Programmation par contraintes
Cours 2 : Problèmes de Satisfaction de Contraintes
Modélisation d'un CSP

Odile PAPINI

POLYTECH
Université d'Aix-Marseille
odile.papini@univ-amu.fr
http://odile.papini.perso.luminy.univ-amu.fr/sources/PCC.html

Plan du cours 2

Bibliographie

Livres :

- **K. Marriott and P. Stuckey. Programming with constraints. MIT Press 1998**
- **F. Fages. Programming Logique par contraintes. Ellipes, 1996**
- **K. R. Apt. Principles in Constraint Programming. Cambridge Univ Press, 2003**

Supports de cours :

- **Support de cours : Christine SOLNON https ://perso.liris.cnrs.fr/christine.solnon/Site-PPC/e-miage-ppc-som.htm**
- **Roman Bartak Charles university : http ://kti.mff.cuni.cz/ bartak/constraints/**

Problèmes de Satisfaction de Contraintes (CSP)

Définition

un CSP est un problème modélisé sous forme de contraintes posées sur des variables prenant valeur dans un domaine

un CSP est quadruplet (X, D, C, R) où :

- $X = \{X_1, \cdots, X_n\}$: un ensemble de variables
- $D = \{D_1, \cdots, D_n\}$: un ensemble de domaines,
  $X_i \in D_i$, $1 \leq i \leq n$
- $C = \{C_1, \cdots, C_m\}$: un ensemble de contraintes,
- $R = \{R_1, \cdots, R_m\}$: un ensemble de relations,
  à chaque contrainte C_j est associée une relation R_j, $1 \leq j \leq m$

Problèmes de Satisfaction de Contraintes (CSP)

Exemple

- $X = \{X_1, X_2, X_3, X_4\}$
- $D = \{D_1, D_2, D_3, D_4\}$ avec $D_1 = D_2 = D_3 = D_4 = \{0, 1\}$
- $C = \{X_1 \neq X_2, X_3 \neq X_4, X_1 + X_3 < X_2\}$
- $R = \{R_1, R_2, R_3\}$

Les relations asociées aux contraintes de C :

R_1	
X_1	X_2
0	1
1	0

R_2	
X_3	X_4
0	1
1	0

R_3		
X_1	X_2	X_3
0	1	0

Modélisation de problèmes en termes de CSP

Modélisation d'un problème

- identifier les variables : les inconnues
- identifier les domaines de valeur de ces variables
- identifier les contraintes

souvent plusieurs modélisations possibles, critères de choix :

- simplicité d'expression
- efficacité : taille de l'espace de recherche de solutions

Modélisation de problèmes en termes de CSP

Exemple : le problème des 4 reines

Placer 4 reines sur échiquier 4×4 de telle sorte qu'aucune ne soit en prise (pas sur la même ligne, pas sur la même colonne, pas sur la même diagonale)

- $X =$?
- $D =$?
- $C =$?

Modélisation de problèmes en termes de CSP

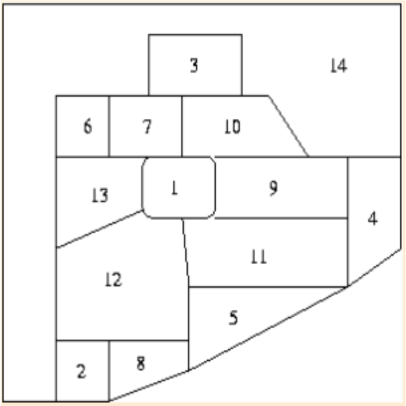
Exemple : le problème des 4 reines

Placer 4 reines sur échiquier 4×4 de telle sorte qu'aucune ne soit en prise (pas sur la même ligne, pas sur la même colonne, pas sur la même diagonale)

- $X = \{X_1, X_2, X_3, X_4\}$
- $D = \{D_1, D_2, D_3, D_4\}$ avec
  $D_1 = D_2 = D_3 = D_4 = \{1, 2, 3, 4\}$
- $C = \{C_{LI}, C_{DA}, C_{DD}\}$ avec
  - $C_{LI} = \{X_i \neq X_j,\ i \neq j,\ i \in \{1,2,3,4\}, j \in \{1,2,3,4\}\}$
  - $C_{DA} = \{X_i + i \neq X_j + j,\ i \neq j,\ i \in \{1,2,3,4\}, j \in \{1,2,3,4\}\}$
  - $C_{DD} = \{X_i - i \neq X_j - j,\ i \neq j,\ i \in \{1,2,3,4\}, j \in \{1,2,3,4\}\}$

Modélisation efficace

Exemple de modélisation de problèmes en termes de CSP



source C. Solnon

Coloriage d'une carte

attribuer une couleur à chaque région t. q.

deux régions voisines soient de couleurs différentes

4 couleurs : bleu, rouge, jaune, vert

Exemple de modélisation de problèmes en termes de CSP

Coloriage d'une carte

- $X =$?
- $D =$?
- $R =$?
- $C =$?

Exemple de modélisation de problèmes en termes de CSP

Coloriage d'une carte

- $X = \{X_1, X_2, X_3, X_4, \cdots, X_{14}\}$
- $D = \{D_1, D_2, D_3, D_4\}$ avec
  $D_1 = D_2 = D_3 = D_4 = \{bleu, rouge, vert, jaune\}$
- $R = \{voisines(X, Y)\}$ avec $voisines(X, Y)i$ ssi $(X, Y) \in \{(1, 7), (1, 9), (1, 10), (1, 11), (1, 12), (1, 13), \cdots (14, 13)\}$
- $C = \{X_i \neq X_j,\ i \neq j,$
  $i \in \{1, \cdots, 14\}, j \in \{1, \cdots, 14\}, voisines(X_i, X_j)\}$

Résolution d'un CSP

Résolution : affectation de valeurs aux variables de telle sorte que toutes les contraintes soient satisfaites

- affectation : instanciation des variables sur les domaines
- affectation totale : affectation de toutes les variables
- affectation partielle : affectation de certaines variables
- affectation consistante : affectation qui ne viole aucune contrainte
- affectation inconsistante : affectation qui viole au moins une contrainte
- solution : affectation totale et consistante

Résolution d'un CSP

Exemple

- $X = \{X_1, X_2, X_3, X_4\}$
- $D = \{D_1, D_2, D_3, D_4\}$ avec $D_1 = D_2 = D_3 = D_4 = \{0, 1\}$
- $C = \{X_1 \neq X_2, X_3 \neq X_4, X_1 + X_3 < X_2\}$

- $A = \{(X_2, 0), (X_3, 1)\}$ affectation
- $A = \{(X_1, 0), (X_2, 0), (X_3, 0), (X_4, 0)\}$ affectation totale
- $A = \{(X_1, 0), (X_2, 0)\}$ affectation partielle
- $A = \{(X_1, 0), (X_2, 0)\}$ affectation inconsistante
- $A = \{(X_3, 0), (X_4, 1)\}$ affectation consistante
- $A = \{(X_1, 0), (X_2, 1), (X_3, 0), (X_4, 1)\}$ solution

Résolution d'un CSP

Résolution : affectation de valeurs aux variables de telle sorte que toutes les contraintes soient satisfaites
complexité : en général problème NP-difficile
solution : affectation totale et consistante

On peut chercher :

- une solution, n'importe laquelle
- toutes les solutions
- une solution optimale ou au moins une bonne solution selon une fonction de coût ou d'objectif (Problème d'Optimisation de Contraintes (COP))

Intérêt des CSP

Intérêt des CSP par rapport à la programmation mathématique

- la représentation des CSP est plus proche des problèmes originaux, la formulation est plus simple
- les algorithmes de résolution des CSP sont relativement simples et sont plus rapides que ceux de la programmation en nombres entiers

Binarisation des contraintes

Les contraintes peuvent être n-aires : 2 cas particuliers

- les contraintes unaires

  peuvent être traitées par un pré-traitement sur les domaines

- les contraintes binaires

  si toutes les contraintes sont binaires : représentation par un graphe dont les sommets sont les variables et les arcs sont les contraintes.

- toute contrainte n-aire peut s'exprimer en termes de contraintes binaires

Binarisation des contraintes

- toute contrainte n-aire peut s'exprimer en termes de contraintes binaires
- tout CSP peut se représenter comme un CSP binaire
- un CSP binaire peut être représenté comme un graphe de contraintes

exemples

Binarisation d'un CSP

Méthode 1 : conservation de variables initiales

- création d'une variable encapsulée U

  domaine de U : produit cartésien des variables encapsulées

- application des contraintes sur les variables encapsulées pour réduire le domaine

- ajout d'une binaire entre la variable originale et la variable encapsulée : $X_i = pos_i(U)$

  $i$ème position de X_i dans U : $pos_i(U)$

Binarisation d'un CSP

Méthode 1 : exemple

- $X = \{X_1, X_2, X_3\}$
- $D = \{D_1, D_2, D_3\}$ avec $D_1 = \{1, 2\}$, $D_2 = \{3, 4\}$, $D_3 = \{5, 6\}$
- $C = \{X_1 + X_2 = X_3, X_1 < X_2\}$

CSP binaire : $X' = \{X_1, X_2, X_3\ U\}$, $D' = \{D_1, D_2, D_3, D'_U\}$
création de la variable $U = \{(X_1, X_2, X_3)$, tq $\{X_1 + X_2 = X_3\}$,
$D_U = D_1 \times D_2 \times D_3$
réduction de domaine : D'_U
nouvel ensemble de containtes : C'

Binarisation d'un CSP

Méthode 2 : sans conservation de variables initiales

- création de variables encapsulées

  domaine : produit cartésien des variables qu'elles encapsulent

- application des contraintes sur les variables encapsulées pour réduire le domaine

- ajout d'une binaire entre la variable originale et la variable encapsulée : $X_i = pos_i(U)$

  $i$ème position de la variable encapsulée U est la $j$ème position de la variable encapsulée V : $pos_i(U) = pos_j(V)$

Binarisation d'un CSP

Méthode 2 : exemple

- $X = \{X_1,\ X_2,\ X_3\}$
- $D = \{D_1,\ D_2,\ D_3\}$ avec $D_1 = \{1,2\}$, $D_2 = \{3,4\}$, $D_3 = \{5,6\}$
- $C = \{X_1 + X_2 = X_3, X_1 < X_2\}$

CSP binaire : $X' = \{U,\ V\}$, $D' = \{D'_U, D'_V\}$, C'
création de 2 variable $U = \{(X_1, X_2, X_3)$, tq $\{X_1 + X_2 = X_3\}$, et $V = \{(X_1, X_2)$, tq $\{X_1 < X_2\}$ $D_U = D_1 \times D_2 \times D_3$ et $D_V = D_1 \times D_2$
réduction de domaine : D'_U et D'_V
nouvel ensemble de containtes : C'

Résolution d'un CSP

hypothèse : domaines finis

algorithmes génériques de résolution de CSP

- algorithmes complets
- algorithmes incomplets

Autres algorithmes pour :

- CSP numériques linéaires sur les réels
- CSP numériques linéaires sur les entiers
- CSP numériques non linéaires

Algorithmes de résolution d'un CSP

algorithmes génériques de résolution de CSP

- recherche systématique
  - genere_et_teste (GET)
  - retour_arrière (SRA) ou (backtracking (BT))
- techniques de filtrage
  - consistance de noeud (NC), d'arc (AC), de chemin (PC) $\cdots$
- techniques de propagation de contraintes
  - forward_checking (FC)
  - look_ahead (LH)
- techniques basées sur l'ordre des variables et des valeurs
  - heuristiques