Constraint networks

Thomas Schiex
INRA

Constraint Network - Minimal definition

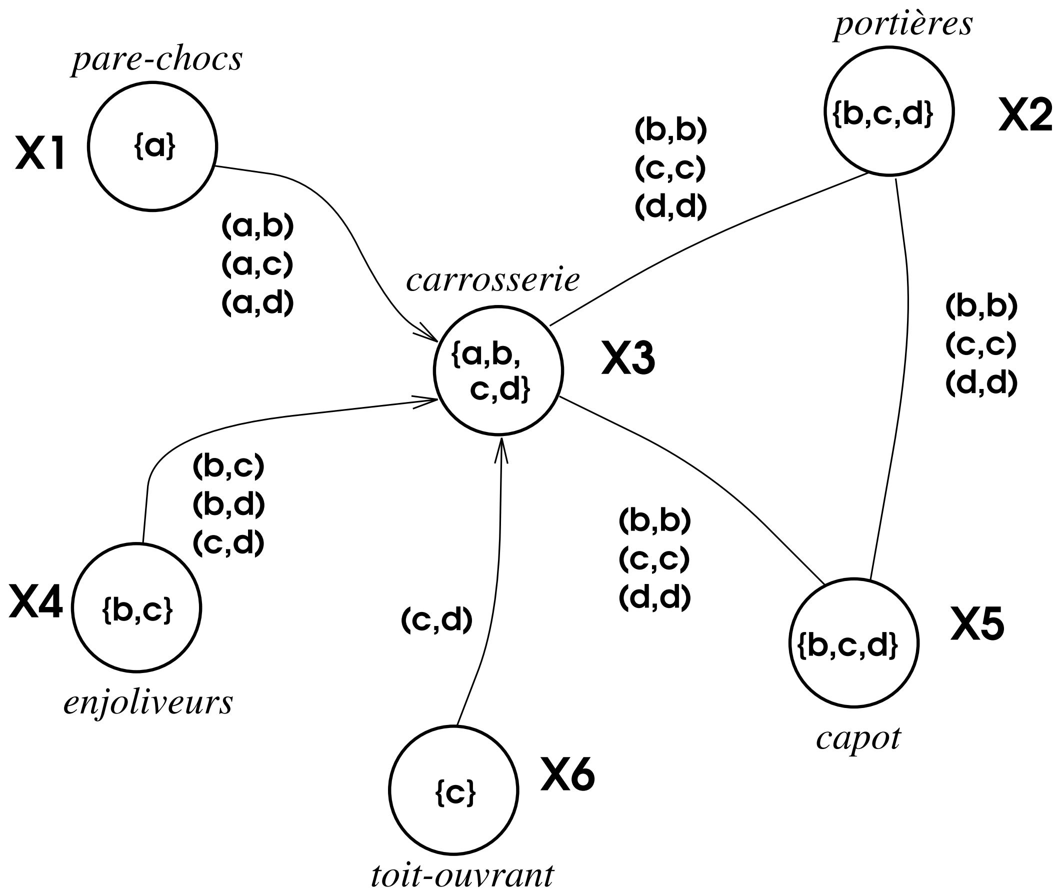
A constraint network (X, D, C):

- $X = \{1, \ldots n\}$ is a set of **variables**.
- $D = \{d_1, \ldots d_n\}$ is a set of finite **domains**.
- $C = \{c_1, \ldots, c_e\}$ is a set of **constraints**.

Each constraint $c \in C$ involves some variables in X and restricts the combination of values these variables may take to a given set of tuples.

The **Constraint Satisfaction Problem**: find an assignment of values to variables such that no forbidden combination of value is used. The decision problem is NP-complete.

Constraint Networks -Example



Constraint networks applications: vision, natural language processing, sequencing, timetabling, assignment, constraint programming. . .

Main limitations

Tentative applications of the CSP technology to real problems raise several issues:

- **Efficiency**: algorithms for the CSP are **time exponential**.
- **Immutability**: the set of constraints is supposed to be **fixed**.
- **Rigidity**: **all** constraints must be **completely** satisfied.

Central motivation: trying to address/understand these limitations.

Arc consistency

One of the simplest and most widely used level of local consistency.

A binary CSP is **arc consistent** if any **single value** assignment can be extended to **any other variable** without any constraint **violation** (viable value)

Arc consistency enforcing: remove non viable values until quiescence.

- massive inference of **unary** constraints preserving equivalence.
- optimal algorithms: AC4/AC6/AC7 $O(ed)$ space and $O(ed^2)$ time.
- may be used as preprocessing or during backtrack tree search (MAC algorithm (Sabin and Freuder 1994; Bessière *et al.* 1995)) as inconsistency detector.

Lazy arc consistency (Schiex *et al.* 1996)

Central idea: instead of producing the complete arc consistent closure, just prove that it is **non empty** (Lazy FC (Zweben and Eskey 1989)).

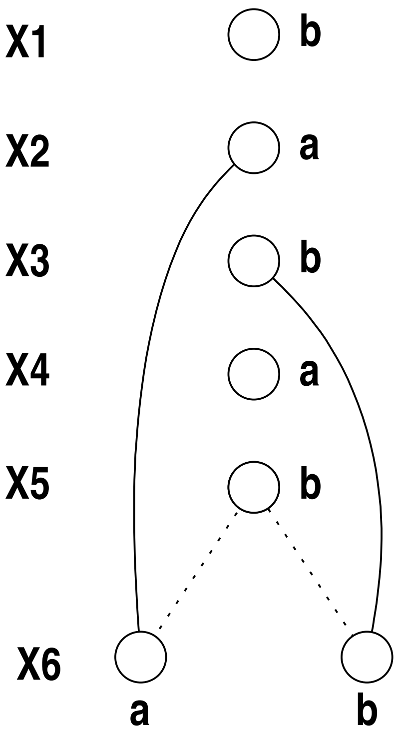
The Lazy AC7 algorithm produced:

- keeps the $O(ed)$ space/$O(ed^2)$ time complexities of AC7
- may be orders of magnitude **faster** and only marginaly slower
- provides extreme **incrementality**
- is still able to identify the **maximum cardinality** arc consistent domain (variale ordering heuristics).

To do: integrate inside Backtrack-tree search (MLAC).

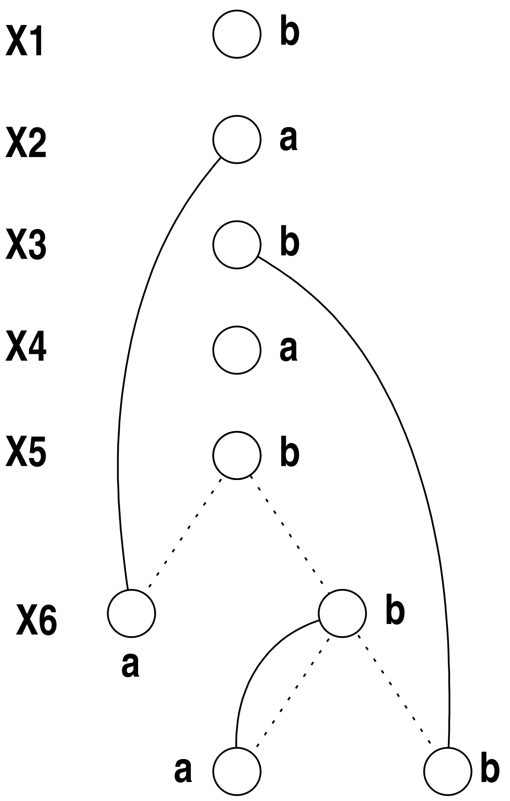
Filtering from failure

During backtrack tree-search, there are situations where one may filter for free.



- **Shallow learning** (Dechter 1990).
- Potential **exponential** number of nogoods learned ⇒ limitation.
- **Backjumping** (BJ): fire and forget nogoods (Gaschnig 1979).

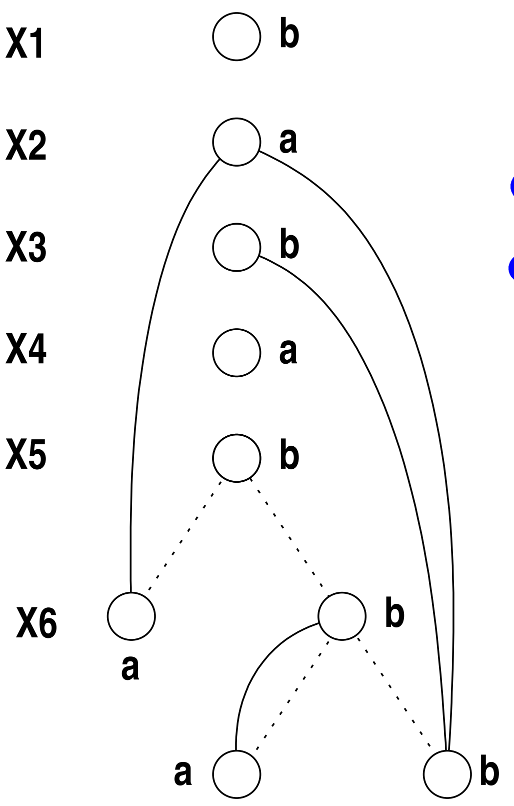
Filtering more: Nogood Recording (Schiex and Verfaillie 1993)



- **NR algorithm**: more filtering, more efficient.
- performs (CBJ) backjumping (Prosser 1993).
- **NR-FC**: extended to work with FC filtering
- **DBT-FC** (Verfaillie and Schiex 1994b): similar ideas applied to DBT (Ginsberg 1993). DBT-MAC is promising (Jussien *et al.* 2000).

Filtering even more: Stubborn NR (Schiex and Verfaillie 1994)

We only build nogoods from **failures**. We should try to fail more.



Central idea: stop constraint checking at the **second** (not first) failure.

- different possible backjumping points (High/Low)
- better efficiency than NR-FC on Prosser's Zebra benchmark
- **puzzling results**. More tests needed

Dynamic CSP

Typical real problems are **not fixed**: constraints/variables/values appear and disappear as time goes. Need for **incremental algorithms**.

Existing work:

- **local consistency**: DnAC4 (Bessière 1991).
- **solution maintenance**: oracles (van Hentenryck 1990).

Two aims:

- **efficient** behavior on **incremental** modifications
- minor solution disturbance on incremental modifications: **optimization**.

Dynamic CSP

Central ideas: reuse previous **solution** (conserved on **relaxation**) or previous **filtering** (conserved on **restrictions**).

- reuse of the previous solution: **Local changes** (Verfaillie and Schiex 1994a).
- reuse of the previous filtering: **justified nogoods** in NR(FC) (Schiex and Verfaillie 1993) and DBT-FC (Verfaillie and Schiex 1994b).

Important improvements. Local changes well adapted to under constrained problems, NR(-FC) to over and critically constrained ones, DBT-FC to both.

Modeling possible states

Not all states are possible. We may therefore use:

- a **CSP** to model the set of all **possible states**
- a **CSP** to model the **problem** to solve
- **relate** the two by constraints.

This defines a **mixed constraint network** (Fargier *et al.* 1996).

We can also use probability distributions on states (Fargier *et al.* 1995).

Modeling possible states

Possible problems:

- **consistency**: **for all** possible states, there **exists** a solution.
- **strong consistency**: there **exist** a solution that works **for all** possible states

Complexity at the **second level** of the PH (Fargier *et al.* 1996).

Incremental algorithms defined.

Probabilistic mixed CSP only NP-complete under independence assumptions (Fargier *et al.* 1995).

CSP and optimization

Real constraint based problems include:

- **hard** constraints
- **preferably satisfied** constraints
- **uncertain** constraints

Modeling such problems as CSP can lead to inconsistency (**over-constrained**) or to abundancy of poor quality solutions (**under-constrained**).

The ability of expressing soft constraints allows one to:

- handle both **over/under-constrained** problems
- define an homogeneous framework for **optimization in CSP**

Existing stuff

- **Max** CSP: minimize the sum of the weights of violated constraints (Shapiro and Haralick 1981; Freuder and Wallace 1992).
- **possibilistic/fuzzy** CSP: minimize the priority of the most important violated constraint (Rosenfeld *et al.* 1976; Schiex 1992).
- **lexicographic** CSP: minimize the number of violated constraint at each evel of priority from the most important to the least important ones (Fargier *et al.* 1993).
- **probabilistic** CSP: minimize the probability of not being a solution of the real problem (Fargier and Lang 1993).

Puzzling situation

In each extension, the usual CSP algorithms, theorems, properties and polynomial classes where (possibly) extended.

Large **redundancy** of work.

- **possibilistic/fuzzy CSP**: (almost) all usual algorithms and properties extend (Rosenfeld *et al.* 1976; Snow and Freuder 1990; Schiex 1992; Fargier 1994; Ruttkay 1994)
- **other extensions**: the majority of the usual properties and algorithms resist to extension. Problems are also harder to solve.

What is the source of this hardness ? Is it possible to bypass it ?

Main motivations

- **Generality**: to build a generic framework that covers existing proposals.
  - avoid redundant work.
  - analyze the differences of behavior.
- **Specificity**: to gather enough properties to be able to built non trivial theorems and generic algorithms.

Eventually: build an homogeneous framework for optimization with constraint networks.

The VCSP Framework (Schiex *et al.* 1995)

- with each **constraint** we associate a **valuation** that reflects its importance: preference, weight, priority, probability of existence. . .
- The valuation of an assignment is defined as the **combination** of the valuations of the **violated constraints** using an operator that must satisfy a set of axioms.
- The instanciations can be **compared** using a **total order** on valuations.
- The problem is to produce an assignment of **minimum valuation**.

Commutative totally ordered semigroup with a monotonic operator.

Similar approaches

- **(Minoux 1976)**, **Aho/Ullman** 197? : **semi-ring** used to generalize all-pair shortest path algorithms;
- **(Shafer and Shenoy 1988)** : a **semi-group** structure is used to generalize dynamic programming like algorithms (variable elimination, CSP: adaptative consistency, tree-clustering. . .);
- Reasoning under uncertainty: **triangular norms and co-norms** are based on semi-groups.
- **Partial CSP** (Freuder 1989), uses a notion of **distance** between problems.
- **(Bistarelli et al. 1995)** : a **semi-ring** structure is used to extend CSP, dynamic programming-like algorithms and local consistency algorithms;

Similar structures, different aims, different compromises.

More precisely

- $\mathrm{VCSP} = (V, D, C, S, \varphi)$, $S = (E, \succeq, \top, \bot, \oplus)$
- E = **set of valuations**, made of numbers or symbols, used to annotate constraints and to assess assignments;
- $\top$ = **maximum element** of E, used to annotate hard constraints, corresponds to totally inconsistent assignments;
- $\bot$ = **minimum element** of E, corresponds to completely consistent assignments;
- $\succeq$ = **total order** on E, used to compare two valuations;
- $\oplus$ = **operator** used to **combine** two valuations;
- $\varphi : C \rightarrow E$ = **valuation fonction**, used to annotate each constraint with a valuation.

Different sub-classes

CSP	E	$\succ$	$\top$	$\perp$	$\oplus$
classical	{t,f}	f $\succ$ t	f	t	$\wedge$
Max-CSP	$\overline{\mathbb{N}}$	$>$	$+\infty$	0	$+$
possibilistic	$[0, 1]$	$>$	1	0	max
lexicographic	$[0, 1]^*$	$>^*$	$\top$	$\emptyset$	$\cup$
probabilistic	$[0, 1]$	$>$	1	0	$a \oplus b = 1 - (1 - a)(1 - b)$

Two important additional properties

- **idempotency**: $\forall a \in E, a \oplus a = a$
- **strict monotonicity**: $\forall a, b, c \in E, a \succ b \wedge c \neq \top \Rightarrow (a \oplus c) \succ (b \oplus c)$

Using VCSP axioms, we have:

- $|E| > 2 \Rightarrow \neg($**idempotency** $\wedge$ **strict monotonicity**$)$.
- **idempotency** $\Rightarrow \oplus = \max$. Drowning effect. Local consistency.

  $A_1 \quad : \{0.6\}$ $\qquad\qquad$ $A_2 \quad : \{0.6, 0.5, 0.4, 0.4, 0.4\}$

- **strict monotony**: no drowning effect/local consistency.

Max-CSP and guaranteed optimality

- **Dynamic programming**: applies directly to **acyclic** networks $\Rightarrow$ **Polynomial class** of well-behaved (bounded induced width) graphs (Shenoy 1991; Dechter *et al.* 1990; Bistarelli *et al.* 1995).
- **Branch and bound**:
  - systematic search using a **depth first** strategy.
  - **upper bound** $\mathfrak{ub}$: valuation of the best known solution.
  - **lower bound** $\mathfrak{lb}$: on the best valuation one could obtain by extending the current assignment to a complete one.
  - **backtrack**: when $\mathfrak{lb} \succeq \mathfrak{ub}$.

Max-CSP & lower bounds

- **Backward checking** (BC): exploits constraints between assigned variables.
- **Forward checking** (FC): exploits also constraints between assigned and unassigned variables.
- Constraints among unassigned variables:

  Arc consistency !

  First trial, using the notion of relaxation:

  MAX-AC-VCSP is **NP-complete** in stricly monotonic structures (Schiex *et al.* 1995)

Max-CSP: improved lower bounds

- **Directed arc-consistency counts** (DAC) (Wallace 1994): static variable ordering. Constraints propagated in one direction fixed by the variable ordering.
- **Russian dolls** (Verfaillie *et al.* 1996): uses the optimal valuation of the problem defined by unassigned variables combined with the (BC+FC) lower bound, inductively.
- **Reversible DAC** (Larrosa *et al.* 1998; 1999): dynamic variable ordering. Each constraint is propagated in a direction that (cleverly) changes during search.
- **Weighted arc-consistency counts** (WAC) (Affane and Bennaceur 1998): each constraint is propagated partly in one direction, the rest in the other. Generalizes RDAC.

The taste of local consistency without local consistency !

Max-CSP: arc consistency (Schiex 2000)

AC can be extended to **Max-CSP** by:

- clearly **separating** information flow (constraints to domains/domains to constraints).
- preserving **equivalence** by **compensating** for induced unary constraints (using substraction).
- **loss**: uniqueness of the AC closure. Each closure induces a LB.
- this LB can be at least **twice larger** than RDAC/WAC and may be infinitely stronger.

Similar work: Penalty shifting (Koster 1999), probabilistic AC (CP'2000). Several open questions.

Frequency assignment (Cabon *et al.* 1999)

Given a set of radio **communication links**, assign to each link two frequencies such that:

- they satisfy a **fixed distance** constraint between the two frequencies of a same link

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

- theu minimize the **interferences** (weighted sum of violated constraints).

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

Optimality proof obtained by solving a sub-problem SUB_4 of CELAR6 using a parallelized improved **RDS** implementation (5.10^6 seconds of Sparc 5) (de Givry *et al.* 1997).

Other results

Using the **reversible DAC** based algorithms (Sparc 2).

	PFC-DAC		PFC-RDAC		PFC-MRDAC	
	nodes	cpu	nodes	cpu	nodes	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

Since, CELAR 6, 9 and 10 have been directly solved using **dynamic programming** and a nice tree decomposition heuristics (Koster 1999).

Main research directions

- **practical** exploitation of graph structure in **dynamic programming** and **tree search** algorithms beyond acyclic problems (treewidth,cutset number, pseudo-tree height)...

- improving **lower bounding functions** for **branch and bound methods**.

References

M. S. Affane and H. Bennaceur. A weighted arc consistency technique for Max-CSP. In *Proc. of the 13$^{\mathrm{th}}$ ECAI*, pages 209–213, Brighton, United Kingdom, 1998.

C. Bessière, E.C. Freuder, and J.C. Régin. Using inference to reduce arc-consistency computation. In *Proc. of the 14$^{\mathrm{th}}$ IJCAI*, Montréal, Canada, August 1995.

Christian Bessière. Arc-consistency in dynamic constraint satisfaction problems. In *Proc. of AAAI'91*, pages 221–226, Anaheim, CA, 1991.

S. Bistarelli, U. Montanari, and F. Rossi. Constraint solving over semirings. In *Proc. of the 14$^{\mathrm{th}}$ 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.

Rina Dechter. Enhancement schemes for constraint processing : Backjumping, learning and cutset decomposition. *Artificial Intelligence*, 41(3):273–312, 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$^{\mathrm{st}}$ European Congress on Fuzzy and Intelligent Technologies*, 1993.

H. Fargier, J. Lang, R. Martin-Clouaire, and Thomas Schiex. A constraint satisfaction framework for decision under uncertainty. In *Proc. of the 11$^{\mathrm{th}}$ Int. Conf. on Uncertainty in Artificial Intelligence*, Montréal, Canada, August 1995.

H. Fargier, J. Lang, and T. Schiex. Mixed constraint satisfaction: a framework for decision problems under incomplete knowledge. In *Proc. of AAAI'96*, Portland, OR, August 1996. AAAI Press.

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*$^{\text{th}}$ *IJCAI*, pages 278–283, Detroit, MI, 1989.

J. Gaschnig. *Performance measurement and analysis of certain Search Algorithms*. PhD thesis, Carnegie-Mellon University, 1979.

Matthew L. Ginsberg. Dynamic backtracking. *Journal of Artificial Intelligence Research*, 1:25–46, 1993.

N. Jussien, R. Debruyne, and P. Boizumault. Maintaining arc-consistency within dynamic backtracking. In *Actes de JNPC'00*, Marseille, June 2000.

Arie M.C.A Koster. *Frequency assignment: Models and Algorithms*. PhD thesis, University of Maastricht, The Netherlands, November 1999. Available at www.zib.de/koster/thesis.html.

J. Larrosa, P. Meseguer, T. Schiex, and G. Verfaillie. Reversible DAC and other improvements for solving max-CSP. In *Proc. of AAAI'98*, Madison, WI, July 1998.

J. Larrosa, P. Meseguer, and T. Schiex. Maintaining reversible DAC for Max-CSP. *Artificial Intelligence*, 107(1):149–163, January 1999.

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

Patrick Prosser. Hybrid algorithms for the Constraint Satisfaction Problem. *Computational Intelligence*, 9(3):268–299, August 1993.

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

Zsofia Ruttkay. Fuzzy constraint satisfaction. In *Proc. FUZZ-IEEE'94*, Orlando, Florida, 1994.

Daniel Sabin and Eugene C. Freuder. Contradicting conventional wisdom in constraint satisfaction. In *Proceedings of the 2nd International Workshop on Principles and Practice of Constraint Programming*, number 874 in LNCS, Rosario, Orcas Island (WA), May 1994. Springer-Verlag.

Thomas Schiex and Gérard Verfaillie. Nogood recording for static and dynamic CSP. In *Proceeding of the 5$^{\mathrm{th}}$ IEEE International Conference on Tools with Artificial Intelligence (ICTAI'93)*, pages 48–55, Boston, MA, November 1993.

Thomas Schiex and Gérard Verfaillie. Stubborness: an enhancement scheme for backjumping and nogood recording. In *Proc. of the 12$^{\mathrm{th}}$ ECAI*, pages 165–169, Amsterdam, Netherlands, August 1994.

T. Schiex, H. Fargier, and G. Verfaillie. Valued constraint satisfaction problems: hard and easy problems. In *Proc. of the 14$^{\mathrm{th}}$ IJCAI*, pages 631–637, Montréal, Canada, August 1995.

Thomas Schiex, Jean-Charles. Régin, Chistine Gaspin, and Gérard Verfaillie. Lazy arc consistency. In *Proc. of AAAI'96*, Portland, OR, August 1996. AAAI Press.

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

T. Schiex. Arc cohérence pour contraintes molles. In *Actes de JNPC'00*, Marseille, June 2000.

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$^{\text{th}}$ biennal conf. of the canadian society for comput. studies of intelligence*, pages 227–230, May 1990.

Pascal van Hentenryck. Incremental constraint satisfaction in logic programming.

In *ICLP'90*, pages 189–202, 1990.

G. Verfaillie and T. Schiex. Solution reuse in dynamic constraint satisfaction problems. In *Proc. of AAAI'94*, pages 307–312, Seattle, WA, August 1994.

Gerard Verfaillie and Thomas Schiex. Dynamic backtracking for dynamic CSPs. In *Proceedings ECAI'94 Workshop on Constraint Satisfaction Issues raised by Practical Applications*, Amsterdam, August 1994.

G. Verfaillie, M. Lemaître, and T. Schiex. Russian doll search. In *Proc. of AAAI'96*, pages 181–187, Portland, OR, 1996.

R. Wallace. Directed arc consistency preprocessing. In M. Meyer, editor, *Selected papers from the ECAI-94 Workshop on Constraint Processing*, number 923 in LNCS, pages 121–137. Springer, Berlin, 1994.

M. Zweben and M. Eskey. Constraint satisfaction with delayed evaluation. In *Proc. of the 11*$^{\text{th}}$ *IJCAI*, pages 875–880, Detroit, MI, 1989.