USTHB, FEI, Département d'Informatique

LMD Master 2 ''Systèmes Informatiques Intelligents'' 2020/2021

Module ''Programmation Par Contraintes''

Travaux Dirigés

Série numéro 2 : Résolution d'un CSP

Exercice 1

1) Donner une représentation graphique du CSP P=(X,D,C) suivant :
   - $X = \{X_1,X_2,X_3,X_4,X_5\}$
   - $D(X_1)=\{1,2,5,7\}$
   - $D(X_2)=\{2,3,8,12\}$
   - $D(X_3)=\{5,6,7,8\}$
   - $D(X_4)=\{1,2,3,4,5\}$
   - $D(X_5)=\{1,2,3,4,5,6,7\}$
   - $C = \{c_1,c_2,c_3,c_4,c_5,c_6,c_7,c_8\}$ avec
     - $c_1 : X_1=X_4$
     - $c_2 : X_5 \leq X_4$
     - $c_3 : X_2 \geq X_3$
     - $c_4 : X_2 \neq X_5$
     - $c_5 : X_1 < X_3$
     - $c_6 : X_3 > X_4$
     - $c_7 : 3*X_5 \geq X_3$
     - $c_8 : 2*X_5 = X_3$
2) En utilisant la procédure de propagation incrémentale de contraintes dans l'ordre croissant des indices des contraintes (c_1, c_2, …), rendre ce CSP consistant d'arc (arc-consistant). A chaque étape de cette procédure donner la liste des valeurs supprimées, les contraintes à reposer.
3) Dérouler l'algorithme Look-Ahead sur le CSP P.

Exercice 2

Calculer la complexité du pire cas de l'algorithme de consistance d'arc AC3.

Exercice 3

Soit le CSP binaire discret à variables entières P=(X,D,C) suivant :

- $X = \{X_1,X_2,X_3,X_4,X_5,X_6\}$
- $D(X_1)=\{ 0,1,2,3,9,10,11,12\}$
- $D(X_2)= \{-2,-1\}$
- $D(X_3)=\{ 5,6,7\}$
- $D(X_4) =\{-11,-10,-9,-2,-1,0\}$
- $D(X_5)=\{0,1,2,3\}$
- $D(X_6)=\{0,1,2\}$
- $C = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$ avec
    - $c_1 : X_1 = 2*X_4+3$
    - $c_2 : X_6 = X_1-2$
    - $c_3 : X_5 > X_2$
    - $c_4 : X_4+2 \geq X_6$
    - $c_5 : X_3 = X_5+4$
    - $c_6 : X_2 < X_6$

1) Donner une représentation graphique du CSP
2) Rendre le CSP arc-consistant. Donner les domaines arc-consistants obtenus
3) En partant des domaines rendus arc-consistants, trouver, si c'est possible, une solution de ce CSP par l'algorithme SRA dans l'ordre d'instanciation X_1, X_2, X_3, X_4, X_5, X_6

- Instancier les variables avec les valeurs choisies dans l'ordre croissant
- Donner l'arbre de recherche de solution de l'algorithme sur ce problème

4) Même question en utilisant l'algorithme du forward-checking FC avec un ordonnancement dynamique des variables construit en choisissant à chaque étape la variable X_i pour laquelle le quotient *cardinalité(domaine(* X_i*))/cardinalité(contraintes(* X_i*))* est le plus petit
5) Même question en utilisant l'algorithme Look_Ahead