

**Domain Decomposition  
for Parallel Resolution  
of Constraint Satisfaction Problems  
with OpenMP**

---

Daniel Singer, Zineb Habbas, Michaël Krajecki\*

LITA - Université de Metz / \* LERI - Université de Reims

**FWOMP'2000**

## Talk overview

- **Introduction** : What is a CSP ?
- **Sequential Resolution** : BT, FC, MRV.
- **Parallel Resolution** : with Domain Decomposition.
- **Experimentations** : first results with OpenMP.
- **Conclusion** : it's only a beginning...

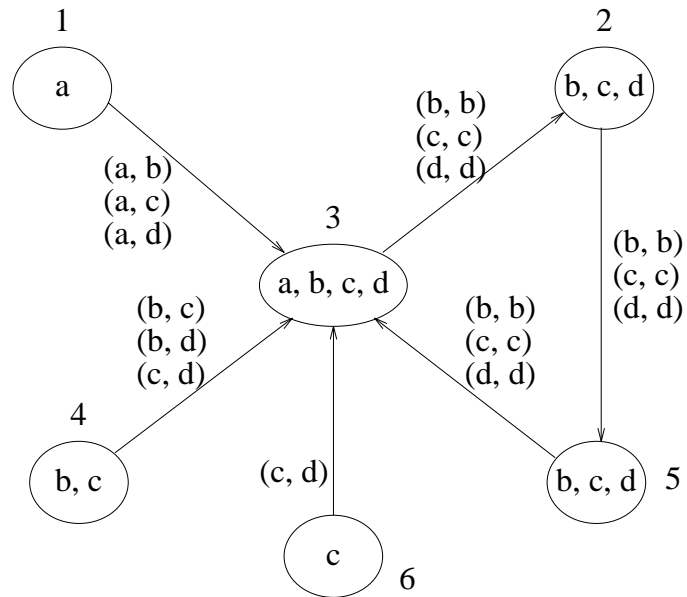
## CSP Definition

A **Constraint Satisfaction Problem (CSP)**:

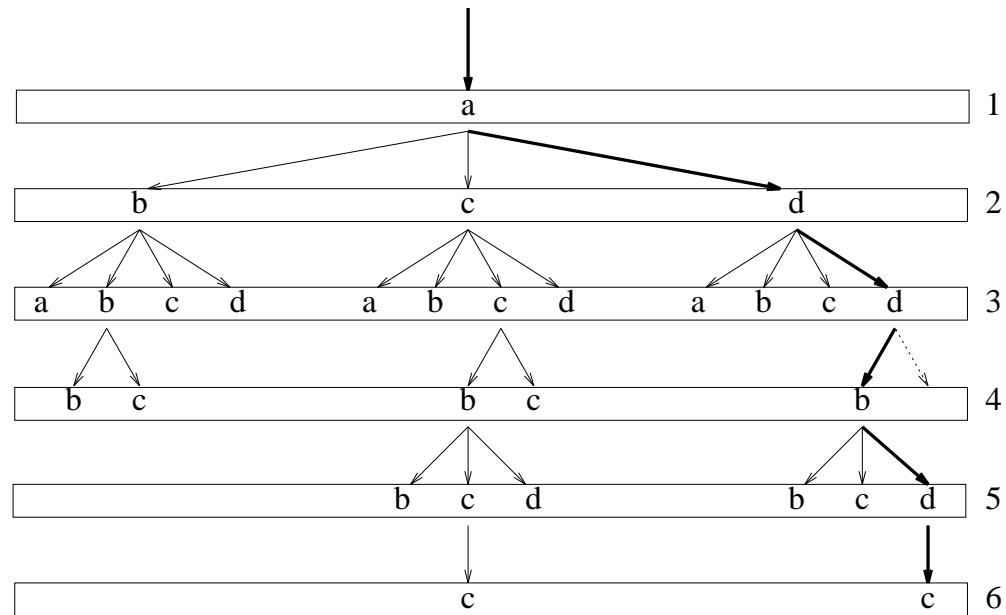
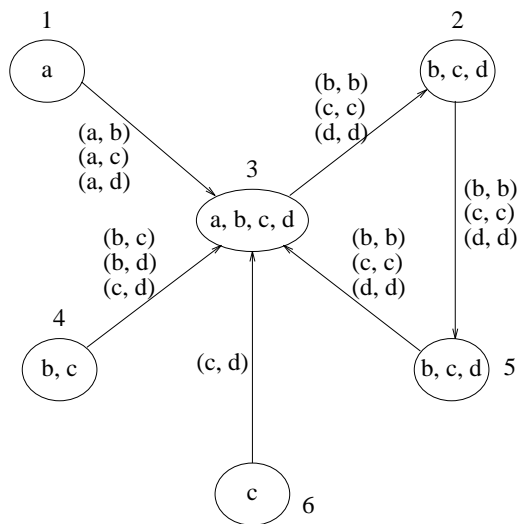
- $\mathcal{P} = (X, D, C, R)$  :
- $X = \{X_1, \dots, X_n\}$  is a set of  $n$  *variables*;
- $D = \{D_1, \dots, D_n\}$  is a set of  $n$  *domains* with at most  $d$  values;
- $C = \{C_1, \dots, C_m\}$  is a set of  $m$  *constraints* defined as variables subsets  $\{X_{i_1}, \dots, X_{i_{n_i}}\} \subseteq X$ ;
- $R = \{R_1, \dots, R_m\}$  is a set of  $m$  *relations* defined as  $n_i$ -tuples on  $D_{i_1} \times \dots \times D_{i_{n_i}}$  compatible with  $C_i$ .

# Example

---



# Backtracking: the search tree



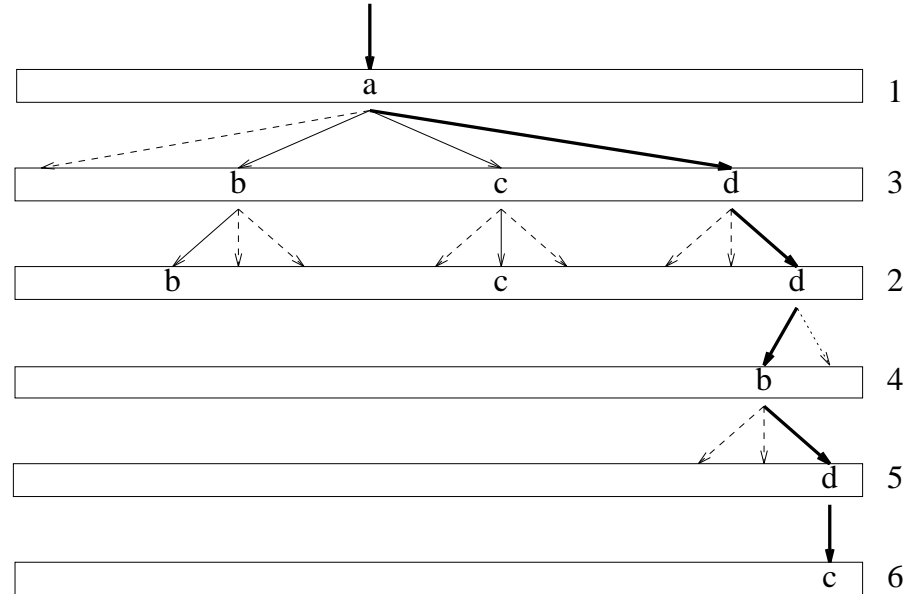
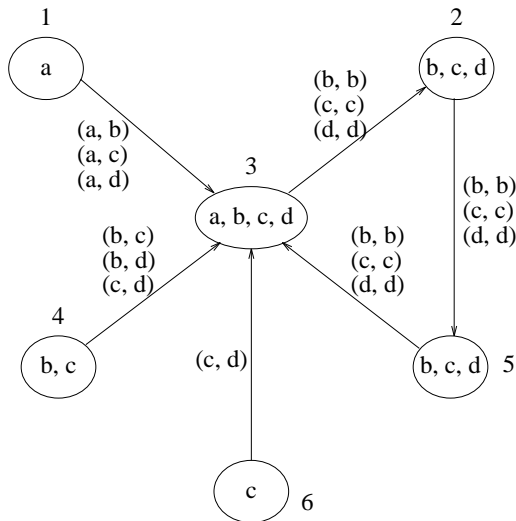
## Backtracking improvements

---

- **Filtering by Forward Checking: FC**  
removes values of domains which are detected locally inconsistent before or while searching .
- **« Intelligent » Backtrack : MRV**  
Heuristics for orders such as **Dynamic Variables Ordering**.
- **Decomposition :**  
replaces if possible one CSP by some easier subproblems.

# Forward-Checking

- **Principle** : partial filtering with **Arc Consistency** property when a variable is instantiated and restoring when Backtracking;
- **Example** :



## Decomposition Methods

---

→ **Structural** : (Freuder's Theorem 82)

"Cycle Cutset" or "Tree Clustering" methods.

**main idea** : use of relations between the variables;

→ **Domains Decomposition** : (Jégou 93)

**main idea** : use of relations between the values of the domains;

– **Decomposition** ( $\mathcal{P}$ ) :  $\mathcal{P}_1, \dots, \mathcal{P}_s$

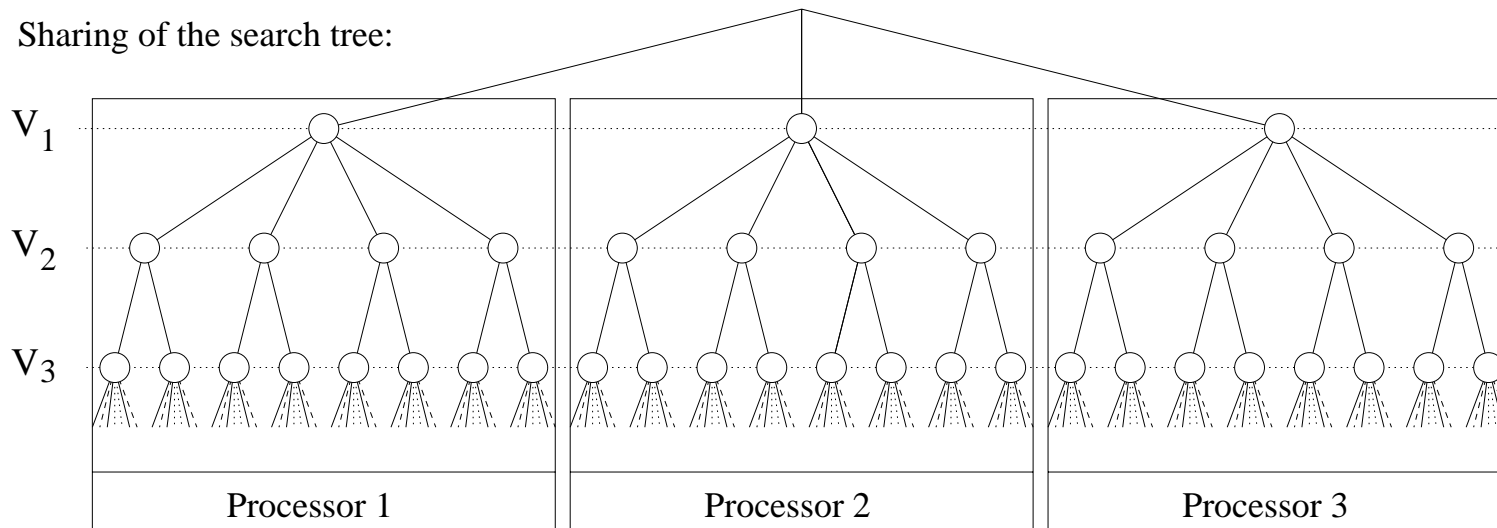
– **Micro-Structure** ( $\mathcal{P}$ ) :  $\mu(\mathcal{P})$

– **Triangulation** ( $\mu(\mathcal{P})$ ) :  $\Gamma(\mu(\mathcal{P})), \sigma$

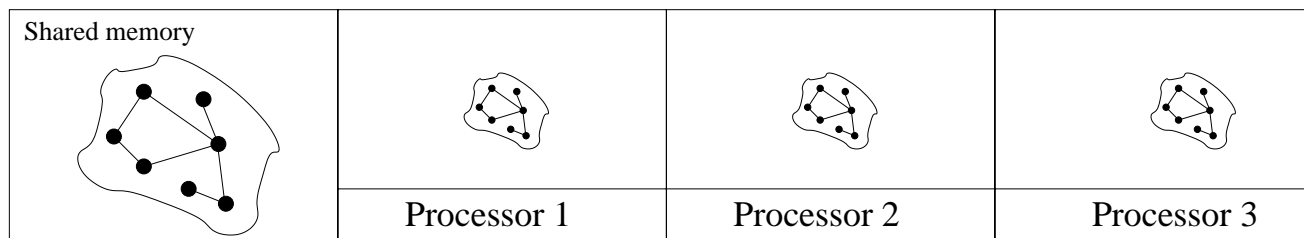
– **Maximal Cliques** ( $\Gamma(\mu(\mathcal{P}), \sigma)$ ) :  $Y_1, \dots, Y_k$

# Parallel Resolution of CSP: two approaches

Sharing of the search tree:



Domain decomposition:



## Parallel Resolution with Domain Decomposition

---

- **For  $i = 1$  to  $\delta$  Do // Solve( $\mathcal{P}_i$ )**
  - one parallel task = one subproblem
  - one sub-CSP = same structure, but on restricted domains
- ⊕ **independant tasks**
- ⊖ **each sub-CSP  $\Rightarrow$  Irregular search tree.**

## Experimentations

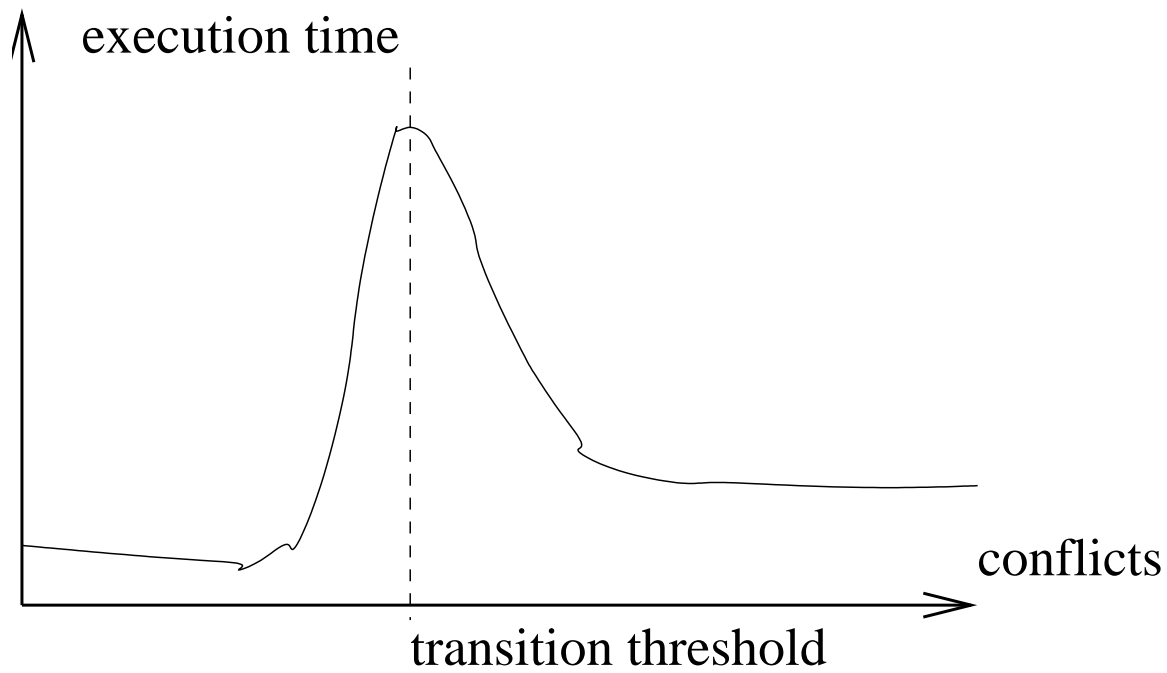
---

(Preliminary study of the method Feasibility)

- **Parallel Machine** : SGI Cray Origin'2000
- **Development** : C++ with OpenMP
- **Search Algorithms** : FC, MRV
- **Random Generation model** :  $\langle n, d, m, p \rangle$ 
  - problems benchmarks for "Sparse Graphs" (Frost & Dechter):
  - 50 variables, varying  $d$  from 25 to 40, 123 constraints.
  - **Phase Transition Phenomenon** : some problems are harder than others.

# Phase Transition Phenomenon

---



## The benefits and drawbacks of OpenMP

---

- **Benefits**
  - ⊕ **easy to use: parallel prototype programming**
  - ⊕ **relatively good portability**
  - ⊕ **efficient for regular applications**
- **Drawbacks**
  - ⊖ **Irregular applications  $\Rightarrow$  poor Dynamic Memory allocation**
  - ⊖ **Hard to know the real place(processor) where the data are stocked**

## Remarks:

---

- Different behaviours of Decomposition method for Consistent and for non-Consistent problems.
- The number of subproblems generated is proportional to the Domains size.

## First results:

---

- **for non-consistent problems**
  - Decomposition method alone gives good results.
  - very poor parallel efficiency.
  - ? : all the sub-problems are easy to find non-consistent?
- **for consistent problems**
  - Decomposition method alone gives not good results.
  - Superlinear speedup.
  - ? : the first sub-problem which gives a solution stops the search!

Conclusion: it's only a beginning...

---

- **other Experimentations** :
  - higher domains size
  - real problems
- **other Resolution Algorithms** :
  - other Filtering : MAC
  - Parallel resolution inside each subproblem
- **other Decomposition methods** :
  - *Tree clustering*.
- **Comparisons with pure Message Passing methods** :