



HAL
open science

Contraintes globales et décompositions

Christian Bessiere

► **To cite this version:**

Christian Bessiere. Contraintes globales et décompositions. 8èmes Journées Francophones de Programmation par Contraintes (JFPC 2012), May 2012, Toulouse, France. hal-00830336

HAL Id: hal-00830336

<https://inria.hal.science/hal-00830336>

Submitted on 15 Feb 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Contraintes globales et décompositions

Christian Bessiere


Univ Montpellier, CNRS

(joined work with E. Hebrard, B. Hnich, G. Katsirelos,
Z. Kiziltan, N. Narodytska C.G. Quimper, T. Walsh)

Plan de l'exposé

- Programmation par contraintes
- Contraintes globales
- Décompositions
- Résultat de non décomposabilité
- Décompositions étendues

Réseau de contraintes

- Variables $X = \{x_1, \dots, x_n\}$
- Domaines $D(x_i)$: ensemble **fini** de valeurs pour x_i
- Contraintes $C = \{c_1, \dots, c_i, \dots\}$, où c_i spécifie les combinaisons de valeurs autorisées sur $X(c_i) = (x_{i_1}, \dots, x_{i_r})$, la **portée** de c_i
- Exemples
 - c_1 : $x_1 < x_2$ est l'ensemble de tuples (v, w) in Z^2 tels que $v < w$
 - c_2 : $X(c_2) = (x_1..x_3)$ en extension: 

112
213
342
353
413
511

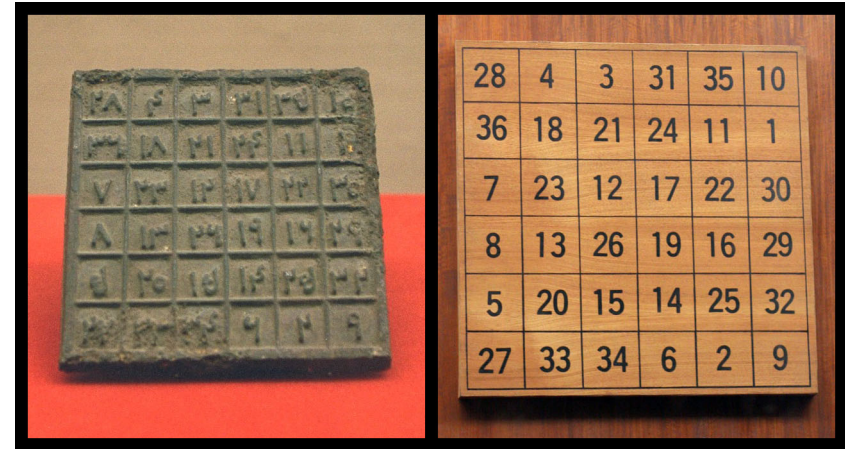
Problème de satisfaction de contraintes (CSP)

- Etant donné un réseau (X,D,C) ,
→ trouver une **solution**, c'est à dire une instantiation de X sur D qui satisfait toutes les contraintes de C

CSP est NP-difficile

Exemple : magic square

- $X = \{x_{11}, x_{12}, \dots, x_{ij}, \dots, x_{nn}\}$
- $D(x_{ij}) = \{1..n^2\}$
- C:
 - $\text{sum}(x_{i1} \dots x_{in}) = k, \forall i$ (rows)
 - $\text{sum}(x_{1j} \dots x_{nj}) = k, \forall j$ (columns)
 - $\text{sum}(x_{11} \dots x_{ii} \dots x_{nn}) = k$ (diagonal 1)
 - $\text{sum}(x_{1n} \dots x_{i(n-i+1)} \dots x_{n1}) = k$ (diagonal 2)
 - $\text{Alldifferent}(x_{11}, x_{12}, \dots, x_{ij}, \dots, x_{nn})$



Résoudre un problème à contraintes

Function Solve (P)

propagate (P)

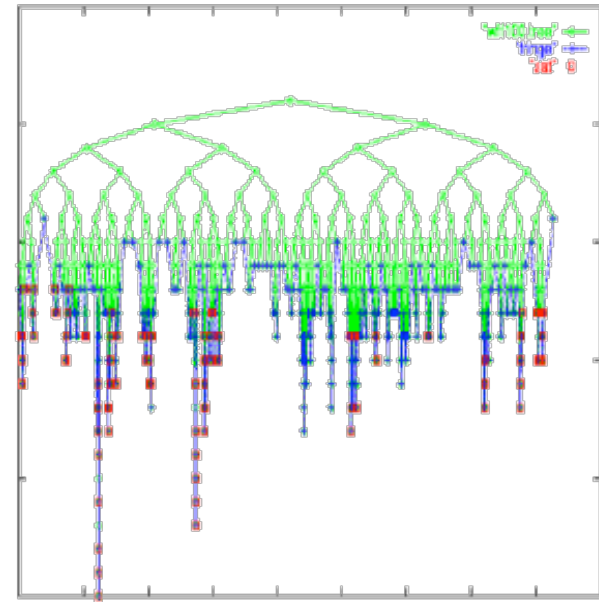
if empty domain then return 0

if P fully instantiated then return 1

select variable X_i and value v

if Solve (P + { $X_i=v$ }) then return 1

else return Solve (P + { $X_i \neq v$ })



Efficace quand **propagate** réduit beaucoup l'espace de recherche

Propager l'arc cohérence

$$D(x)=\{0,2,4\}, D(y)=\{1,2,3\},$$

$$D(z)=\{\cancel{0}, 1, 2, 3, 4, 5, 6, 7, \cancel{8}, \cancel{9}\}$$

$$x + y = z$$

Arc cohérence sur la contrainte $c(x,y,z)$:
Supprimer **toutes** les valeurs de x,y,z qui ne peuvent pas satisfaire la contrainte c

$x+y=z$

011

022

033

213

224

235

415

426

437

Propager l'arc cohérence

$$D(x)=\{0,2,4\}, D(y)=\{1, \cancel{2}, 3\},$$

$$D(z)=\{\cancel{0}, 1, \cancel{2}, 3, \cancel{4}, 5, \cancel{6}, 7, \cancel{8}, \cancel{9}\}$$

$$x + y = z$$

$$y \neq 2$$

Arc cohérence sur la contrainte $c(x,y,z)$:
Supprimer **toutes** les valeurs de x,y,z qui ne peuvent pas satisfaire la contrainte c

$x+y=z$

011

~~022~~

033

213

~~224~~

235

415

~~426~~

437


Complexité de l'arc cohérence

Algorithmes optimaux pour l'arc cohérence en $O(d^r)$, où r est le nombre de variables de la contrainte et d est la taille des domaines

Contraintes globales

- Contraintes qui peuvent porter sur un nombre quelconque de variables
 - $\text{Alldifferent}(x_1, \dots, x_n) \Leftrightarrow x_i \neq x_j \quad \forall i, j$
 - $\text{sum}(x_1, \dots, x_n, K) \Leftrightarrow \sum x_i = K$

Pourquoi des contraintes globales ?

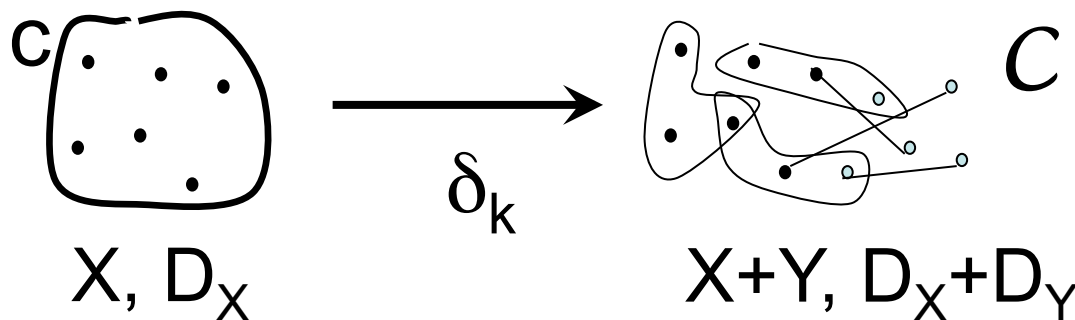
- Utile pour exprimer des relations complexes entre variables (→ frequent patterns in applications)
 - Alldifferent : two courses cannot occur simultaneously
 - Atleast_{k,v} : at least two hostesses must speak japanese
 - Stretch : no more than 5 working days; not morning after night --> **N N R R M M A A A A R N M M R R** (nurse rostering)


Pourquoi des contraintes globales ?

- Expressivité, mais aussi propagation forte
 - ➔ Les contraintes globales ont permis de résoudre des problèmes ouverts
 - Sport league scheduling, time-tabling, etc.
- Les contraintes globales sont une spécificité de la CP
- La plupart (tous?) des solvers contiennent des contraintes globales
- 354 global constraints dans Beldiceanu's catalog
- Mais les algorithmes génériques d'arc cohérence sont en $O(d^r)$...
- ➔ On doit implémenter un propagateur **ad hoc** pour chaque contrainte du solveur!

A-t-on besoin de 300 contraintes globales?

- Non!
- On peut les réécrire en CNF (SAT solveurs)
- On peut les **décomposer en contraintes** plus 'simples' (arité fixe, en général 2 or 3)



$\text{Sol}(N) = \text{sol}(\delta_k(N))[X]$
 $|X(c_i)| \leq k, \forall c_i \in C$
 $|\delta_k(c)|$ is polynomial

$N = (X, D_X, \{c\})$

Pourquoi les décomposer ?

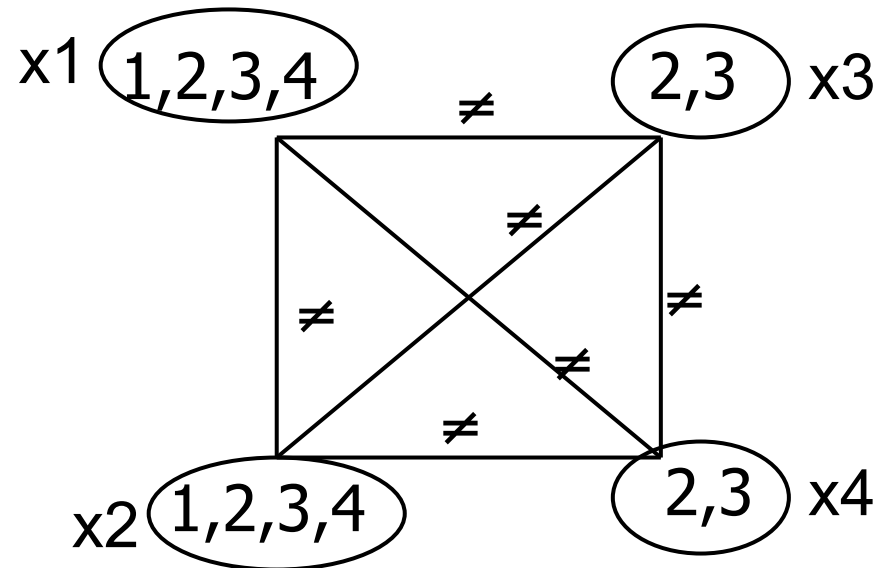
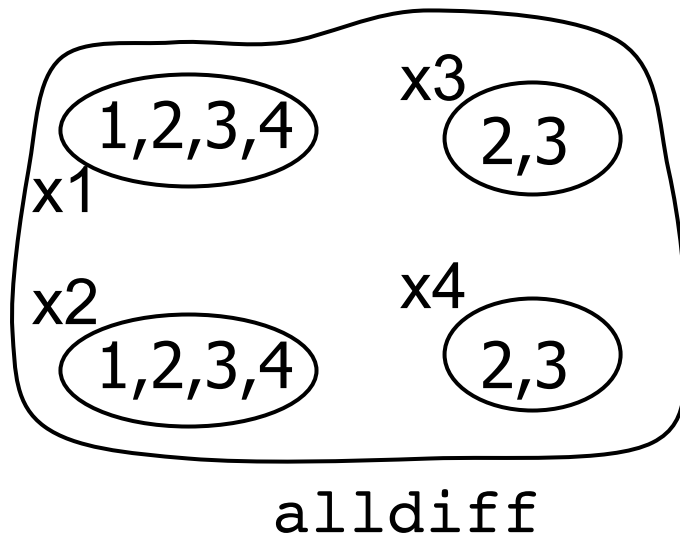
- Économiser du temps/effort pour le concepteur du solveur
- Incrémentalité inhérente
- Perspectives pour les SMT solvers:
 - Le solveur SAT reçoit des explications des contraintes globales
 - Il est important d'avoir des explications courtes

Décompositions

- Que peut-on attendre d'une décomposition ?
- D'exprimer la même chose
 - ***Semantic*** decomposition
- The permettre la même propagation (e.g., arc consistency)
 - ***Operational*** decomposition

Semantic decomposability (no extra variables)

- Alldiff



Solutions of the CSP on the right are the same as the allowed tuples of the Alldiff on the left

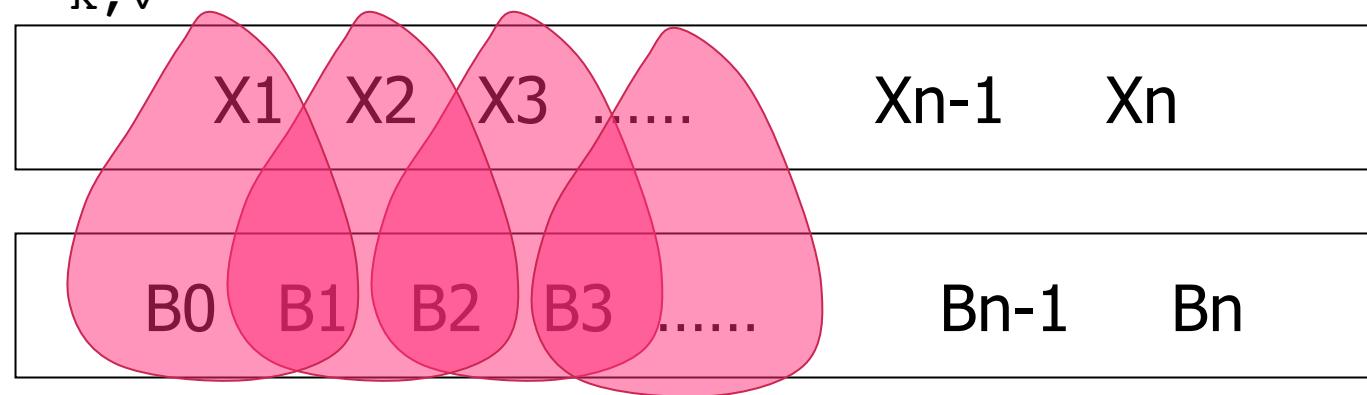
Semantic decomposability (extra variables)

- $\text{Atleast}_{k,v}$

X1	X2	X3	Xn-1	Xn
----	----	----	-------	------	----

Semantic decomposability (extra variables)

- $\text{Atleast}_{k,v}$

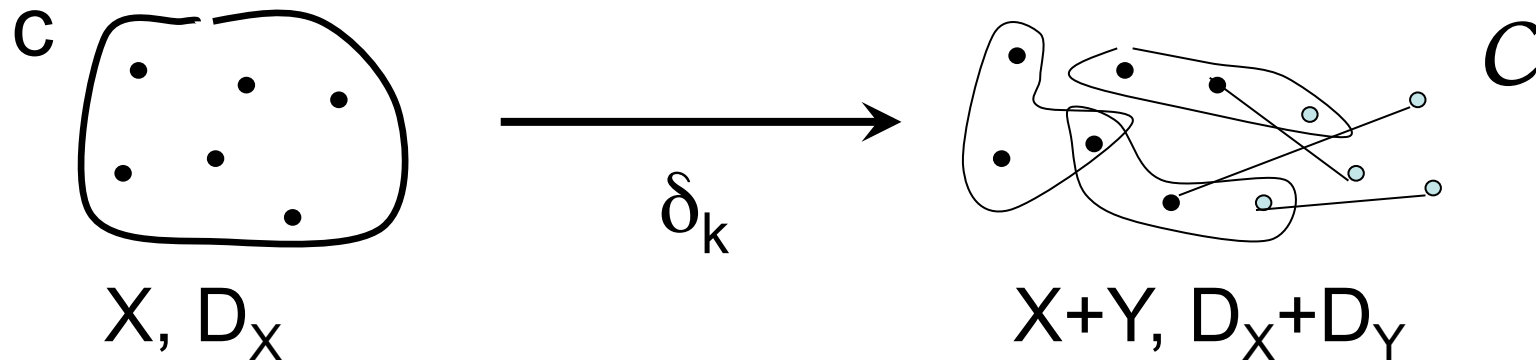


- $B_0 \dots B_n, D(B_i) = \{0, \dots, n\}$
- $(X_i = v \ \& \ B_i = B_{i-1} + 1) \vee (X_i \neq v \ \& \ B_i = B_{i-1}), \forall i$
- $B_0 = 0, B_n \geq k$

Solutions of this CSP projected on the X_i 's
are the same as the tuples allowed by Atleast

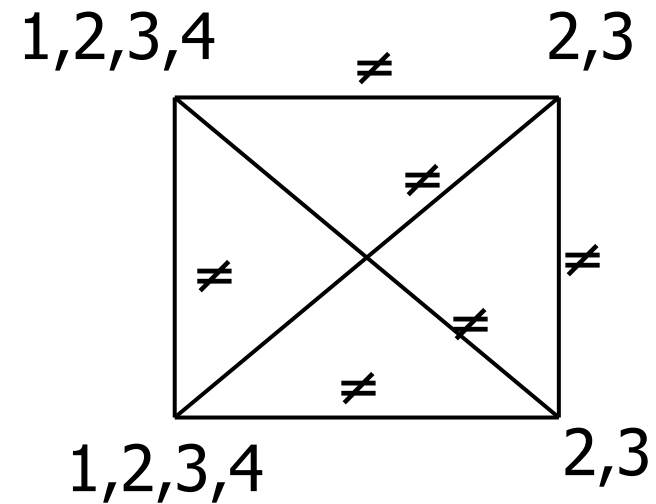
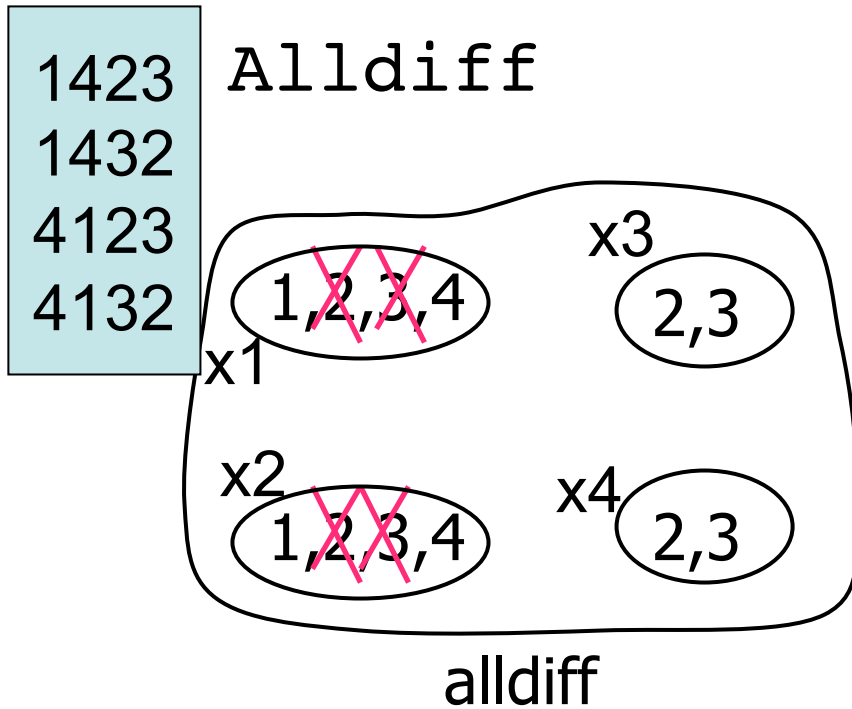
Operational decomposability (AC-decomposition)

- On veut non seulement exprimer la même chose que la contrainte globale, mais aussi **préserver** le niveau de propagation (i.e., arc cohérence)



For any $D'_X \subseteq D_X$: $AC(\{c\}) = AC(C)|_X$

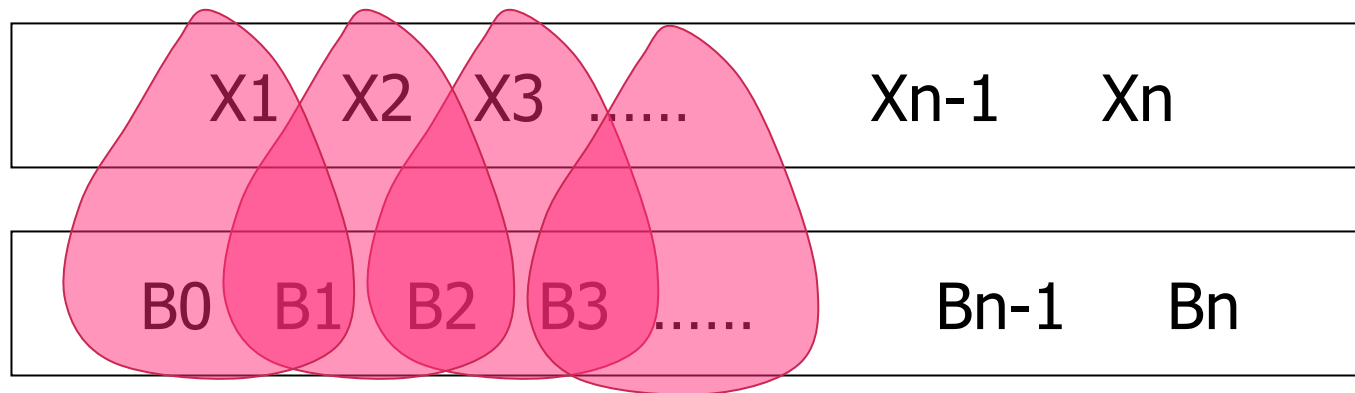
Example 1



This decomposition hinders propagation

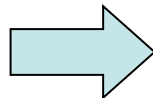
Example 2

- Atleast



- $B_0 \dots B_n, D(B_i) = \{0, \dots, n\}$
- $(X_i = v \ \& \ B_i = B_{i-1} + 1) \vee (X_i \neq v \ \& \ B_i = B_{i-1}), \forall i$
- $B_0 = 0, B_n \geq k$

Acyclic hypergraph



This decomposition preserves propagation

'Chain-like' AC-decomposition

- Beaucoup de contraintes peuvent être décomposées en chaîne de contraintes ternaires qui forment un hypergraphe *Berge-acyclique* (→ AC est préservée)
- E.g., `Atleast`, `Atmost`, `consecutive-1`, `lex`, `stretch`, `regular`

Taxonomie (de ces 354) ?

- A l'aide des outils de théorie de la complexité
- Soit c une contrainte globale sur $X(c)=(x_1 \dots x_n)$
- $\text{checker}(c) \Leftrightarrow$ "existe-t-il un tuple de $D(x_1) \times \dots \times D(x_n)$ satisfaisant c ?"
- **Si** $\text{checker}(c)$ est NP-complet
alors propager c est NP-difficile
alors pas de AC-decomposition pour c

Contraintes NP-difficiles

- Elles peuvent être détectées par réduction polynomiale... et il y en a beaucoup !
- Exemples:
 - $Nvalue(N, x_1, \dots, x_n)$ (N = number of values used by x_1, \dots, x_n)
 - $Sum(x_1, \dots, x_n, K)$
- Cette technique permet de découvrir que certains propagateurs ne sont pas complets (\rightarrow ils ne suppriment pas toutes les valeurs arc incohérentes)

Relaxer la propagation: borne cohérence (BC)

$D(x) = \{0, 2, 4\}$, $D(y) = \{1, \cancel{2}, 3\}$,

$D(z) = \{\cancel{0}, 1, \textcircled{2}, 3, \textcircled{4}, 5, \textcircled{6}, 7, \cancel{8}, \cancel{9}\}$

$x + y = z$

$y \neq 2$

$x+y=z$

011

022

033

213

224

235

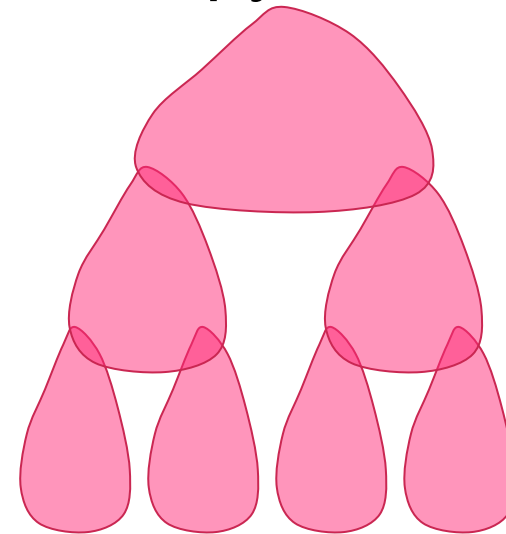
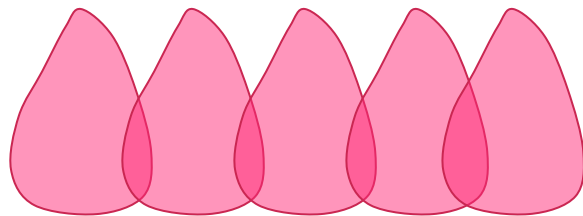
415

426

437

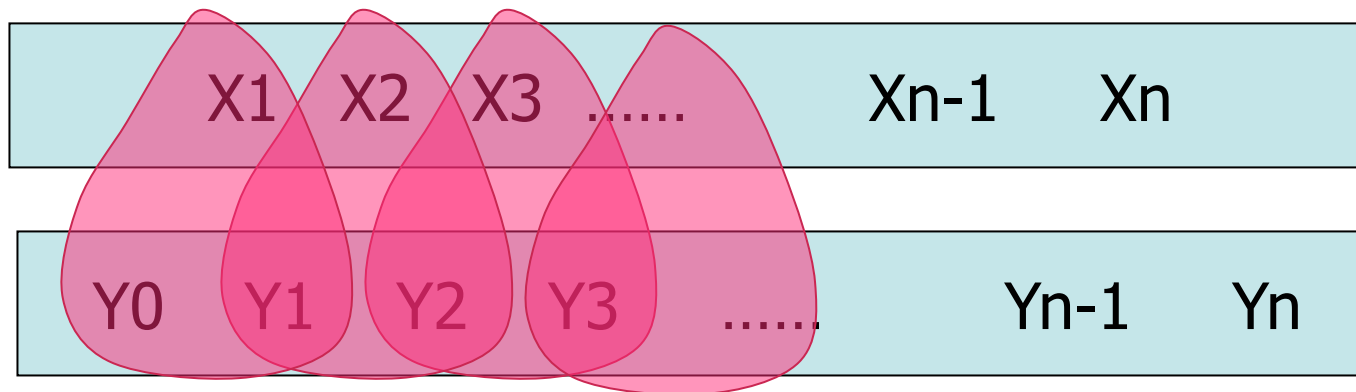
BC-decompositions

- Plusieurs contraintes importantes pour lesquelles AC est NP-difficile ont des BC-decompositions en contraintes ternaires structurées en chaîne (sum) ou en pyramide (Nvalue)



Attention: taille de la décomposition

Exemple: $\text{sum}(x_1, \dots, x_n, K)$



- $Y_i = Y_{i-1} + X_i, \forall i$
- $Y_0 = 0, Y_n = K$
- $D(Y_i) = ???$

$D(X_1) = \{0, 1, \dots, 9\}$; $D(X_2) = \{0, 10, \dots, 90\}$; $D(X_3) = \{0, 100, \dots, 900\}$;
 $D(X_4) = \{0, 1000, \dots, 9000\} \dots$

- ➔ For AC, $D(Y_4)$ must contain 10^4 values ➔ exponential size
- ➔ BC can use the interval domain $[0, \dots, 999]$

Etat des lieux

- Contraintes NP-difficiles à propager
 - pas de AC-decomposition
(parfois une BC-decomposition)
- Contraintes polynomiales à propager
 - AC-decomposition quand on en trouve une
(atleast, stretch, regular, etc.)
 - Quid des autres ???

Non AC-decomposabilité

- AC-decomposition pour c
 - ⇒ décomposition en CNF qui calcule AC
 - ⇒ CNF checker (= décide si la contrainte a un tuple solution dans le domaine)
- CNF checker
 - ⇒ codé en monotone circuit de taille poly

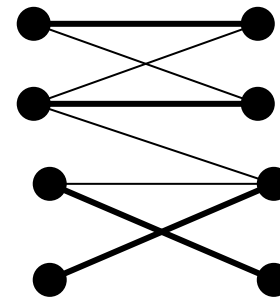
Théorème

Si pas de poly-size monotone circuit ***alors*** pas de CNF qui calcule AC ni de AC-decomposition

Complexité des circuits

- Classes de fonctions qui ne peuvent pas être calculées par des monotone circuits de poly size [Rasborov 85, Tardos 88]

- Exemple:
 - perfect matching
 - Subsumed by `checker(alldiff)` [Knuth92, Regin94]



$$X_1 \in \mathbf{1,2}$$

$$X_2 \in \mathbf{1,2,3}$$

$$X_3 \in \mathbf{3,4}$$

$$X_4 \in \mathbf{3}$$

→ `alldiff` n'a pas de AC-décomposition ni de CNF

- Other examples: `global cardinality`, `same`, etc.²⁹

So what?

- La prog par contraintes ***ne peut pas*** se réduire à SAT
- La prog par contraintes ***ne peut pas*** se réduire aux contraintes avec arité bornée

Résumé

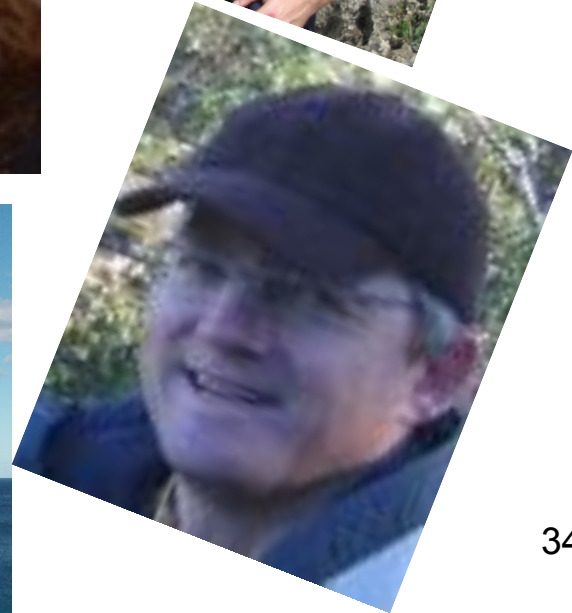
- NP-hard constraints
 - Use a lower level of consistency (eg., bound consistency)
- AC-decomposable constraints
 - Use the decomposition (when we know it!)
- Constraints that are poly but non AC-decomposable
 - ***You must implement the poly algorithm :(***
 - ...or use a lower level of consistency

Décomposer plus

- Idée : équiper les solvers d'un ensemble \mathcal{L} de quelques (une douzaine ?) contraintes globales qui codent toutes les autres
 - AC(\mathcal{L})-decomposability of c :
 - c peut être décomposée en contraintes de \mathcal{L}
 - pas de nouveau propagateur à implémenter !
- Exemples:
 - `range + roots` peuvent exprimer 70 constraints in the catalog (version of 2005 with 214 constraints)
 - `Slide` (or `counter`) expresses many others

Global Cost Functions

- A cost function $c(x_i \dots x_j)$ gives a cost to each tuple in $D(x_i) \times \dots \times D(x_j)$
- Decompose global cost functions?
 - See de Givry's talk at the end of this morning! 😊



Some references

Deriving Filtering Algorithms from Constraint Checkers
N. Beldiceanu, M. Carlsson, T. Petit.
Proceedings CP'04, Toronto CA, pages 107-122.

Global Constraint Catalogue: Past, Present and Future
N. Beldiceanu, M. Carlsson, S. Demasse, T. Petit.
Constraints 12 (1): 21-62 (2007).

The Problem of Compatible Representatives
D.E. Knuth, A. Raghunathan.
SIAM J. Discrete Math. 5(3): 422-427 (1992).

A Filtering Algorithm for Constraints of Difference in CSPs
J.C. Régin.
Proceedings AAAI'94, Seattle WA, pages 362-367.

Modeling and solving sports league scheduling with constraint Programming
J.C. Régin.
Proceedings INFORMS'98, Montreal, Canada.

To be or not to be...a global constraint
C. Bessière and P. Van Hentenryck.
Proceedings CP'03, Kinsale, Ireland, pages 789-794.

Local consistencies in SAT
C. Bessière, E. Hebrard and T. Walsh.
Proceedings SAT'03, LNCS 2919, pages 299-314.

The Complexity of Reasoning with Global Constraints
C. Bessière, E. Hebrard, B. Hnich, T. Walsh.
Constraints, volume 12(2), pages 239-259 (2007).

Range and Roots: Two Common Patterns for Specifying and Propagating Counting and Occurrence Constraints
C. Bessière, E. Hebrard, B. Hnich, Z. Kiziltan, T. Walsh.
Artificial Intelligence, volume 173(11), pages 1054-1078 (2009).

Circuit Complexity and Decompositions of Global Constraints
C. Bessière, G. Katsirelos, N. Narodytska, T. Walsh.
Proceedings IJCAI'09, Pasadena CA, pages 412-418.

Decompositions of All Different, Global Cardinality and Related Constraints
C. Bessière, G. Katsirelos, N. Narodytska, C.G. Quimper, T. Walsh.
Proceedings IJCAI'09, Pasadena CA, pages 419-424.

SLIDE: A Useful Special Case of the CARDPATH Constraint
C. Bessière, E. Hebrard, B. Hnich, Z. Kiziltan, T. Walsh.
Proceedings ECAI'08, Patras, Greece, pages 475-479.