

Conclusion et Perspectives

Cours de Programmation par Contraintes

Ecole Jeunes Chercheurs en Programmation

Rencontre du 25 mai 2003

François Fages
Projet Contraintes
INRIA Rocquencourt

<http://contraintes.inria.fr>

1. Problèmes de satisfaction de contraintes CSP

Algorithmme = Logique + Contrôle [Kowalski 72]

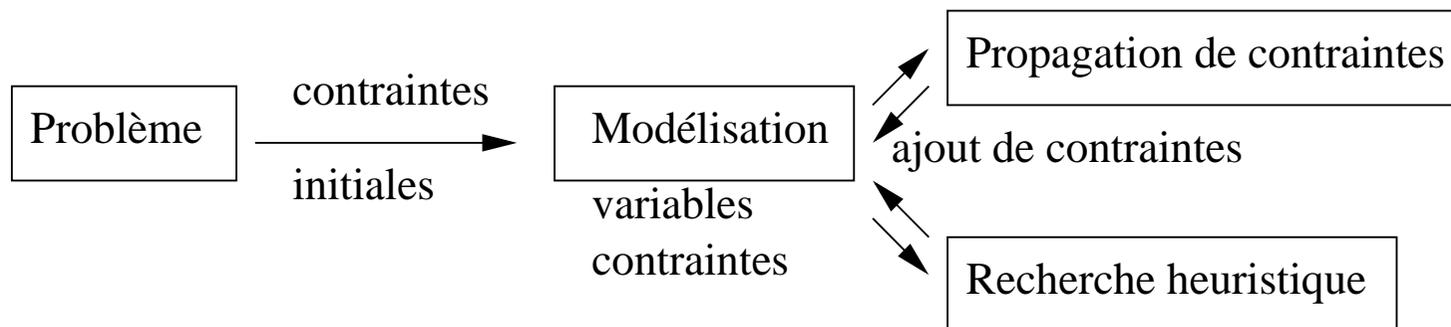
Variables $\{x_1, \dots, x_v\}$ à domaine fini de valeurs $D = \{e_1, \dots, e_d\}$.

Contraintes unaires $x \in \{e_i, e_j, e_k\}$, binaires $x \geq y$, globales n-aires

1. Modélisation du problème par des variables et des contraintes,
2. Résolution par:

Propagation de contraintes: ajout dynamique de contraintes inférées

Recherche heuristique avec backtracking: ajout non-déterministe de contraintes (décisions)



2. Programmes Logiques avec Contraintes PLC

PLC = CSP + définitions inductives de relations (Prolog)

- **Contraintes de base** sur domaine de termes H , entiers bornés FD , réels R , booléens B , ontologies H_{\leq} , etc.

- **Relations définies** *en extension* par faits contraints:

`precedence(X,D,Y) :- X+D<Y.`

`disjonctives(X,D,Y,E) :- X+D<Y.`

`disjonctives(X,D,Y,E) :- Y+E<X.`

et en intention par règles:

`labeling([]).`

`labeling([X|L]):- indomain(X), labeling(L).`

- **Programmation des procédures de recherche et heuristiques:**

Parallélisme “et” (choix variable): heuristique “first-fail” domaine min

Parallélisme “ou” (choix valeur): heuristique “best-first” valeur min

Théorie de la programmation logique avec contraintes

Programme = Formule logique (contraintes, clauses)

Execution = Recherche de preuves (satisfiabilité contraintes, résolution)

Classe de langages PLC(\mathcal{X}) [Jaffar Lassez 87].

- Sémantique opérationnelle: $c_0|B_0 \rightarrow_P \dots c_{n-1}|B_{n-1} \rightarrow_P c_n|\square$,
- Sémantique logique: $P, th(\mathcal{X}) \models c_n \Rightarrow c_0 \wedge B_0$,
- Sémantique de point fixe: $c_n|B_0 \in \text{lfp}(T_P)$.

Exemples d'applications de la sémantique de point fixe:

- Analyse de programme par *interprétation abstraite*,
- *Model checking avec contraintes* de systèmes de transition concurrents à états infinis $EF(\phi) = \text{lfp}(T_{P+\phi})$, $EG(\phi) = \text{gfp}(T_{P\wedge\phi})$.

3. Programmes Concurrents avec Contraintes CC

$$\text{CC} = \text{PLC} + \text{ask} \text{ [Saraswat 89]}$$

“ask”: ($c \rightarrow A$) suspension de l'agent A tant que la contrainte c n'est pas *impliquée* par la mémoire de contraintes.

Concurrence=Communication (variables partagées)+Synchronisation (ask)

- Expression des **propagateurs de contraintes** comme des agents CC
[Van Hentenryck et al. 91, Codognet-Diaz 98]
- Expression de **modélisations multiples** et résolution concurrente.
- Retardement de la pose des contraintes sur gros problèmes
($10^4, 10^5$ contraintes)

Théorie des langages CC

- Sémantique opérationnelle: $c_0|B_0 \rightarrow_P \dots c_{n-1}|B_{n-1} \rightarrow_P c_n|B_n\dots,$
- Sémantique de point fixe: contraintes et agents comme *opérateurs de fermeture* dans *treillis des contraintes*
 - extensif: $f(x) \leq x$
 - monotone: si $x \leq y$ alors $f(x) \leq f(y)$
 - idempotent: $f(x) = f(f(x))$.
- Sémantique logique: dans la Logique Linéaire de Jean-Yves Girard
[Fages Ruet Soliman 2001]

4. Contraintes et changements d'états dans LCC

LCC = CC + consommation de contraintes (changement d'états)

- Systèmes de contraintes basés sur la logique linéaire

$A \Rightarrow B = !A \multimap B$ [Girard 1987]

L'opération Ask consomme les contraintes

- Modélisation des variables impératives (backtrackables) par contrainte

token linéaire `val(N,V)`

```
read(N,V):- val(N,V).
```

```
write(N,V):- ask(val(N,_), val(N,V)).
```

- **LCC**: langage noyau d'une nouvelle implantation modulaire d'un langage `CC(H,FD,R,...)`+changement d'états

Technologie de compilation en code natif de GNU-Prolog.

5. Panorama de quelques outils existants

- Outils basés sur Prolog:

Cosytec CHIP, GNU-Prolog CLP(H,FD,B),

Sicstus-Prolog CC(H,FD,R,Q,B), Prolog IV, Eclipse, CHR, ...

(fin 2003: LCC: noyau+LCC(H,FD,B,R))

- Bibliothèques C++:

Ilog-Solver, Ilog-Scheduler, Cosytec CHIP++, Choco, ...

- Bibliothèques Java:

J-Solver, J-Configurator, CHR JCK toolkit, Koalog, ...

- Autres:

Claire, Eclair, ..., FaCiLe CAML, ...

Sujets de recherches d'actualité

- Benchmarks, problèmes ouverts (ex. “open shop” 6×6 !)
- **Contraintes globales** (algorithmes de Recherche Opérationnelle, graphes,...),
- Synthèse automatique de solveurs de contraintes (définies en extension),
- **Hybridation d'algorithmes** de PPC et Recherche Locale, PPC et MIP
- Coopération de solveurs,
- Élimination de **symétries**,
- Langages de modélisation, vérifications statiques, typage
- Combiner contraintes et changements d'états,
- “Model checking” avec contraintes
- **Applications nouvelles** en bioinformatique, multimedia, Web, business