Programmation par Contraintes

Module du Master “Systèmes Informatiques Intelligents” 2ème année

Annexe 2-1

Résolution d’un CSP binaire discret : consistance de chemin

Mr ISLI

Faculté d’Informatique

Département Intelligence Artificielle et Science des Données

Université des Sciences et de la Technologie Houari Boumediène

BP 32, El-Alia, Bab Ezzouar

DZ-16111 ALGER

https://perso.usthb.dz/~aisli/TA_PpC.htm

amar.isli@usthb.edu.dz

Annexe 2-1

Résolution d'un CSP binaire discret : consistance de chemin

Introduction

Cette annexe présente les deux algorithmes de consistance de chemin PC-1 et PC-2 ; de même que l'algorithme Look_Ahead avec PC-2 comme procédure de filtrage durant la recherche.

Annexe 2-1

Résolution d'un CSP binaire discret : consistance de chemin

Consistance de chemin (path-consistency)

Un CSP P=(X,D,C) est consistant de chemin si, pour tout chemin (Xi0, … ,Xij) de longueur j :

Xi0	Xi1	…	Xik	Xi(k+1)	…	Xij
V0	v1	…	vk	v(k+1)	…	vj

Pour tous (vo,vj) tels que unaires+binaires sur Xi0 et Xij satisfaites,

il existe v1, …, v(j-1) tels que pour tout k=0…(j-1):

(vk,v(k+1)) satisfait unaires+binaires sur Xik et Xi(k+1)

Annexe 2-1

Résolution d’un CSP binaire discret : consistance de chemin

consistance de chemin (path-consistency)

On montre facilement qu’on peut se restreindre aux chemins de longueur 2 (triplets de variables).

Annexe 2-1

Résolution d'un CSP binaire discret : consistance de chemin

consistance de chemin (path-consistency)

Un CSP P=(X,D,C) est consistant de chemin si :

Xi0	Xi1	Xi2
V0	v1	v2

Pour tous (v0,v2) tels que unaires+binaires sur Xi0 et Xi2 satisfaites, il existe v1 tel que :

- (v0,v1) satisfait unaires+binaires sur Xi0 et Xi1
- (v1,v2) satisfait unaires+binaires sur Xi1 et Xi2

v1 support de la paire (v0,v2)

Annexe 2-1

Résolution d'un CSP binaire discret : consistance de chemin

Principe d'un algorithme de consistance de chemin

Filtrage des paires de valeurs permises :

- pour chaque paire (X_i,X_j) de variables
  - supprimer les paires permises (vi,vj) n'ayant pas de support dans le domaine d'une troisième variable Xk

Xi	Xk	Xj
V0	v1	v2

Supprimer la paire permise (v0,v2) si elle n'a aucune valeur v1 de D(Xk) comme support.

Annexe 2-1

Résolution d'un CSP binaire discret : consistance de chemin

fonction REVISE_pc((X_i,X_k,X_j),(X,D,C)) : booléen

début

- temp=$M_P[i,j] \cap M_P[i,k] \circ M_P[k,k] \circ M_P[k,j]$
- **si** temp$\neq M_P[i,j]$ **alors**
    - $M_P[i,j]$=temp
    - retourner VRAI
- **sinon** retourner FAUX
- **finsi**

fin

Annexe 2-1

Résolution d'un CSP binaire discret : consistance de chemin

fonction PC1((X,D,C))

Début

$Q \leftarrow \{(X_i,X_k,X_j) \in X^3 : \neg(X_i=X_k=X_j)\}$

répéter

$R \leftarrow$ FALSE

pour tous les (X_i,X_k,X_j) de Q faire

si REVISE_pc((X_i,X_k,X_j),(X,D,C)) alors $R \leftarrow$ TRUE finsi

fin pour

jusqu'à non R

retourner (X,D,C)

fin

Annexe 2-1

Résolution d’un CSP binaire discret : consistance de chemin

procedure PC2(M_P) %version 1%

début

Q $\leftarrow$ $\{(X_i,X_k,X_j)\in X^3 : (i\leq j)$ **et** $\neg(i=k=j)\}$; entrée_nulle=false;

tant que ($Q\neq\emptyset$ **et** $\neg$entrée_nulle) **faire**

  Prendre un triplet (X_i,X_k,X_j) de variables de Q, et l’en supprimer ($Q \leftarrow Q\backslash\{(X_i,X_k,X_j)\}$);

  temp=$M_P[i,j]\cap M_P[i,k]\circ M_P[k,k]\circ M_P[k,j]$;

  si temp=[0] **alors** retourner FAUX **finsi**

  si temp$\neq M_P[i,j]$ **alors**

    $M_P[i,j]$=temp; $M_P[j,i]=(\text{temp})^t$;

    $Q=Q \cup \{(X_i,X_j,X_m)\in X^3 : (i\leq m)$ **et** $\neg(i=j=m)\} \cup \{(X_m,X_i,X_j)\in X^3 : (m\leq j)$ **et** $\neg(m=i=j)\}$

      $\cup \{(X_j,X_i,X_m)\in X^3 : (j\leq m)$ **et** $\neg(j=i=m)\} \cup \{(X_m,X_j,X_i)\in X^3 : (m\leq i)$ **et** $\neg(m=j=i)\}$

finsi **fait**

retourner VRAI;

fin

Annexe 2-1

Résolution d’un CSP binaire discret : consistance de chemin

procedure PC2(M_P){ %version 2%

début

$Q \leftarrow \{(X_i,X_j) \in X^2$: $(i \leq j)$ et il existe une contrainte entre X_i et $X_j\}$; entrée_nulle=faux

tant que $Q \neq \emptyset$ et $\neg$entrée_nulle **faire**

Prendre une paire (Xi,Xj) de variables de Q, et l’en supprimer ($Q \leftarrow Q \backslash \{(X_i,X_j)\}$); k=1

tant que $k \leq n$ et $\neg(k=i=j)$ et $\neg$entrée_nulle **faire**

temp=$M_P[i,k] \cap M_P[i,j] \circ M_P[j,j] \circ M_P[j,k]$; si temp=[0] alors entrée_nulle=true

si temp$\neq M_P[i,k]$ **alors**

$M_P[i,k]$=temp; $M_P[k,i]=(\text{temp})^t$

si $i \leq k$ **alors** $Q=Q \cup \{(X_i,X_k)\}$ **sinon** $Q=Q \cup \{(X_k,X_i)\}$ **finsi finsi**

temp=$M_P[k,j] \cap M_P[k,i] \circ M_P[i,i] \circ M_P[i,j]$; si temp=[0] alors entrée_nulle=true

si temp$\neq M_P[k,j]$ **alors**

$M_P[k,j]$=temp; $M_P[j,k]=(\text{temp})^t$

si $k \leq j$ **alors** $Q=Q \cup \{(X_k,X_j)\}$ **sinon** $Q=Q \cup \{(X_j,X_k)\}$ **finsi finsi fait fait fin**

Annexe 2-1

Résolution d’un CSP binaire discret : consistance de chemin

fonction Look_Ahead(M) : booléen
Début
PC2(M)
si inconsistance alors retourner FAUX finsi
si tous les domaines sont singletons alors retourner VRAI
Sinon
    choisir une variable disjonctive X_i de X
    pour toute valeur V_i de $D(X_i)$ faire
        M’=M
        M’[i,i][a,b]= 1 si a=b=V_i, O sinon
        si Look_Ahead(M’) alors retourner VRAI finsi
    fait
    retourner FAUX finsi
fin

Look_Ahead avec l’algorithme de consistance de chemin PC2 comme procédure de filtrage durant la recherche

Annexe 2-1

Résolution d'un CSP binaire discret : consistance de chemin

Exemple :

Appliquer les algorithmes de consistance de chemin PC1 et PC2 (version 2) au CSP P=(X,D,C) suivant :

- $X=\{X_1,X_2,X_3\}$
- $D(X_1)=D(X_2)=D(X_3)=\{1,2,3,4,5\}$
- $C=\{c_1 : X_1<X_2 ;$
  $c_2 : X_1<X_3 ;$
  $c_3 : X_2<X_3\}$

➔ Voir Annexe 2-2