PROLOG *clp/bounds* pour la résolution d'un CSP

Exemple #1 : fabrication sous contraintes d'une voiture

« Une firme va proposer un nouveau modèle de voiture fabriquée dans toute l'Europe :

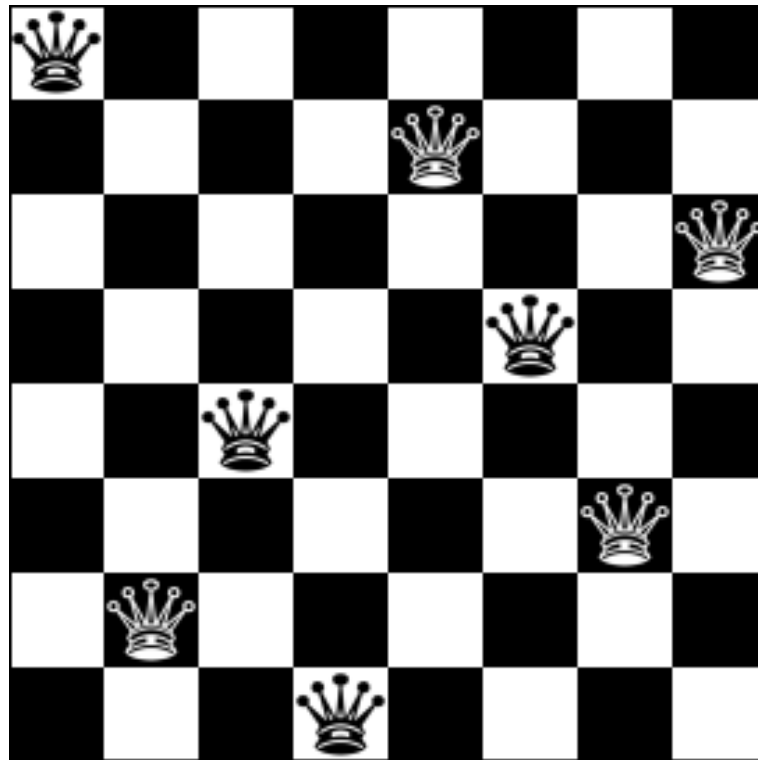
- les *portières* et le *capot* sont fabriqués et peints à Lille, où le constructeur ne dispose que de peinture *rose*, *rouge* et *noire*;
- la *carrosserie* est réalisée à Hambourg avec de la peinture *blanche*, *rose*, *rouge* ou *noire*;
- les *pare-chocs* sont peints uniquement en *blanc* à Palerme;
- la *bâche du toit ouvrant* est peinte à Madrid en *rouge*;
- les *enjolveurs* sont fabriqués et peints à Athènes, où l'on ne dispose que de peinture *rose* et *rouge*.

Le constructeur impose un certain agencement des couleurs :

- la *carrosserie* doit être de la même couleur que les *portières*, les *portières* de la même couleur que le *capot*, le *capot* de la même couleur que la *carrosserie*
- les *enjolveurs*, les *pare-chocs* et le *toit ouvrant* doivent être plus clairs que la *carrosserie*. »

Exemple #2 : les N dames

« Déterminer la position de N dames sur un échiquier de NxN cases. Aucune des N dames ne doit être attaquée par une autre. »



Une dame attaque sur sa ligne, sa colonne et sa/ses diagonale(s).

Problèmes de satisfaction de contraintes

- Introduction
- Définitions
- Exemples de CSP : fabrication sous contraintes d'un modèle de voiture, les N dames
- Algorithmes de résolution de CSP :
 - *generate & test*,
 - *test & generate (Backtrack)*
- Présentation et utilisation de la librairie PROLOG *clp/bounds* pour la résolution d'un CSP

Premier algorithme de résolution : *generate & test*

Principe

On considère un espace où chaque état est une instantiation A . Passer d'un état e_1 à un état e_2 consiste à étendre l'instanciation de e_1 en y ajoutant une variables avec toutes les valeurs possibles de son domaine.

Une feuille de l'arbre : instantiation complète, dont il est facile de vérifier si elle représente une solution ou non.

Remarque

Sur un CSP de n variables, avec des domaines de cardinal d et m contraintes, l'ensemble des états terminaux (feuilles) a un cardinal de d^m , et on peut être amené à évaluer $m * d^m$ contraintes...

Deuxième algorithme de résolution : *test & generate (Backtrack)*

Remarque

Si une instanciation viole une contrainte, toutes ses instanciations filles ne seront pas consistante.

Toute affectation non consistante est non globalement consistante.

Principe

Vérifier la satisfaction d'une contrainte dès que toutes ses variables ont été instanciées.

Lors du parcours en profondeur d'abord, dès qu'une impasse est rencontrée (contrainte violée), un retour arrière est réalisé sur le noeud père.

Deuxième algorithme de résolution : *test & generate (Backtrack)*

Algorithme

Backtrack(V , A)

Si $V = \emptyset$ Alors

A est une solution

Sinon

 Soit $x_i \in V$

 Pour chaque $v \in d_i$ Faire

 Si $A \cup \{ x_i \rightarrow v \}$ est consistante Alors

 Backtrack($V - \{ x_i \}$, $A \cup \{ x_i \rightarrow v \}$)

 FinSi

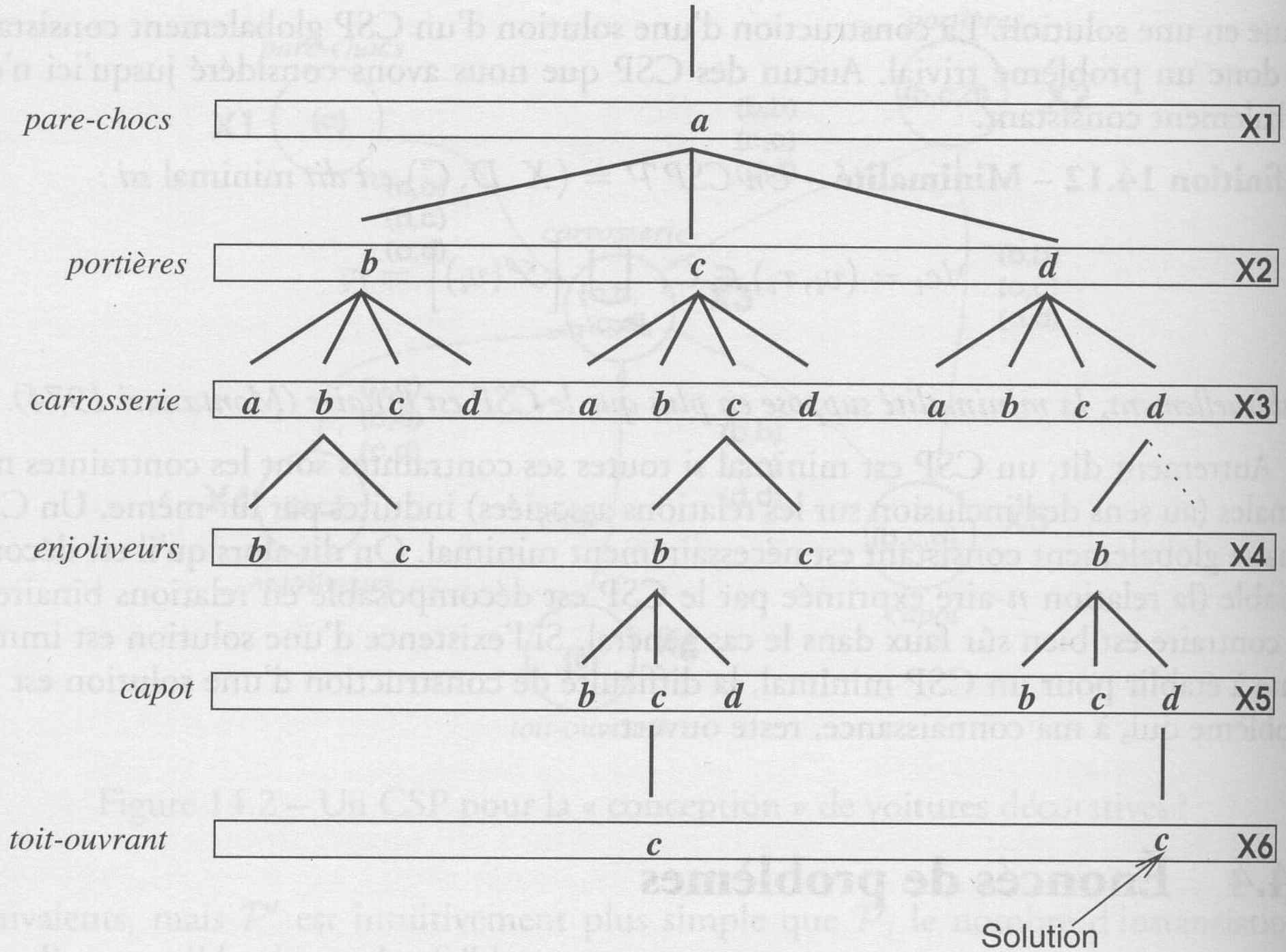
 FinPour

FinSi

V : séquence de variables
à instancier, initialement X .

A : instantiation, initialement
vide.

Application de *BT* sur le CSP #1



Pour améliorer l'algorithme *BT*

L'ordre d'instanciation des variables (ordre vertical) influe sur la taille de l'espace exploré.

Heuristiques souvent utilisées :

- choix de la variable de plus petit domaine d'abord
- choix de la variable participant au plus grand nombre de contraintes d'abord
- choix de la variable participant à la contrainte la moins satisfiable d'abord

Le nombre de noeuds explorés et le nombre de contraintes vérifiées dépendent de l'heuristique utilisée.

Problèmes de satisfaction de contraintes

- Introduction
- Définitions
- Exemples de CSP : fabrication sous contraintes d'un modèle de voiture, les N dames
- Algorithmes de résolution de CSP :
 - *generate & test*,
 - *test & generate (Backtrack)*
- Présentation et utilisation de la librairie PROLOG *clp/bounds* pour la résolution d'un CSP

Librairie 'clp/bounds' (1/4)

Cette librairie contient des prédicats qui facilite la résolution d'un CSP P de la forme $P = \{ X, D, C \}$.

Prédicats de définition des domaines

Var in +Range

Le domaine de valeur de *Var* est *Range*. *Range* est de la forme $I..S$, qui représente l'intervalle $[I; S]$.

Vars in +Range

Chaque variable de la liste de variables *Vars* a pour domaine *Range*.

Librairie 'clp/bounds' (2/4)

Prédicats permettant d'établir des contraintes

all_different(+Vars)

Toutes les variables *Vars* sont contraintes de telle sorte à être deux à deux différentes.

tuples_in(+Tuples, +Extension)

Permet de poser une contrainte c_i de la forme $c_i = (v_i, r_i)$.

Tuples est une liste composée de listes de N variables, *Extension* est la liste des relations devant être satisfaites par les variables définies dans *Tuples*.

Exemple :

```
tuples_in( [[X1,X2],[X2,X3]], [[1,2],[3,2]] ).
```

Librairie 'clp/bounds' (3/4)

Prédicats permettant d'établir des contraintes

?E1 #Op ?E2, avec $Op := \{ =, \neq, >, <, \geq, \leq \}$

Les expressions E1 et E2 sont contraintes selon l'opérateur Op.

Exemples :

- $X1 \neq 3$
- $X2 \geq X4 + 5$
- $X3 \neq X5$

...

Librairie 'clp/bounds' (4/4)

Prédicats de résolution

`label(+Vars)`

Toutes les variables de *Vars* sont instanciées avec des valeurs de leurs domaines respectifs et telles que toutes les contraintes soient satisfaites.