Réseaux de contraintes valuées

Thomas Schiex
INRA

CSP et optimisation

La modélisation d'un problème réel fait naturellement apparaître :

- des contraintes **dures**
- des contraintes **préférablement satisfaites**
- des contraintes **incertaines**

La modélisation dans un cadre CSP peut conduire à l'incohérence (**sur-contraint**) ou à une abondance de solutions médiocres (**sous-contraint**).

La possibilité d'exprimer des contraintes molles (soft) permet de:

- traiter les problèmes **sur/sous-contraints**
- définir un cadre homogène pour l'**optimisation dans les CSP**

Existant

- CSP **additifs** : minimiser la somme des valuations des contraintes insatisfaites (Shapiro and Haralick 1981).
- CSP **possibilistes** : minimiser la valuation maximale des contraintes insatisfaites (Rosenfeld *et al.* 1976).
- CSP **lexicographiques** : minimiser le nombre de contraintes insatisfaites de valuation maximale, puis, à nombre égal, le nombre de contraintes insatisfaites de valuation inférieure, et ainsi de suite (Fargier *et al.* 1993).
- CSP **probabilistes** : minimiser la probabilité qu'une instanciation ne soit pas solution du problème réel (Fargier and Lang 1993).

Situation troublante

Dans chaque extension, les algorithmes, théorèmes, propriétés et classes polynomiales usuelles des CSP sont (éventuellement) étendus.

Grande **redondance** des travaux.

- **CSP possibilistes** : (presque) tous les algorithmes et propriétés usuelles s'étendent (Rosenfeld *et al.* 1976; Snow and Freuder 1990; Schiex 1992; Fargier 1994)

- **autre cas** : la majorité des algorithmes et propriétés usuelles résistent à l'extension. Les problèmes sont plus difficiles à résoudre.

Quelle est la source de cette difficulté ? Peut-on la contourner ?

Motivations essentielles

- **Généralité** : fournir un cadre générique qui couvre les propositions existantes.
  - éviter la redondance.
  - analyser les différences de comportement
- **Spécificité** : accumuler suffisamment de propriétés pour construire des théorèmes et des algorithmes non triviaux.

Accessoirement : fournir un cadre homogène pour l'optimisation dans les réseaux de contraintes.

Avantages escomptés

- avoir **un seul cadre** de représentation : variables, domaines de valeur, contraintes ;
- prendre en compte dans le même cadre des **contraintes dures** (à satisfaire de façon impérative) et des **contraintes molles** (à satisfaire au mieux) ;
- représenter de la même façon des **problèmes cohérents**, avec préférences sur les solutions admissibles, et des **problèmes incohérents**, avec préférences sur les relaxations cohérentes.

Cadre VCSP (T. Schiex, H. Fargier, G. Verfaillie, IJCAI95)

- À chaque **contrainte** est associé une **valuation** reflétant son importance : préférence, poids, gain, priorité, probabilité d'existence . . .
- La valuation d'une affectation est le résultat de la **combinaison** des valuations des **contraintes insatisfaites** par un opérateur satisfaisant un ensemble d'axiomes.
- Les instanciations peuvent être **comparées** entre elles en exploitant un **ordre total** sur les valuations.
- Le problème est de produire une affectation de **valuation minimum**.

Monoïde commutatif totalement ordonné avec opérateur monotone.

Approches similaires

- **Aho/Ullman** (années 70), (Minoux 1976) : **demi-anneau** utilisé pour généraliser les algorithmes de plus court chemin (all-pair).
- **(Shafer and Shenoy 1988)** : une structure de type **demi-groupe** est utilisée pour généraliser les algorithmes de type programmation dynamique (CSP: tree-clustering. . .)
- **(Bistarelli et al. 1995)** : une structure à base de **demi-anneau** est utilisée pour étendre les CSP. Extension des alg. de type programmation dynamique et cohérence locale.
- Raisonnement dans l'incertain : normes et **co-normes triangulaires**, basées sur des demi-groupes.
- **Partial CSP** (Freuder 1989), s'appuient sur une **distance** entre problèmes.

Structures proches, compromis différents, buts différents.

Plus précisément

- $\mathrm{VCSP} = (V, D, C, S, \varphi)$, $S = (E, \succeq, \top, \bot, \oplus)$
- $E =$ **ensemble de valuations**, numérique ou symbolique, permettant de valuer les contraintes et les affectations ;
- $\top =$ **élément maximum** de E, permettant de valuer les contraintes dures et les affectations totalement incohérentes ;
- $\bot =$ **élément minimum** de E, permettant de valuer les affectations totalement cohérentes ;
- $\succeq =$ **ordre total** sur E, permettant de comparer deux valuations ;
- $\oplus =$ **opérateur**, permettant de **combiner** deux valuations ;
- $\varphi : C \to E =$ **fonction de valuation**, permettant d'annoter chaque contrainte.

Valuation d'une affectation

$$\varphi(A) = \bigoplus_{c \in C} \varphi(c, A)$$

$\varphi(c, A) = \perp$, si c est **satisfaite** par A.

$\varphi(c, A) = \varphi(c)$, sinon.

Propriétés minimales

- $\oplus$ **commutatif** et **associatif** ;
  la valuation d'une affectation ne dépend pas de la façon de réaliser l'agrégation ;
- $\perp$ **élément neutre** de $\oplus$: $\forall a \in E, a \oplus \perp = a$
  les contraintes satisfaites ne modifient pas le résultat de l'agrégation ;
- $\top$ **élément absorbant** de $\oplus$: $\forall a \in E, a \oplus \top = \top$
  les contraintes insatisfaites et impératives induisent une insatisfaction totale ;
- $\oplus$ **monotone** relativement à $\succ$:
  $\forall a, b, c \in E, a \succeq b \Rightarrow (a \oplus c) \succeq (b \oplus c)$
  si A est localement meilleur que A', toutes choses égales par ailleurs, A est globalement meilleur que A'.

Différents sous-cadres

CSP	E	$\succ$	$\top$	$\bot$	$\oplus$
classiques	{t,f}	f $\succ$ t	f	t	$\wedge$
additifs	$\overline{\mathbb{N}}$	$>$	$+\infty$	0	$+$
possibilistes	$[0,1]$	$>$	1	0	max
lexicographiques	$[0,1]^*$	$>^*$	$\top$	$\emptyset$	$\cup$
probabilistes	$[0,1]$	$>$	1	0	$a \oplus b =$ $1-(1-a)(1-b)$

Des variantes du cadre VCSP

- valuation des **tuples** :
  $\varphi(A) = \oplus_{c \in C} \ \varphi(c, A)$
  $\varphi(c, A) = \varphi(A \downarrow_{V(c)})$
  plus fin que le cadre VCSP de base ;

- valuation des **valeurs** :
  $\varphi(A) = \oplus_{v \in V} \ \varphi(v, A)$
  $\varphi(v, A) = \varphi(A \downarrow_{v})$
  cas particulier du précédent ;

- valuation des **variables** :
  $\varphi(A) = \oplus_{v \in V} \ \varphi(v, A)$
  $\varphi(v, A) = \varphi(v)$ si v n'est pas affectée dans A, $\perp$ sinon.
  cas particulier du précédent.

Un cadre proche

SCSP : **Semiring-based CSP**

(S. Bistarelli, F. Rossi, U. Montanari, IJCAI95)

Principale différence : **ordre partiel** possible entre valuations (une hypothèse d'ordre total suffit à confondre les deux cadres).

Présentation des deux cadres et comparaison :

(S. Bistarelli, F. Rossi, U. Montanari, T. Schiex, G. Verfaillie, H. Fargier, Constraints 99)

Notion de relaxation

La **relaxation** d'un VCSP (V, D, C, S, φ) est un **CSP classique** (V, D, C') tel que $C' \subseteq C$.

Sa valuation est définie comme la **combinaison** des valuations des **contraintes relaxées** :

$$\varphi(C, C') = \bigoplus_{c \in C - C'} \varphi(c)$$

Lien entre affectations et relaxations cohérentes

Affectation $A \rightarrow$ Relaxation des contraintes insatisfaites par A : relaxation cohérente de même valuation ;

Relaxation cohérente $C' \rightarrow$ Solutions de C' : affectations de valuation inférieure ou égale ;

$\Rightarrow$ Produire une affectation de valuation minimale ou produire une relaxation cohérente de valuation minimale sont deux **problèmes équivalents**.

On préfère traiter le premier : espace de recherche généralement moins grand, test de complexité **polynomiale** contre test de complexité **exponentielle**.

Deux propriétés supplémentaires importantes

- **idempotence** : $\forall a \in E, a \oplus a = a$;
- **monotonie stricte** : $\forall a, b, c \in E, a \succ b \wedge c \neq \top \Rightarrow (a \oplus c) \succ (b \oplus c)$;

Propriétés contradictoires :

Propriétés de base + $|E| > 2$

$\Rightarrow \neg$ (idempotence $\wedge$ monotonie stricte).

Le cadre des **CSP classiques** est le seul qui bénéficie des deux propriétés.

Idempotence

Un seul choix : l'opérateur **max** $\rightarrow$ CSP **possibilistes**

Propriétés de base + idempotence $\Rightarrow \oplus = \max$

Du point de vue de la **modélisation**, **effet de noyade** non désirable :

- priorités des contraintes insatisfaites par A_1 : $\{0.6\}$;
- priorités des contraintes insatisfaites par A_2 : $\{0.5, 0.4, 0.4, 0.4, 0.4\}$;
- A_2 préférée à A_1.

Une relaxation cohérente optimale au sens du critère de valuation n'est pas forcément maximale au sens de l'inclusion.

Du point de vue de la **résolution**, à la base des mécanismes de **propagation de contraintes**.

Monotonie stricte

$\rightarrow$ CSP **additifs**, **probabilistes**, **lexicographiques** . . .

Du point de vue de la **modélisation**, pas d'**effet de noyade**.

Toute relaxation cohérente optimale au sens du critère de valuation est maximale au sens de l'inclusion.

Du point de vue de la **résolution**, difficulté à étendre les mécanismes de **propagation de contraintes**.

Cohérence locale - Arc cohérence

Propriété **cohérence locale** : propriété vérifiable en **temps polynomial**, **relaxation de la cohérence** globale.

Filtrage associé: algorithme qui produit en **temps polynomial** un CSP **équivalent** vérifiant la propriété de cohérence locale.

Arc cohérence (cas binaire):

- chaque valeur a de chaque variable i peut s'étendre de façon cohérente sur toute autre variable.
- projection de chaque contrainte, jointe avec les domaines des variables qu'elle implique sur ces variables et intersection avec les domaines.

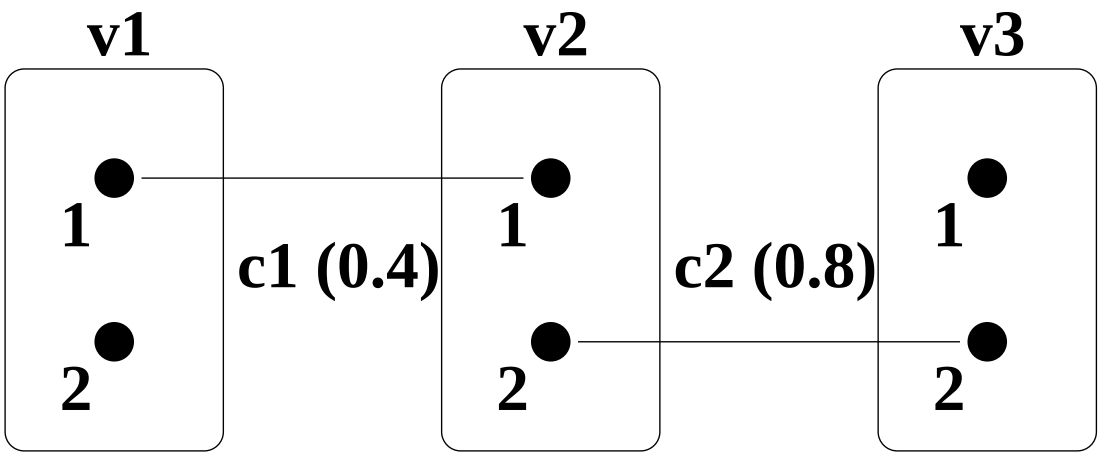
$$\varphi(v, val) \leftarrow \bigoplus_{v' adj.} \min_{val' \in d(v')} \oplus[\varphi(v', val'), \varphi((v, val), (v', val'))]$$

Opérateur idempotent

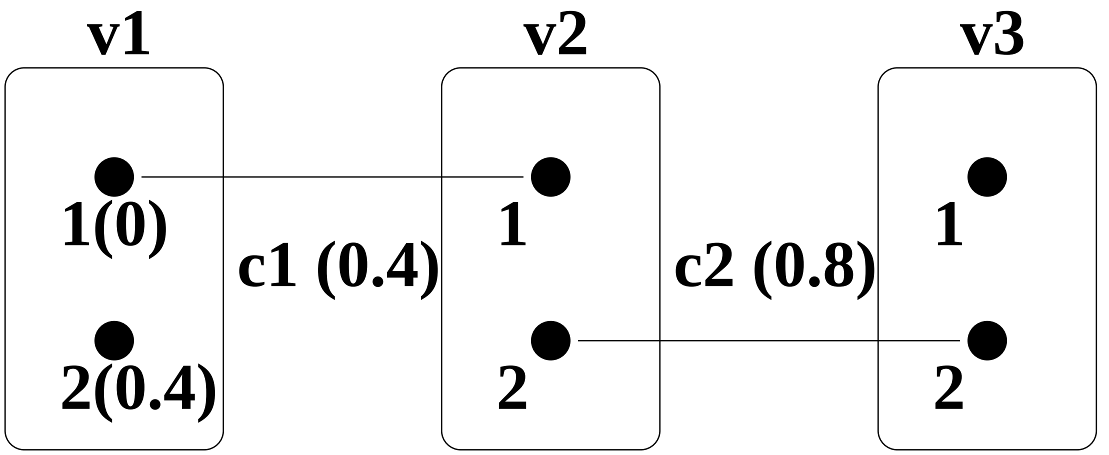
Première **bonne nouvelle** : Extension des définitions de **cohérence locale** et des algorithmes de **propagation** associés.

Opération de base:

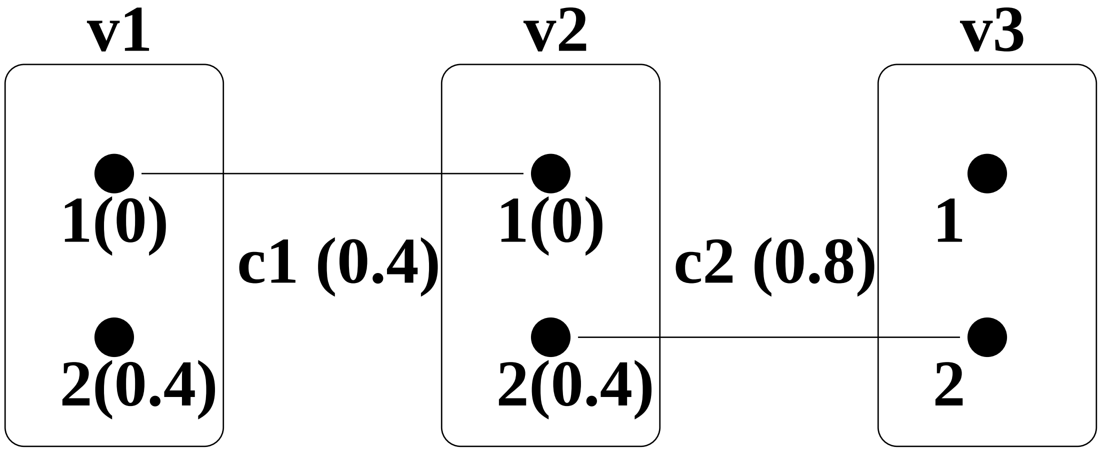
$\varphi(v, val) \leftarrow \max_{v' adj.} \min_{val' \in d(v')} \max[\varphi(v', val'), \varphi((v, val), (v', val'))]$



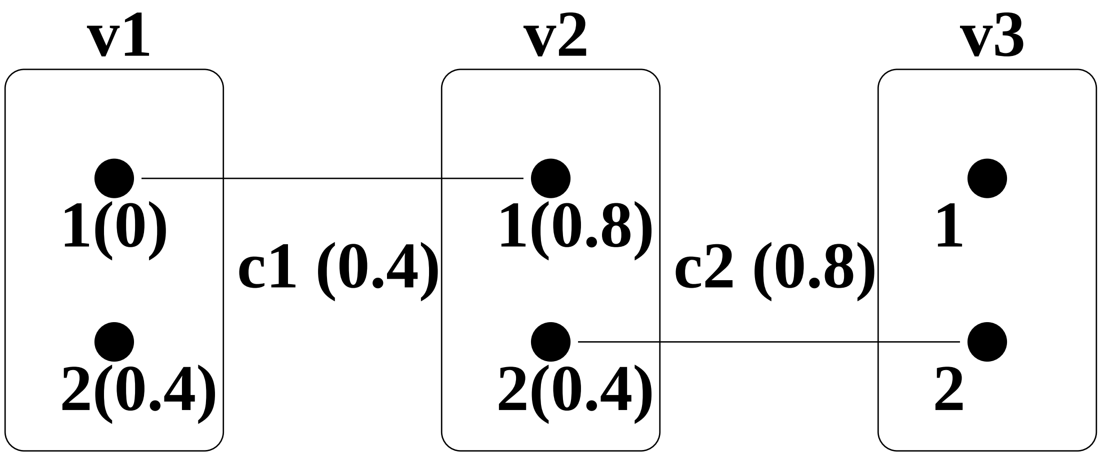
CSP possibilistes et propagation de contraintes



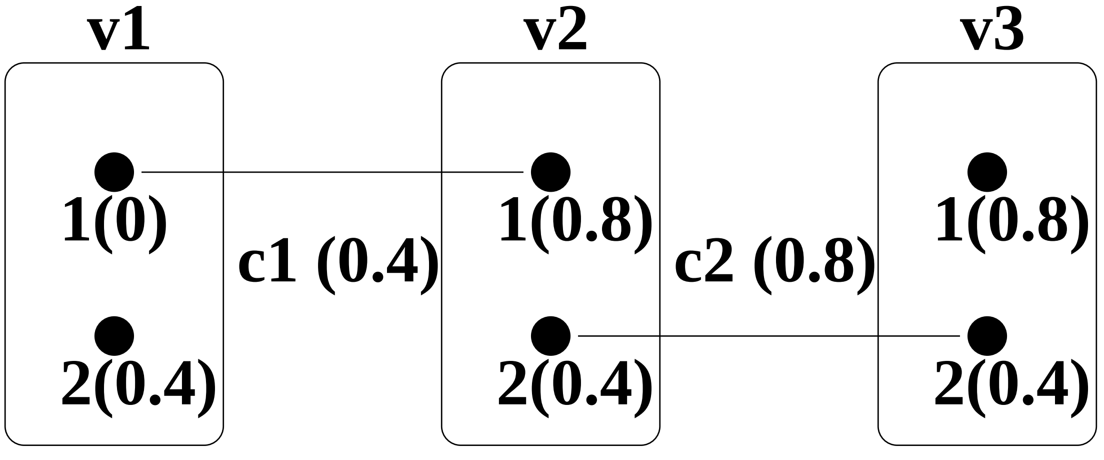
CSP possibilistes et propagation de contraintes



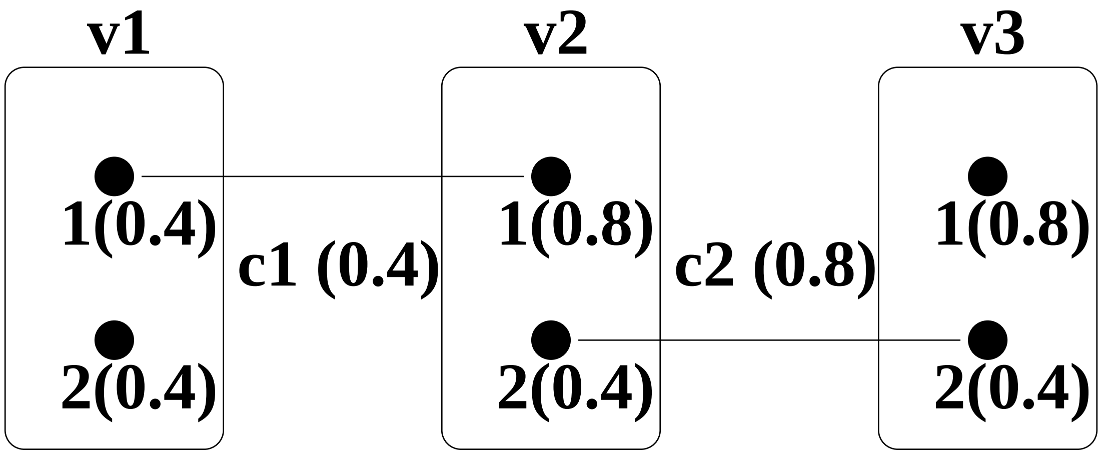
CSP possibilistes et propagation de contraintes



CSP possibilistes et propagation de contraintes



CSP possibilistes et propagation de contraintes



Un **minorant de l'optimum** : $\oplus_{v\in V}[\min_{val\in d(v)} \varphi(v,val)]$
$\max_{v\in V}[\min_{val\in d(v)} \varphi(v,val)] = 0.4$

Introduit dans (Rosenfeld *et al.* 1976). Amélioré dans (Snow and Freuder 1990). Mécanique AC3.

CSP possibilistes, flous

Bonne nouvelle : la résolution d'un CSP possibiliste ou flou se ramène à $O(\log(k))$ résolutions de CSP classiques, avec k égal au nombre de niveaux de valuation utilisés.

Soit la relaxation obtenue en éliminant les contraintes de valuation inférieure ou égale à α (appelée α-coupe) : sa valuation est égale à α.

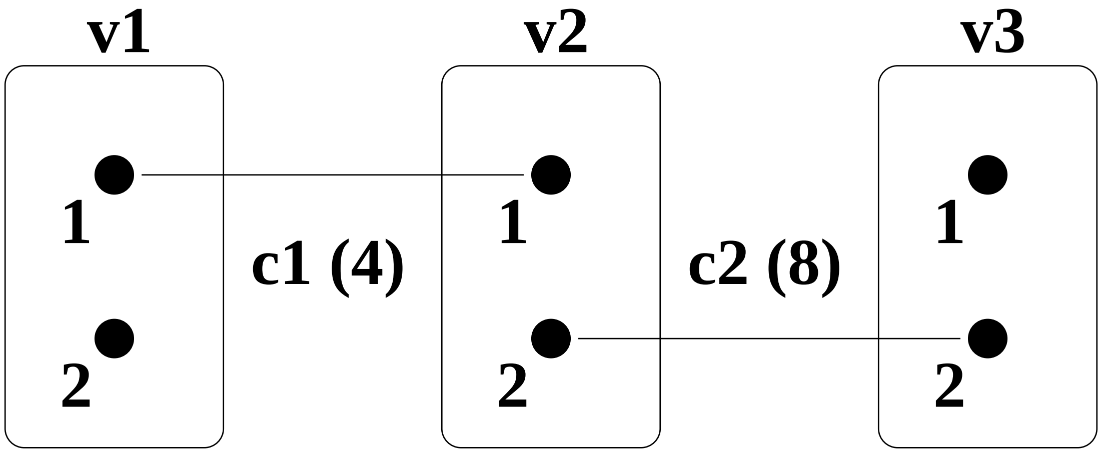
Une affectation optimale est solution de l'**α-coupe cohérente de valuation minimale**.

Opérateur strictement monotone

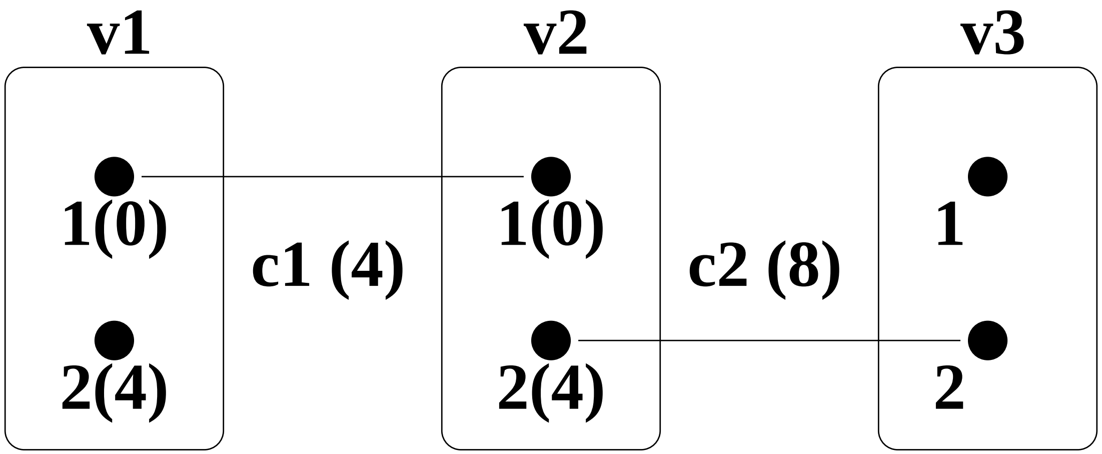
Première **mauvaise nouvelle** : Pas d'extension immédiate des définitions de **cohérence locale** et des algorithmes de **propagation** associés.

Opération de base envisagée:

$\varphi(v, val) \leftarrow \sum_{v' adj.} \min_{val' \in d(v')} [\varphi(v', val') + \varphi((v, val), (v', val'))]$

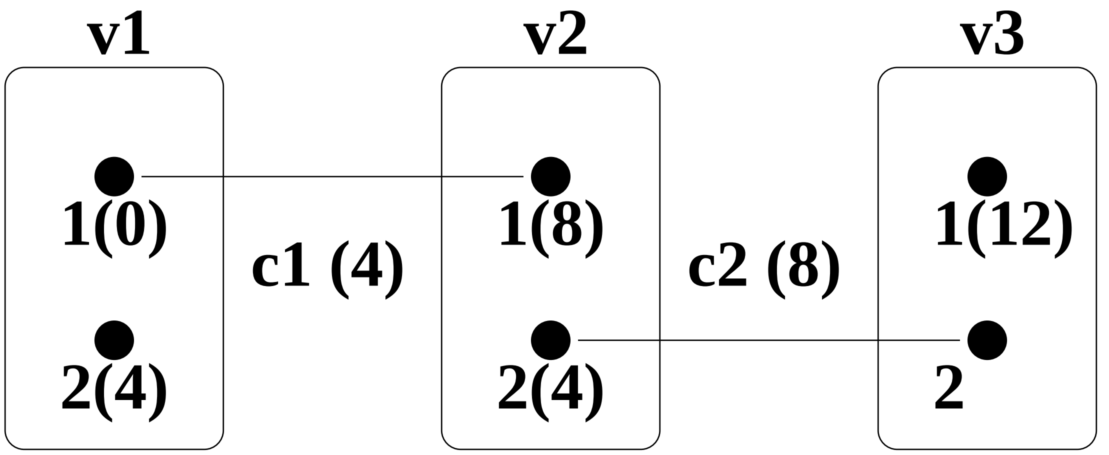


CSP additifs
et propagation de contraintes



On perd l'équivalence.

CSP additifs et propagation de contraintes



Rien ne va plus !

12 n'est pas un minorant de la valuation de toutes les affectations passant par $(v_3, 1)$.

On a pris deux fois en compte la contrainte c_2 et l'opérateur n'est pas idempotent.

CSP additifs
Méthodes de résolution

- Méthodes **gloutonnes** ou **heuristiques** : OK ;
- Méthodes de **recherche locale** (gradient, recuit simulé, recherche tabou, algorithmes génétiques . . .) : OK ;
- Méthodes d'**approximation** : ???

CSP additifs
Méthodes de résolution

- **Programmation dynamique** : une **bonne nouvelle** : si le graphe de contraintes est **acyclique** (forêt), un algorithme de type programmation dynamique fonctionne, en remontant des feuilles vers la racine de chaque arborescence
  $\Rightarrow$ **Sous-classe polynomiale** ;

Conséquence immédiate de (Shenoy 1991; Dechter *et al.* 1990).

CSP additifs
Méthodes de résolution

Extension des algorithmes de type backtrack : **Séparation et évaluation** (Branch and bound).

- recherche systématique en **profondeur d'abord** :
- **majorant** ub : la meilleure valuation connue jusqu'ici.
- **minorant** lb : de la meilleure valuation que l'on peut obtenir en étendant l'affectation courante.
- **backtrack** : quand $\mathrm{lb} \succeq \mathrm{ub}$.

Crucial: le **minorant**.

On pourrait le dériver des cohérences locales étendues. . .

CSP additifs
et minorants

L'idée : utiliser la notion de **relaxation**.

Tout CSP **cohérent** n'est pas **localement** incohérent.

$\Rightarrow$ La valuation d'une relaxation qui n'est pas localement incohérente optimale est un **minorant** de l'optimum.

- **Bonne nouvelle** : **backward** et **forward-checking** s'étendent facilement. Algorithmes peu efficaces.
- **Mauvaise nouvelle** : soit MAX-AC-VCSP le problème de déterminer s'il existe une relaxation de valuation inférieure à α non arc-incohérente (non détectée comme incohérente par le filtrage par arc-cohérence).

  MAX-AC-VCSP est **NP-complet**.

CSP additifs
Améliorer l'efficacité

- **ordre de valeurs** : faire décroître le majorant plus vite.
- **ordre de variables** : faire croître le majorant plus vite.
- **majorant** : recherche un premier majorant par recherche locale...
- **minorant** : aussi fort que possible, aussi peu couteux à calculer que possible. Compromis temps/puissance similaire à celui de la cohérence locale.

Point essentiel : améliorer le minorant.

CSP additifs: minorants

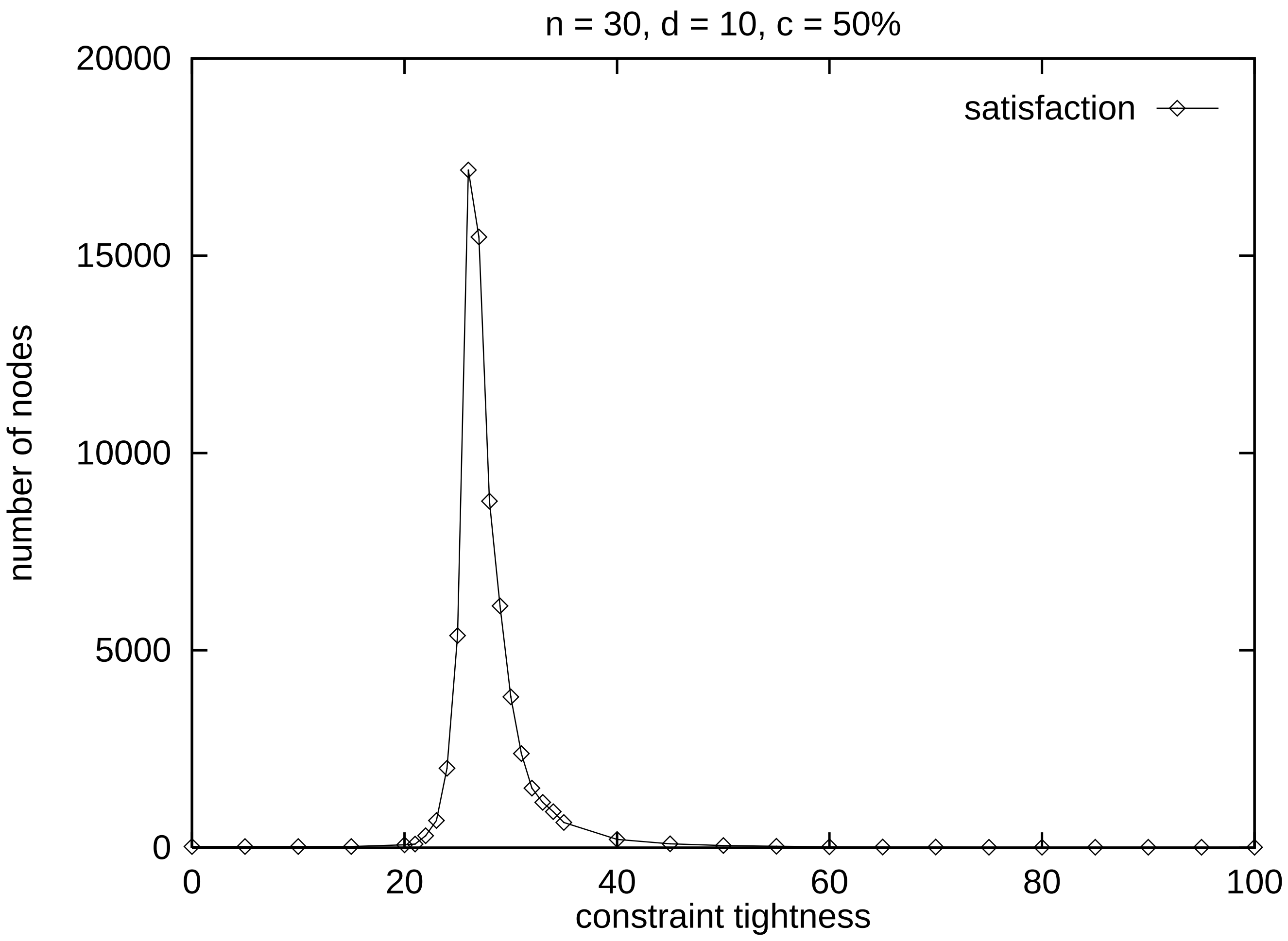
- **Backward checking** (BC) : exploite les contraintes en variables affectées.
- **Forward-checking** (FC) : exploite les contraintes entre variables affectées et non affectées.
- restent les contraintes entre variables non affectées :
  - **poupées russes** (RDS, G. Verfaillie, M. Lemaître et T. Schiex, AAAI96) : calculer la valuation du problème défini par les variables non affectées et la combiner au minorant (BC+FC).
  - **cohérence d'arc** : non définie. Ce qui reste possible : combiner des valuations dont les **justifications**, en termes de contraintes, sont **disjointes** dans des compteurs externes au CSP.

CSP additifs
compteurs d'arc incohérence

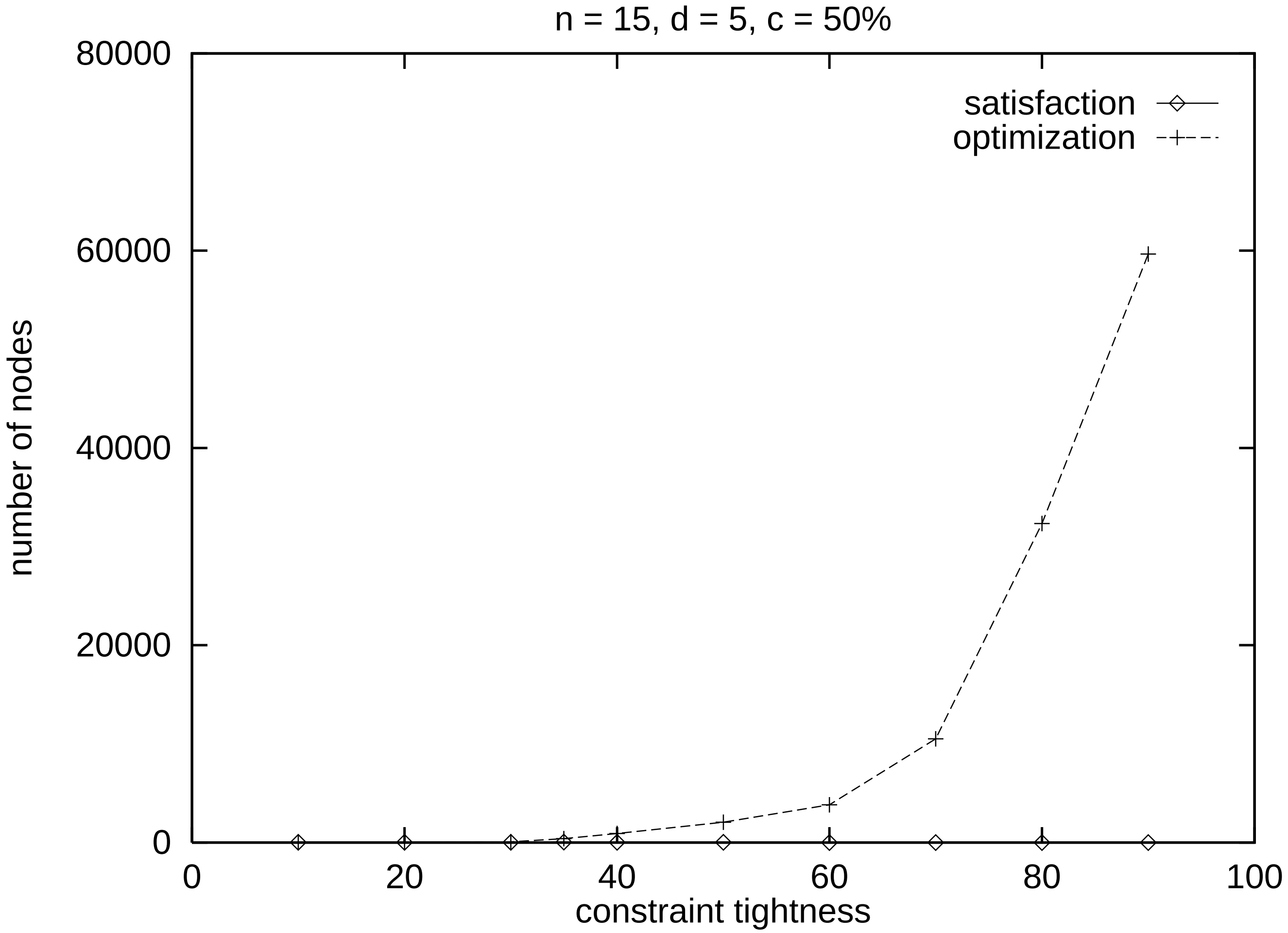
- **Directed Arc-Consistency Counts** (DAC, R. Wallace, ECAI94) : ordre statique des variables. Chaque contrainte est exploitée dans une direction et une seule fixée au départ.
- **Reversible DAC** (RDAC, J. Larrosa, P. Meseguer, T. Schiex, G. Verfaillie, AAAI98, Artificial Intelligence 1999) : ordre dynamique. Chaque contrainte est exploitée dans une direction qui change au cours de la recherche.
- **Weighted Arc-Consistency Counts** (WAC, M. Affane, H. Bennaceur, ECAI98) : chaque contrainte est exploitée en partie dans un sens, en partie dans l'autre. Généralise le précédent. Liberté supplémentaire difficile à exploiter.

Le goût de la cohérence locale sans la cohérence locale !

Quelques expérimentations
CSP aléatoires binaires classiques

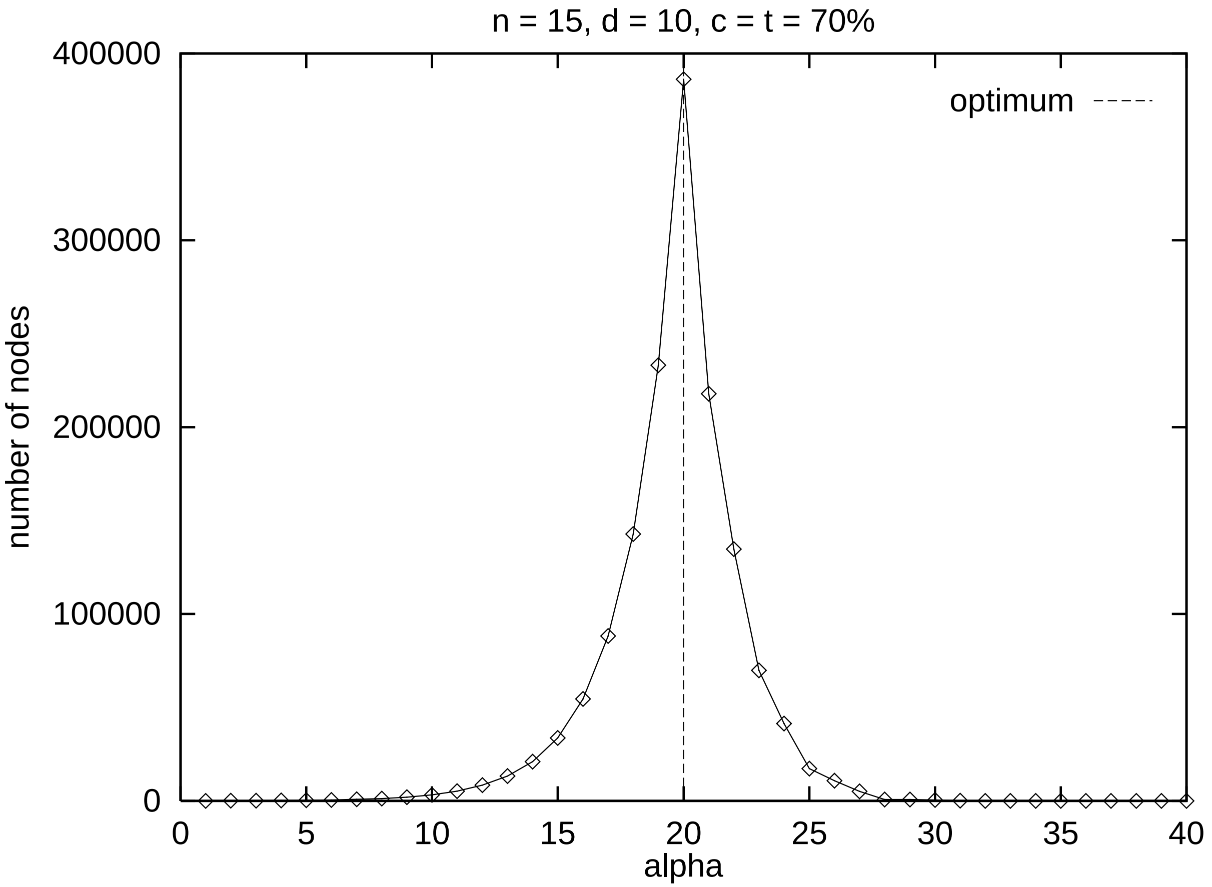


Quelques expérimentations
CSP aléatoires binaires additifs



Quelques expérimentations CSP aléatoires binaires additifs

Probléme $P(\alpha)$: Existe-t-il une affectation de valuation strictement inférieure à α ?



Observations

- la **preuve d'incohérence** ($P(1)$) est un des problèmes les plus faciles ;
- la **preuve d'optimalité** ($P(opt)$) est le plus difficile ;
- la **preuve d'optimalité** ($P(opt)$) est plus difficile que la **production d'une solution optimale** ($P(opt+1)$) ;
- un algorithme de type **séparation et évaluation** en profondeur d'abord résoud une **séquence de problèmes** $P(\alpha)$; normalement, il a au moins à résoudre $P(opt+1)$ et $P(opt)$;
- lui fournir une **bonne solution** de départ, éventuellement optimale, ne le dispense pas de résoudre le problème $P(opt)$.

Principale leçon

Résoudre un problème d'**optimisation** implique de travailler dans la zone du **pic de complexité**, à la frontière entre cohérence et incohérence.

Résoudre un **CSP additif** implique de travailler dans cette zone sans avoir à sa disposition les **outils nécessaires** (propagation de contraintes).

Gestion d'un satellite d'observation

Étant donné un **ensemble de prises de vues** S, selectionner un **sous-ensemble** S' qui satisfait toutes les contraintes dures et qui maximise la somme des poids des prises de vues.

- trois **instruments** sont disponibles
- chaque prise de vue **nécessite** certains instruments, avec un éventuel **temps de transition** pour reconfiguration (contraintes binaires)
- la bande passante du **bus de données** est limitée (contraintes ternaires)
- la **mémoire locale** est limitée (contrainte globale)

Résultats SPOT5

val = prises de vues sélectionnées

b = largeur de bande de l'ordre statique utilisé

$*$ = preuve d'optimalité (temps limité à 30')

				FC		*rds*	
pb	n	e	b	val	time	val	time
404	100	610	33	48	1800	49*	0.5
408	199	2032	62	3076	1800	3082*	14
412	300	4048	77	15078	1800	16102*	29
414	364	9744	150	21096	1800	22120*	86
503	105	403	30	8095	1800	9096*	2.5
505	240	2002	59	12088	1800	13100*	15
507	311	5421	103	12110	1800	15137*	55
509	348	8276	134	19104	1800	19125*	106

Affectation de fréquence (Cabon *et al.* 1999)

Étant donné un ensemble de **liens de communications** radio, affecter à chaque lien de communication deux fréquences en :

- respectant une **distance fixe** entre les deux fréquences d'un lien

$$|f_i - f_{i+1}| = K$$

- minimisant les **interférences** (poids des contraites violées).

$$|f_i - f_j| > \alpha_{ij} \qquad (w_{ij})$$

Problèmes CELAR, **aucune preuve d'optimalité** pour les instances sur-contraintes (CALMA).

Preuve d'optimalité par résolution du sous-problème SUB_4 de CELAR6 par **RDS** parallélisé (5.10^6 secondes de Sparc 5) (de Givry *et al.* 1997).

Autres résultats

En utilisant les algorithmes basés sur les **reversible DAC** (Sparc 2).

	PFC-DAC		PFC-RDAC		PFC-MRDAC	
	nœuds	cpu	nœuds	cpu	nœuds	cpu
SUB_1	8.1×10^8	5.2×10^4	1.9×10^7	3.5×10^3	1.8×10^6	2.6×10^3
SUB_2	4.1×10^9	2.5×10^5	1.2×10^8	2.9×10^4	1.6×10^7	2.3×10^4
SUB_3	$*$	$*$	2.7×10^8	7.1×10^4	3.7×10^7	6.8×10^4
SUB_4	$*$	$*$	1.2×10^9	3.4×10^5	1.2×10^8	2.6×10^5

Depuis, CELAR6 résolu directement par **programmation dynamique** (A. Koster, Maastricht).

Deux pistes de recherche essentielles

- utilisation de la **programmation dynamique** au delà des VCSP dont le graphe de contraintes est **acyclique** ;

- meilleur **calcul de minorant** pour les méthodes de type **séparation et évaluation**.

Librairies/algorithmes pour la résolution de VCSP

- **ftp://ftp.cert.fr/pub/lemaitre/LVCSP/**
  - algorithmes de type **séparation et évaluation** ;
  - algorithmes de type **recherche locale** ;
  - algorithmes **hybrides**.
- **http://www.lsi.upc.es/~larrosa/PFC-MRDAC**

  algorithmes basés sur les **reversible DAC**. Les plus efficaces à ma connaissance.

Références

S. Bistarelli, U. Montanari, and F. Rossi. Constraint solving over semirings. In *Proc. of the 14$^{\text{th}}$ IJCAI*, Montréal, Canada, August 1995.

B. Cabon, S. de Givry, L. Lobjois, T. Schiex, and J.P. Warners. Radio link frequency assignment. *Constraints Journal*, 4:79–89, 1999.

Simon de Givry, Gérard Verfaillie, and Thomas Schiex. Bounding the optimum of constraint optimization problems. In *Proc. of CP'97*, Austria, October 1997.

R. Dechter, A. Dechter, and J. Pearl. Optimization in constraint networks. In R.M Oliver and J.Q. Smith, editors, *Influence Diagrams, Belief Nets and Decision Analysis*, chapter 18, pages 411–425. John Wiley & Sons Ltd., 1990.

H. Fargier and J. Lang. Uncertainty in constraint satisfaction problems: a probabilistic approach. In *Proc. of ECSQARU '93, LNCS 747*, pages 97–104, Grenade, Spain, November 1993.

H. Fargier, J. Lang, and T. Schiex. Selecting preferred solutions in Fuzzy Constraint Satisfaction Problems. In *Proc. of the 1$^{\text{st}}$ European Congress on Fuzzy and Intelligent Technologies*, 1993.

Hélène Fargier. *Problèmes de satisfaction de contraintes flexibles et application à l'ordonnancement de production*. Thèse de doctorat, Institut de Recherche en

Informatique de Toulouse (Université Paul Sabatier), Toulouse, France, June 1994.

E.C. Freuder and R.J. Wallace. Partial constraint satisfaction. *Artificial Intelligence*, 58:21–70, December 1992.

Eugene C. Freuder. Partial constraint satisfaction. In *Proc. of the 11*$^{\mathrm{th}}$ *IJCAI*, pages 278–283, Detroit, MI, 1989.

M. Minoux. Structures algébriques généralisées des problèmes de cheminement dans les graphes. *Recherche opérationnelle*, 10(6):33–62, 1976.

A. Rosenfeld, R. Hummel, and S. Zucker. Scene labeling by relaxation operations. *IEEE Trans. on Systems, Man, and Cybernetics*, 6(6):173–184, 1976.

T. Schiex. Possibilistic constraint satisfaction problems or "How to handle soft constraints ?". In *Proc. of the 8*$^{\mathrm{th}}$ *Int. Conf. on Uncertainty in Artificial Intelligence*, Stanford, CA, July 1992.

G. Shafer and P. Shenoy. Local computations in hyper-trees. Working paper 201, School of business, University of Kansas, 1988.

L. Shapiro and R. Haralick. Structural descriptions and inexact matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3:504–519, 1981.

P. Shenoy. Valuation-based systems for discrete optimization. In Bonissone, Henrion, Kanal, and Lemmer, editors, *Uncertainty in AI*. North-Holland Publishers, 1991.

P. Snow and E.C. Freuder. Improved relaxation and search methods for approximate constraint satisfaction with a maximin criterion. In *Proc. of the 8$^{\mathrm{th}}$ biennal conf. of the canadian society for comput. studies of intelligence*, pages 227–230, May 1990.