Programmation par Contraintes

Module du Master “Systèmes Informatiques Intelligents” 2ème année

Chapitre IV

CSP binaires continus

Mr ISLI

Faculté d’Informatique

Université des Sciences et de la Technologie Houari Boumediène

BP 32, El-Alia, Bab Ezzouar

DZ-16111 ALGER

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

aisli@usthb.dz

CHAPITRE IV
CSP binaires continus

CSP binaires continus

- CSP quantitatifs
  - TCSP : Temporal Constraint Satisfaction Problems
- CSP qualitatifs
  - « make only as many distinctions as necessary »
    - L'algèbre des points (voir Annexe 3)
    - L'algèbre des intervalles (voir Annexe 4)
    - L'algèbre des directions cardinales

CHAPITRE IV
CSP binaires continus (TCSP)

CSP binaires continus

- TCSP : Temporal Constraint Satisfaction Problems

  Un TCSP est une paire P=(X,C) :
  - X ensemble fini de variables : $X=\{X_1, \ldots, X_n\}$
  - C ensemble fini de contraintes sur les variables de P
  - Le domaine de chacune des variables est l'ensemble IR des réels ou l'ensemble Q des rationnels
    - On considère dans toute la suite que le domaine commun des variables est IR

CHAPITRE IV
CSP binaires continus (TCSP)

CSP binaires continus

- TCSP P=(X,C) : Contraintes
  - Contraintes binaires
    - $(X_j - X_i) \in C_{ij}$, avec $C_{ij} \subseteq IR$
  - Contraintes unaires
    - $X_i \in C_i$, $C_i \subseteq IR$

CHAPITRE IV
CSP binaires continus (TCSP)

CSP binaires continus

- TCSP P=(X,C) : ajout d'une variable X_0 origine du monde
  - $X=\{X_0,X_1, \ldots,X_n\}$
  - Contraintes unaires transformées en contraintes binaires
    - $X_i \in C_i$ devient $(X_i-X_0)\in C_{0i}$
    - $C_{0i} = C_i$
- Soit P' le TCSP obtenu (le domaine de X_0 est le même que celui des autres variables)

CSP binaires continus

- **Résultat (théorème 1) :**
  - P est consistant ssi P’ est consistant
  - Toute solution de P peut être transformée en une solution de P’
    - X_0 instanciée à 0, les autres variables gardent les instanciations qu’elles ont dans la solution de P
  - Toute solution de P’ peut être transformée en une solution de P via une translation globale de la solution de P’ de $-d_0$, d_0 étant l’instanciation de la variable origine du monde X_0 dans la solution de P’
    - les contraintes, qui sont relatives, restent satisfaites car les distances séparant les variables ne sont pas modifiées par une translation globale

CSP binaires continus

- Résultat (théorème 2) :
  - P' est consistant de nœud
  - P' est consistant d'arc

CSP binaires continus

- On suppose qu'un TCSP est donné sous la forme P'
- Si ce n'est pas le cas, on le ramène à la forme P' en ajoutant une variable origine du monde X_0, et en transformant les contraintes unaires $X_i \in C_i$ en contraintes binaires $(X_i - X_0) \in C_{0i}$ (avec $C_{0i} = C_i$)

CSP binaires continus

- TCSP : application aux problèmes de job shop

Instance ft06 (Fisher et Thompson)

6 jobs, 6 machines

(2,1)	(0,3)	(1,6)	(3,7)	(5,3)	(4,6)
(1,8)	(2,5)	(4,10)	(5,10)	(0,10)	(3,4)
(2,5)	(3,4)	(5,8)	(0,9)	(1,1)	(4,7)
(1,5)	(0,5)	(2,5)	(3,3)	(4,8)	(5,9)
(2,9)	(1,3)	(4,5)	(5,4)	(0,3)	(3,1)
(1,3)	(3,3)	(5,9)	(0,10)	(4,4)	(2,1)

CHAPITRE IV
CSP binaires continus (TCSP)

Représentation graphique d'un TCSP

- La représentation graphique d'un TCSP P=(X,C) est un graphe orienté pondéré $G_P=(X,E,l)$ défini comme suit :
  - L'ensemble des sommets de G_P est l'ensemble X des variables de P
  - Pour toutes variables X_i et X_j, si P admet une contrainte portant sur X_i et X_j, alors G_P contient un et un seul des deux arcs (X_i,X_j) et (X_j,X_i)
  - Pour toutes variables X_i et X_j, si P n'admet aucune contrainte portant sur X_i et X_j, alors G_P ne contient ni l'arc (X_i,X_j) ni l'arc (X_j,X_i)
  - Pour tout arc (X_i,X_j) de G_P, l'étiquette $l(X_i,X_j)$ de (X_i,X_j) est tirée de la contrainte $(X_j-X_i)\in C_{ij}$ de P portant sur X_i et X_j : $l(X_i,X_j)= C_{ij}$

Représentation matricielle d'un TCSP

- La représentation matricielle d'un TCSP $P=(X,C)$ est une matrice carrée notée M_P, à $|X|$ lignes et $|X|$ colonnes définie comme suit :
  - Pour toute variable X_i, $M_P[i,i]=\{0\}$
  - Pour toutes variables distinctes X_i et X_j, telles que (X_i,X_j) est arc de G_P :
    - $M_p[i,j]$ est l'étiquette de l'arc (X_i,X_j)
    - $M_p[j,i]$ est la transposée de $M_p[i,j]$
  - Pour toutes les autres paires (X_i,X_j) de variables :
    - $M_p[i,j]=M_p[j,i]=Q$

CHAPITRE IV
CSP binaires continus (TCSP)

CSP binaires continus

- TCSP P=(X,C)
  - Transposée d'une contrainte
  - Intersection de deux contraintes
  - Composition de deux contraintes

CSP binaires continus

- TCSP $P=(X,C)$
  - Adaptation de la consistance d'arc : les entrées $M_P[0,i]=C_{0i}$, $i=1$ à n, sont vues comme les domaines binarisés des variables X_1, …, X_n, respectivement.
  - Consistance d'arc des domaines binarisés (binarized-domains arc-consistency, ou bdArc-Consistency) : le domaine binarisé D(Xi)=Coi de la variable Xi est consistant d'arc si
    - Pour toute paire (X_i,X_j) de variables, avec $i\neq j$ et $0\notin\{i,j\}$:

$$C_{oi}\subseteq C_{0j}\circ C_{ji}$$

  - Un TCSP est bdArc-Consistant si tous ses domaines binarisés sont consistants d'arc.

CSP binaires continus

- TCSP P=(X,C)
  - Rendre un TCSP bdArc-Consistant : algorithme bdAC-3 (adaptation aux TCSP de l'algorithme AC-3 de consistance d'arc des CSP discrets)

CHAPITRE IV
CSP binaires continus (TCSP)

fonction bdAC-3(M)

début

$Q \leftarrow \{(X_i,X_j) : (i \neq j)$ et $(0 \notin \{i,j\})$ et (il existe une contrainte entre X_i et X_j)$\}$

tant que $Q \neq \emptyset$ **faire**

Prendre une paire (X_i,X_j) de variables de Q

supprimer la paire de Q : $Q \leftarrow Q \backslash \{(X_i,X_j)\}$

$temp=M[0,i] \cap M[0,j] \circ M[j,i]$

si $(temp \subset M[0,i])$ **alors**

si $temp=\emptyset$ alors retourner faux finsi /* inconsistance détectée */

$M[0,i]=temp$

$Q \leftarrow Q \cup \{(X_k,X_i) : (k \notin \{0,i,j\})$) et (il existe une contrainte entre X_k et X_i)$\}$

finsi

fait

retourner vrai /* TCSP rendu bdArc-Consistant */

fin

CHAPITRE IV
CSP binaires continus (TCSP)

CSP binaires continus

Déroulement de bdAC3 sur le TCSP P=(X,C) :

- X={X0,X1,X2,X3,X4}
- C={c1:(X1-X0)$\in$[10,20],
  c2:(X4-X0)$\in$[60,70],
  c3:(X2-X1)$\in$[30,40],
  c4:(X3-X2)$\in$[-20,-10],
  c5:(X4-X3)$\in$[40,50]}

Domaines binarisés rendus consistants d'arc :

[10,20], [40,50], [20,30], [60,70]

CHAPITRE IV
CSP binaires continus (TCSP)

CSP binaires continus

- TCSP P=(X,C)
  - STP : Simple Temporal Problem
    - Toutes les contraintes sont convexes
  - Importance :
    - Les STP couvrent un large éventail de problèmes réels
    - Ils ont un comportement calculatoire polynomial :
      - La bd-consistance d'arc décide la consistance d'un STP
      - De plus : le résultat de la bd-consistance d'arc est un STP dont les domaines binarisés sont minimaux
    - Le fragment des 2-TCSP : NP-difficile
    - Les algorithmes de recherche que nous verrons sont basés sur la recherche d'un raffinement STP bd-consistant d'arc

CHAPITRE IV
CSP binaires continus (TCSP)

CSP binaires continus

- TCSP P=(X,C)
  - STP : Simple Temporal Problem
  - Appliquer la bd-consistance d'arc à P est équivalent à appliquer l'algorithme des plus courts chemins one-to-all à son graphe des distances

CHAPITRE IV
CSP binaires continus (TCSP)

CSP binaires continus

- TCSP P=(X,C) : résolution
  - Algorithme 1 (naïf) : Générer et Tester (GET)
  - Parcours exhaustif de tous les raffinements convexes maximaux
    - Un raffinement convexe maximal étant un STP : bd-consistance d'arc pour vérifier sa consistance

CHAPITRE IV
CSP binaires continus (TCSP)

GET : version itérative

1. Générer tous les raffinements convexes maximaux
2. Initialement, aucun raffinement convexe maximal n'est marqué
3. Considérer un raffinement convexe maximal M non marqué
4. Marquer M
5. bdAC3(M)
6. Si M ne contient pas de domaines binarisés vides
   - Succès : retourner M
7. Si tous les raffinements convexes maximaux sont marqués
   - Echec : retourner « le TCSP n'admet pas de solution »
8. Passer au raffinement convexe maximal suivant : aller à 3.

CHAPITRE IV

CSP binaires continus (TCSP)

GET : version récursive

```
fonction GET(M) : booléen
début
si (M STP) alors
     bdAC3(M)
     si (aucun domaine binarisé vide dans M) alors retourner VRAI
                                              sinon retourner FAUX finsi
sinon
      choisir une entrée disjonctive (i,j) de M
      pour tout sous-ensemble convexe maximal V de M[i,j] faire
            M2=M; M2[i,j]=V; M2[j,i}=V^t
            si GET(M2) alors retourner VRAI finsi fait
      retourner FAUX
finsi
fin
```

CHAPITRE IV
CSP binaires continus (TCSP)

CSP binaires continus

- TCSP P=(X,C) : résolution
  - Algorithme 2 (intelligent) : LookAhead(M)
    - Répéter : instanciation suivie d'un filtrage
    - On n'instancie pas les variables (domaines infinis)
    - On instancie les arcs disjonctifs : les arcs (X_i,X_j) tels que $i\neq 0$, $j\neq 0$ et M[i,j] n'est pas convexe
    - Filtrage avec l'algorithme de bd-consistance d'arc bdAC3

CHAPITRE IV
CSP binaires continus (TCSP)

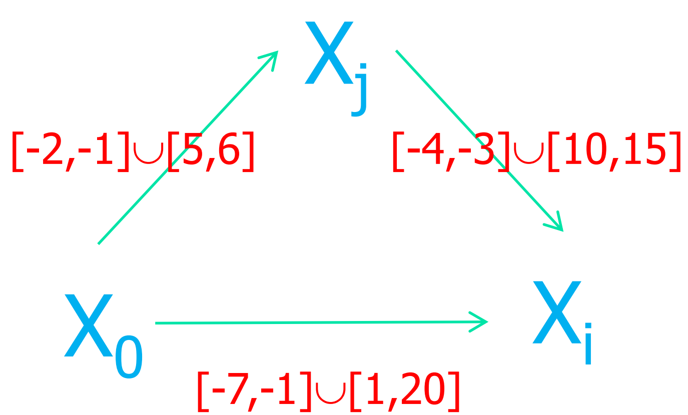
CSP binaires continus

- TCSP P=(X,C) : résolution
  - Algorithme 2 (intelligent) : LookAhead(M)
    - Filtrage avec un algorithme de bd-consistance d’arc
    - Risque : le problème de fragmentation

CHAPITRE IV
CSP binaires continus (TCSP)

Exemple 1 : TCSP

- Le problème de fragmentation : l'opération clé de la bd-consistance d'arc, $M_P[0,i]=M_P[0,i] \cap M_P[0,j] \circ M_P[j,i]$, peut fragmenter le domaine binarisé de la variable X_i
- Remède : bd-consistance d'arc faible → wbdAC3



$[-6,-4] \cup [1,3] \cup [8,14] \cup [15,20]$

CHAPITRE IV
CSP binaires continus (TCSP)

CSP binaires continus

- TCSP P=(X,C)
  - bd-consistance d'arc faible :
    - Remplacer la composition par la composition faible
    - La composition faible est la composition des fermetures convexes
    - Weak bdAC3 (wbdAC3)

CSP binaires continus

- TCSP P=(X,C) : résolution
  - Algorithme 2 (intelligent) : LookAhead(M)
    - Recherche d'un raffinement convexe bd-consistant d'arc :
      - STP bd-consistant d'arc
    - Ordre d'instanciation sur les paires de variables
    - Une paire de variables est instanciée avec les blocs convexes maximaux constituant son poids
    - A chaque fois qu'une nouvelle paire est instanciée
      - Filtrage : bd-consistance d'arc faible (wbdAC3)
      - Si inconsistance détectée : échec (retour arrière)
    - Si instanciation avec succès de toutes les paires de variables
      - raffinement convexe bd-consistant d'arc

        → STP consistant avec domaines binarisés minimaux

```
fonction LookAhead(M) : booléen
début
wbdAC3(M)
si (M contient un domaine binarisé vide) alors retourner FAUX finsi
si toutes les paires de variables sont instanciées alors retourner VRAI
Sinon
      choisir une paire (Xi,Xj) disjonctive de variables qui n'est pas encore instanciée
      pour tout sous-ensemble convexe maximal r de M[i,j] faire
            M'=M
            M'[i,j]=r
            M'[j,i]=r^t
            si LookAhead(M') alors retourner VRAI finsi
      fait
      retourner FAUX      //l'instanciation partielle ne peut pas être étendue à la paire (Xi,Xj)
finsi
fin
```

Algèbre des directions cardinales

- Objets et relations
  - Objets : les points du plan (espace 2-dimensionnel)
  - Relations qualitatives sur des paires de points :
    - Relations atomiques : 9
    - Relations générales (disjonctives) :
      - Sous-ensembles de relations atomiques : $2^9=512$

Algèbre des directions cardinales : les 9 relations atomiques

Symbole	**Signification**	**Traduction**
SW	South-West	sud-ouest
W	West	ouest
NW	North-West	nord-ouest
S	South	sud
EQ	EQual	Egal
N	North	nord
SE	South-East	sud-est
E	East	est
NE	North-East	nord-est

Algèbre des directions cardinales

- CSP qualitatif de directions cardinales

  Paire P=(X,C) :

  - X ensemble fini de variables : $X=\{X_1, \ldots, X_n\}$
  - C ensemble fini de contraintes binaires sur des paires de variables de P
  - Le domaine de chacune des variables est l'ensemble IR^2 ou l'ensemble Q^2
    - Le domaine commun des variables sera noté D(P)

Algèbre des directions cardinales

CSP qualitatif de directions cardinales P=(X,C) : Contraintes

- $R(X_i,X_j)$, R étant une des 512 relations de l'algèbre des directions cardinales

Algèbre des directions cardinales

- CSP qualitatif de directions cardinales P=(X,C)
  - Représentation graphique
  - Représentation matricielle

Algèbre des directions cardinales : Transposée

Relation atomique r	Transposée r^t de r
SW	NE
W	E
NW	SE
S	N
EQ	EQ
N	S
SE	NW
E	W
NE	SW

Algèbre des directions cardinales

- Intersection
  - $R_1 \cap R_2 = \{r : r \in R_1 \text{ et } r \in R_2\}$
  - Intersection ensembliste

Algèbre des directions cardinales

- Table de composition

°	**SW**	**…**	**N**	**…**	**NE**	**EQ**
SW	SW	…	{SW,W,NW}	…	?	SW
…	…	…	…	…	…	…
S	SW	…	{S,EQ,N}	…	{SE,E,NE}	S
…	…	…	…	…	…	…
NE	?	…	NE	…	NE	NE
EQ	SW	…	N	…	NE	EQ

CHAPITRE IV
CSP binaires continus (directions cardinales)

Algèbre des directions cardinales

- CSP qualitatif de directions cardinales P=(X,C)
  - Nœud-consistant
  - Arc-consistant

Algèbre des directions cardinales

- CSP qualitatif de directions cardinales P=(X,C)
  - Consistance de chemin : P est consistant de chemin si
    - Pour tout triplet (X_i,X_k,X_j) de variables :

$$C_{ij} \subseteq C_{ik} \circ C_{kj}$$

```
procedure PC2(M_P)
    début
    Q ← {(X_i,X_k,X_j)∈X³ : (i<j) et (k∉{i,j})};
    entrée_vide=faux
    tant que Q≠∅ et ¬entrée_vide faire
      Prendre un triplet (Xi,X_k,Xj) de variables de Q, et l'en supprimer (Q ← Q\{(X_i,X_k,X_j)});
      temp=M_P[i,j]∩M_P[i,k]°M_P[k,j];
      si temp=∅ alors entrée_vide=vrai
      sinon
           si temp≠M_P[i,j] alors
                     M_P[i,j]=temp; M_P[j,i]=(temp)^t
                     Q=Q∪{(X_i,X_j,X_m): (i<m) et (m≠j)}∪{(X_m,X_i,X_j): (m<j) et (m≠i)}
           finsi
      finsi
    fait
    fin
```

Algèbre des directions cardinales

- CSP qualitatif de directions cardinales P=(X,C)
  - CSP atomique :
    - chaque entrée de la représentation matricielle est une relation atomique
  - Importance :
    - Un CSP atomique a un comportement calculatoire polynomial :
      - La consistance de chemin décide la consistance d'un tel CSP
      - De plus : le résultat de la consistance de chemin est un CSP minimal et même fortement n-consistant
    - Les algorithmes de recherche sont basés sur la recherche d'un raffinement atomique consistant de chemin

CHAPITRE IV
Résolution d'un CSP binaire continu

GET : version itérative

1. Générer tous les raffinements atomiques
2. Initialement, aucun raffinement atomique n'est marqué
3. Considérer un raffinement atomique M non marqué
4. Marquer M
5. PC2(M)
6. Si M ne contient pas d'entrées vides
   - Succès : retourner M
7. Si tous les raffinements atomiques sont marqués
   - Echec : retourner « le CSP n'admet pas de solution »
8. Passer au raffinement atomique suivant : aller à 3.

CHAPITRE IV
Résolution d'un CSP binaire continu

GET : version récursive

```
fonction GET(M) : booléen
début
si (M atomique) alors
    PC2(M)
    si (aucune entrée vide dans M) alors retourner VRAI sinon retourner FAUX finsi
sinon
    choisir une entrée disjonctive (i,j)
    pour toute relation atomique r de M[i,j] faire
        M2=M; M2[i,j]={r}; M2[j,i}={r}^t
        si GET(M2) alors retourner VRAI finsi fait
    retourner FAUX
finsi
fin
```

Algèbre des directions cardinales

- CSP qualitatif de directions cardinales P=(X,C)
  - Algorithme 2 (intelligent) : LookAhead(M)
    - Répéter : instanciation suivie d’un filtrage
    - On n’instancie pas les variables (domaines infinis)
    - On instancie les arcs disjonctifs : les arcs (X_i,X_j) tels que M[i,j] a plus d’une relation atomique
    - Filtrage avec un algorithme de consistance de chemin tel que PC1 ou PC2

```
fonction LookAhead(M) : booléen
début
PC2(M)
si (M contient une entrée vide) alors retourner FAUX finsi
si toutes les paires de variables sont instanciées alors retourner VRAI
Sinon
    choisir une paire (Xi,Xj) disjonctive de variables qui n’est pas encore instanciée
    pour toute relation atomique r de M[i,j] faire
            M’=M
            M’[i,j]={r}
            M’[j,i]={r}^t
            si LookAhead(M’) alors retourner VRAI finsi
    fait
    retourner FAUX    //l’instanciation partielle ne peut pas être étendue à la paire (Xi,Xj)
finsi
fin
```

Exemple 1 :

- CSP qualitatif de directions cardinales
  - Béjaia est à l’est d’Alger
  - Oran est à l’ouest d’Alger
  - Le bâteau est au nord-est d’Alger
    - satellite 1 à l’instant t
  - Le bâteau est au nord ou au nord-est d’Oran
    - satellite 2 au même instant t

CHAPITRE IV
Résolution d'un CSP binaire continu

Exemple 2 :

- CSP qualitatif de directions cardinales
  - Béjaia est à l'est d'Alger
  - Oran est à l'ouest d'Alger
  - Le bâteau est au au nord ou au nord-est d'Alger
    - satellite 1 à l'instant t
  - Le bâteau est au nord-est d'Oran
    - satellite 2 au même instant t

CHAPITRE IV
Résolution d'un CSP binaire continu

Exemple 3 (exo 2 de la série TD n° 5) :

- CSP qualitatif de directions cardinales
  - Béjaia est à l'est d'Alger
  - Oran est à l'ouest d'Alger
  - Le bâteau est au nord-ouest d'Alger
    - satellite 1 à l'instant t
  - Le bâteau est au nord-est de Béjaia
    - satellite 2 au même instant t