Title: A Disjunctive Decomposition Schema for Finite Binary CSPs Using Complete No-Good Sets

Authors: B. Y. Choueiry & G. Noubir

Proc.: FLAIRS'97 (TR KSL-98-23)

Year: 1997

Theory and Practice of Constraint Satisfaction

CSCE 990-05, Fall 1999

www.cse.unl.edu/~choueiry/CSE990-05/

Motivation:

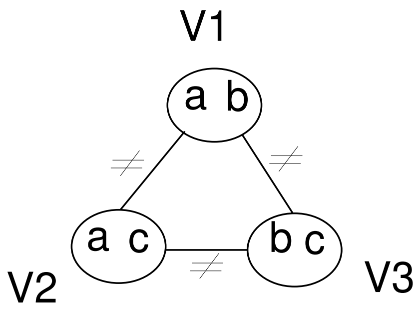
- Decomposition is useful for $\begin{cases} \text{reducing complexity} \\ \text{parallel processing} \end{cases}$
- Generalize intuitions behind the VAD

Outline:

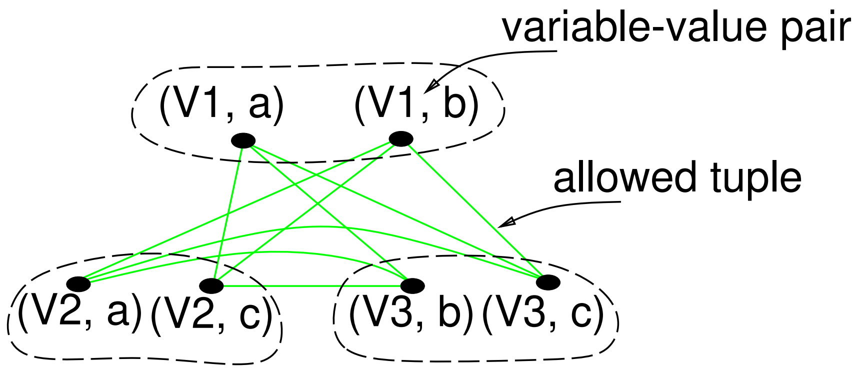
- Basic mechanism
- Properties
- Real-world application
- Comparison with [Jégou '93] (more in the paper)
- Conclusions

Representation of a CSP

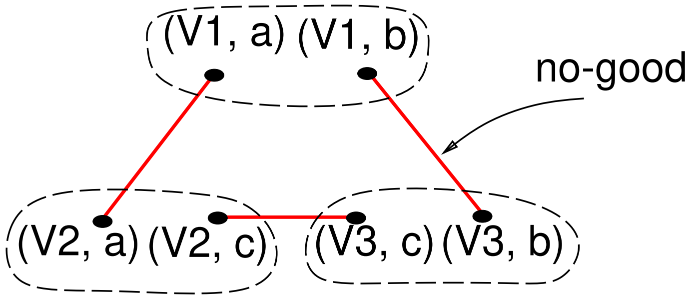
Constraint graph $\mathcal{G}(\mathcal{P})$:



Graph of the microstructure $\mu(\mathcal{P})$:



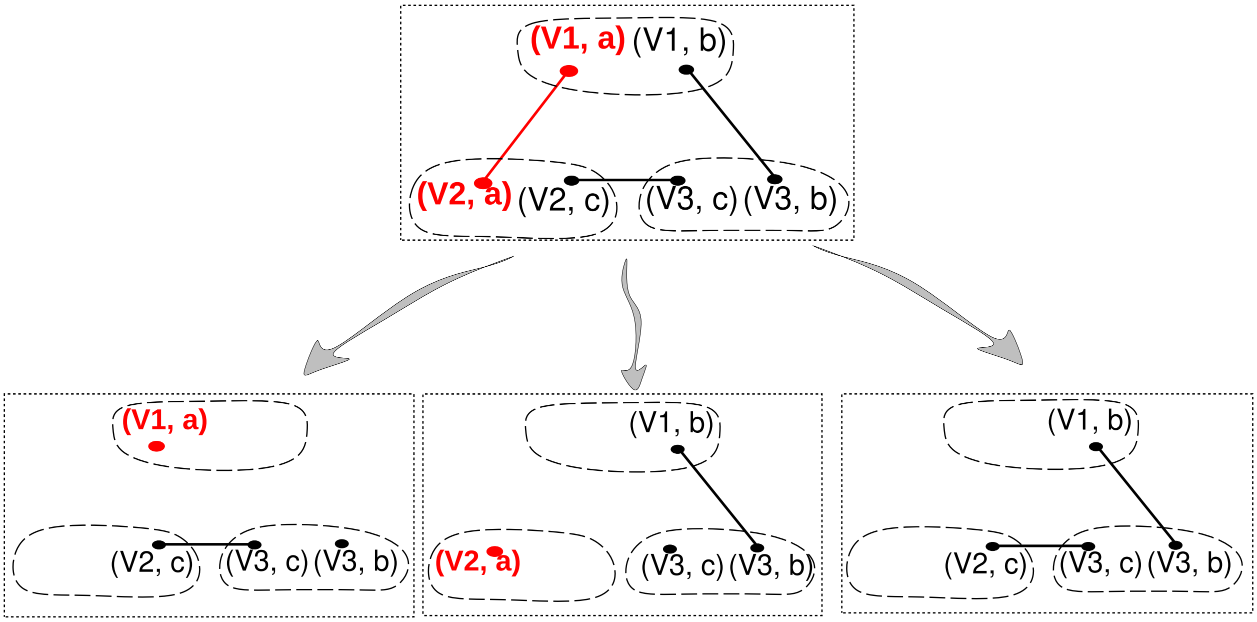
Graph of the co-microstructure co-$\mu(\mathcal{P})$:



Key idea

No solution to the CSP can simultaneously contain 2 variable-value pairs (VVP) of a no-good

Example:

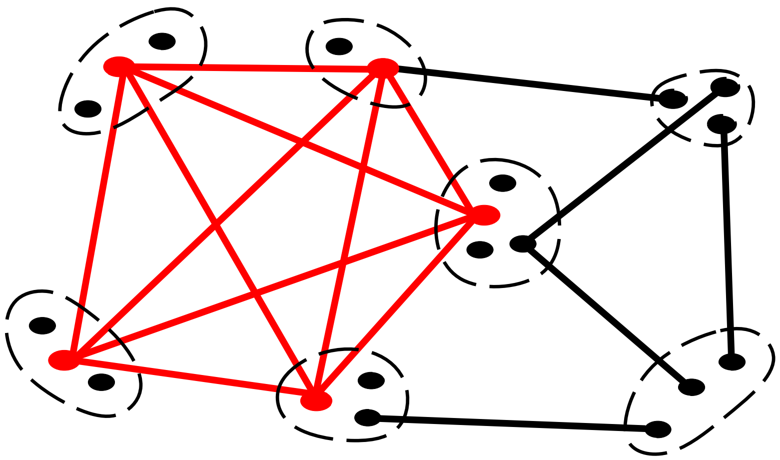


In any solution, we may have either $\begin{cases} (V_1, \text{a}) \\ (V_2, \text{a}) \\ \text{none} \end{cases}$

Complete No-Good Set (CNG)

A clique in co-$\mu(\mathcal{P})$ $\rightarrow$ complete no-good set (CNG)

Clique size 5



Decomposition into:

$\longrightarrow$ 5 subproblems, one per vv-pair of CNG

$\longrightarrow$ +1 subproblem containing no vv-pair of CNG

Decomposition based on co-$\mu(\mathcal{P})$

Principle

Given $\mathcal{P} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$:

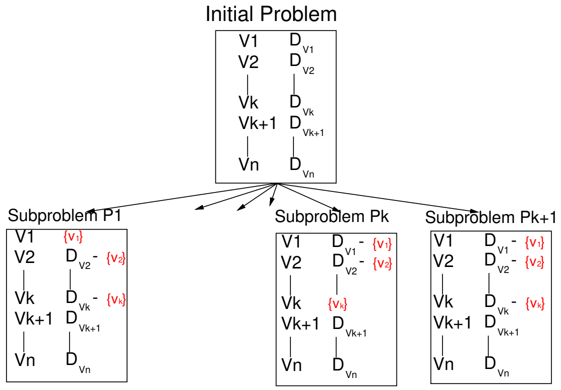
- Identify a clique S of size k in co-$\mu(\mathcal{P})$,
- Apply co-μBD(S).
  Generate $(k+1)$ subproblems:
  $\rightarrow$ k subproblems, each with one vv-pair of S
  $\rightarrow$ $+1$ subproblem with no vv-pair of S

Estimates

If $\quad |\mathcal{V}| = n \qquad$ and $\qquad \forall\ V_i \in \mathcal{V},\ |\mathcal{D}_{V_i}| = d$,

$$\text{then} \begin{cases} \text{Size}(\mathcal{P}) = d^n \\ \sum_{i=1}^{k+1} \text{Size}(\mathcal{P}_i) = k(d-1)^{k-1}d^{n-k} + (d-1)^k d^{n-k} \\ \text{Gain}(\text{co}-\mu\text{BD}(S)) = \sum_{i=2}^{k} \binom{k}{i}(d-1)^{k-i}d^{n-k} \end{cases}$$

Properties of co-μBD



co-μBD **partitions** the solution space of $\mathcal{P}$

This decomposition strategy is:

disjunctive	$\longrightarrow$ Solving any $\mathcal{P}_i$ gives a solution to $\mathcal{P}$
consistent	$\longrightarrow$ Sol($\mathcal{P}_i$) is consistent for $\mathcal{P}$
simplifying	$\longrightarrow$ Size($\mathcal{P}_i$) $<$ Size($\mathcal{P}$)
complete	$\longrightarrow$ No solution is lost
non-redundant	$\longrightarrow$ No solution replicated in $\{\mathcal{P}_i\}$
reducible	$\longrightarrow$ $\sum_i$ Size($\mathcal{P}_i$) may be $<$ Size($\mathcal{P}$)

Discussion

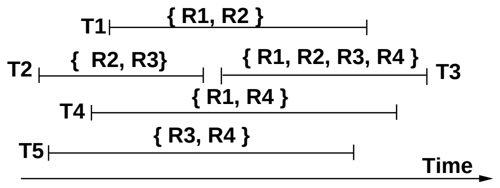
$$\text{Gain}(\text{co}-\mu\text{BD}(S)) = \sum_{i=2}^{k} \binom{k}{i} (d-1)^{k-i} d^{n-k}$$

Gain grows with the size of the clique S (CNG-set)

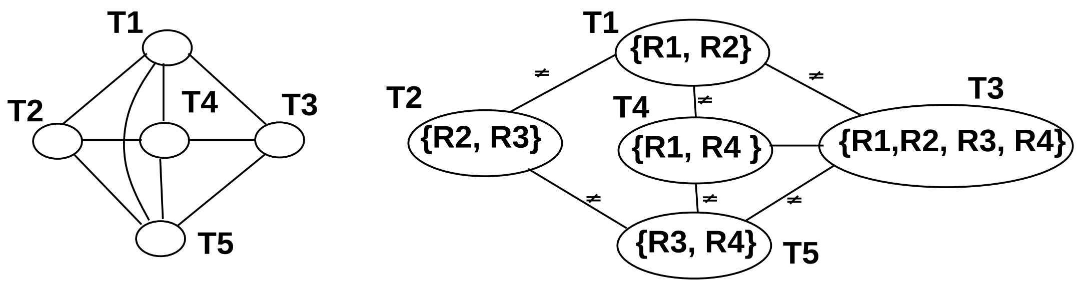
Finding maximal-cliques is NP-hard... but

- Cliques **need not** be maximal:
  co-μBD works with cliques of any size
- Finding a clique of a **fixed** size k is polynomial
- Gain is exactly determined. It is controllable:
  more time/computing power, better gain
- Controllable # of subproblems: parallelism!
- When co-$\mu(\mathcal{P})$ is **triangulated**, maximal-cliques can be found in linear time
  $\longrightarrow$ real-world application: resource allocation

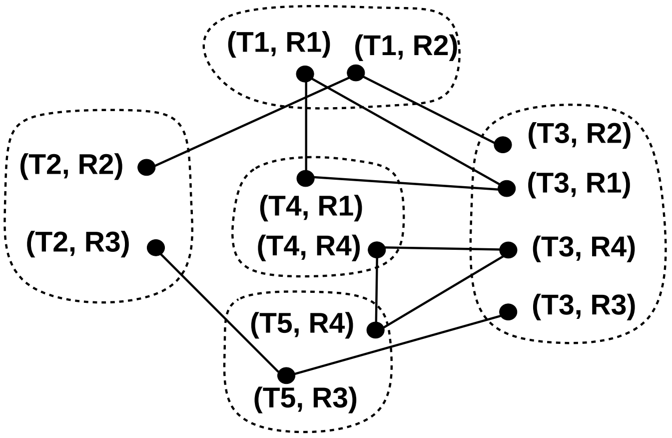
Resource Allocation



RA is NP-complete [Arkin & Silvenberg 87]



IG is triangulated; CG is not triangulated



co-μ(RA) is necessarily triangulated

$\longrightarrow$ Maximal cliques can be found in linear time

$\longrightarrow$ co-μBD can be efficiently applied to RA

A disjunctive decomposition schema

- Basic mechanism
- Properties
- Application to RA

Other decomposition techniques

- Disjunctive: μBD [Jégou 93]
- More in the paper Conjunctive:
  Value Assignment Delay Heuristic (VAD) [Choueiry]

  Disjunctive:

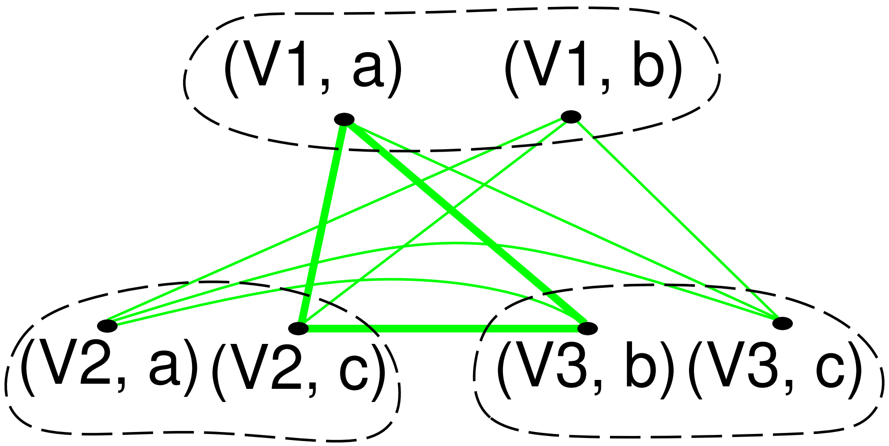
  Inferred Disjunctive Constraint (IDC) [Freuder 93]

  Factor Out Failure (FOF) [Freuder 95]

Decomposition based on $\mu(\mathcal{P})$

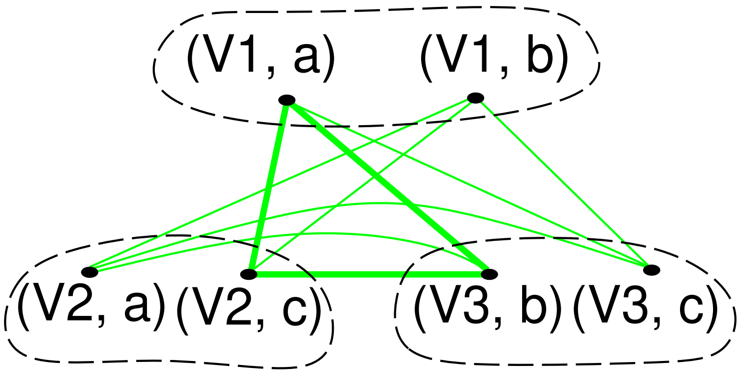
[Jégou 93]

Given $\mathcal{P} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ with $|\mathcal{V}| = n$



Key idea: <u>any</u> clique of size n in $\mu(\mathcal{P})$ is a solution to $\mathcal{P}$

Principle of μBD



1. Triangulate $\mu(\mathcal{P})$
   $\rightarrow$ triangulation introduces new edges (fill-in)
   $\rightarrow$ new edges correspond to inconsistent tuples
2. Find max-cliques in triangulated $\mu(\mathcal{P})$
   $\rightarrow$ linear time complexity
   $\rightarrow$ max-cliques may contain inconsistent tuples
3. Max-cliques compared against original CSP
   $\rightarrow$ induces subproblems of reduced domains
   $\rightarrow$ disjunctive decomposition

μBD

$$\left\{\begin{array}{l}\text{loses no solution (complete)}\\ \text{BUT}\\ \text{may replicate solutions in subproblems (redundant)}\end{array}\right.$$

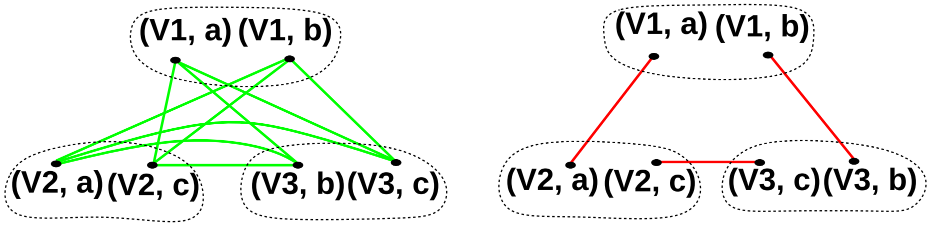
Comparing co-μBD to μBD

Criterion	**μBD**	**co-μBD**
Decomposition	'side-effect'	goal
Redundant?	yes	no
Max-cliques	required	no
# of $\mathcal{P}_i$	unpredictable	$k+1$, for a clique size k
Size of $\mathcal{P}_i$	unpredictable	exactly determined
Gain	unpredictable	exactly determined

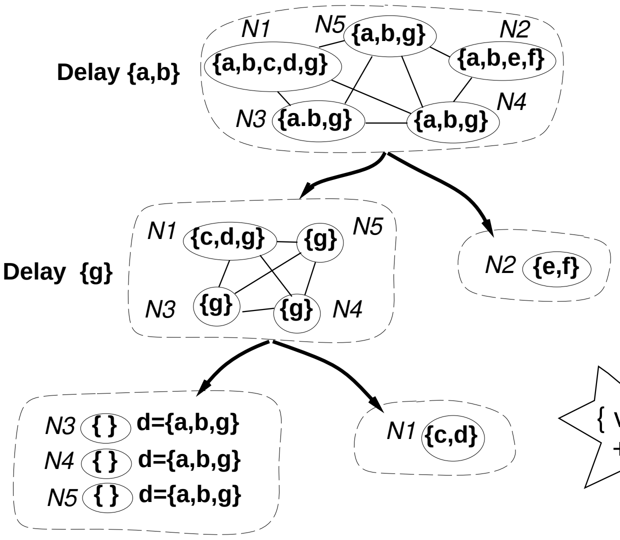
Open question

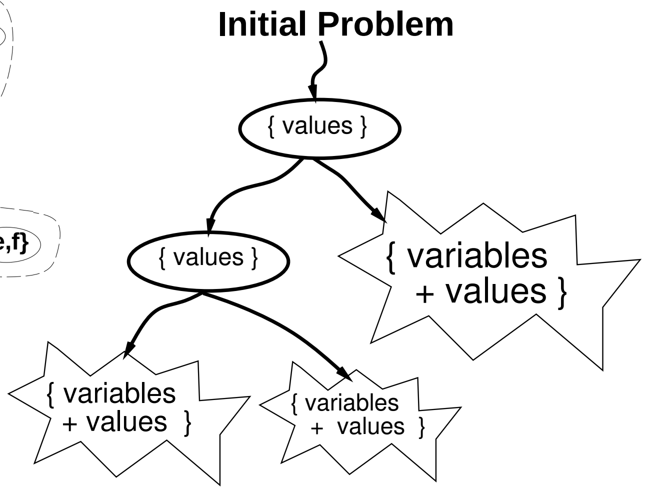
Tight $\mathcal{P}$, few allowed tuples, few edges in $\mu(\mathcal{P}) \rightarrow$ Use μBD ?

Loose $\mathcal{P}$, few no-goods, few edges in co-$\mu(\mathcal{P}) \rightarrow$ Use co-μBD ?



Comparing co-μBD to VAD

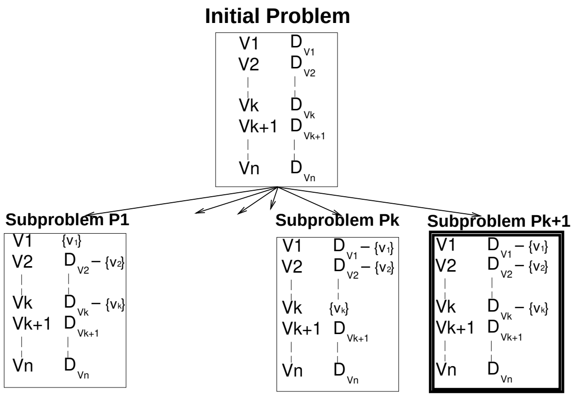




$\rightarrow$ identify most contented values (heuristic)

$\rightarrow$ hide these values & iterate (VAD)

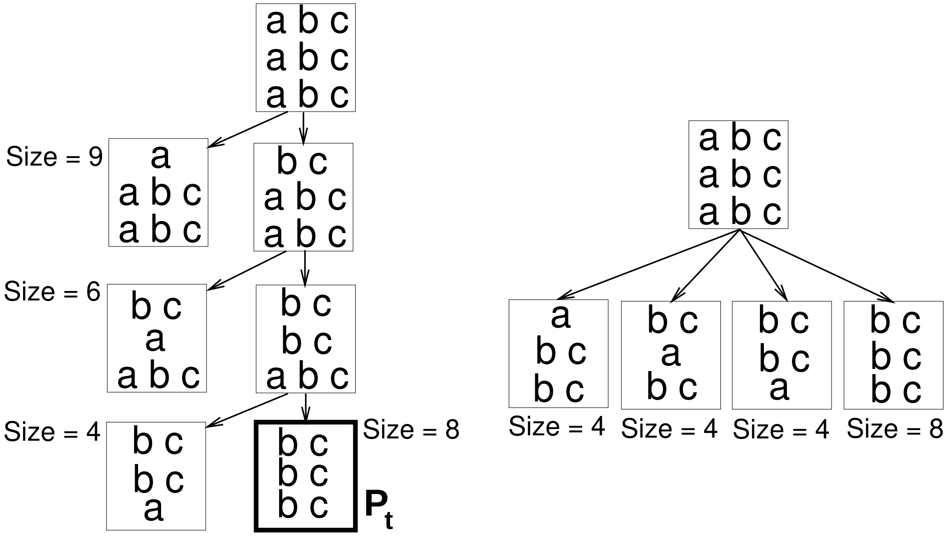
- VAD conjunctive, co-μBD disjunctive
- RA: most contented res. are exactly CNG-sets
- VAD explores $\mathcal{P}_{k+1}$



co-μBD vs. subproblem extraction (SE)

[Freuder & Hubbe 95]

Extracts a 'target' subproblem $\mathcal{P}_t$ from the initial CSP



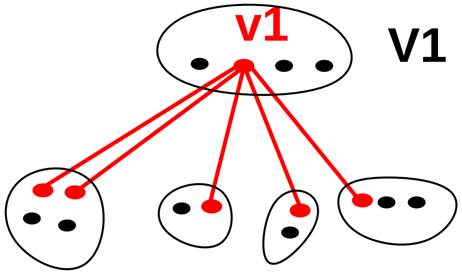
When the values 'outside' $\mathcal{P}_t$ belong to a CNG, co-μBD

generates $\left\{\begin{array}{c}\text{smaller}\\ \text{fewer}\end{array}\right\}$ subproblems than SE

With CNG-sets, co-μBD is more simplifying than SE

Inferred Disjunctive Constraint (IDC) [Freuder 93]

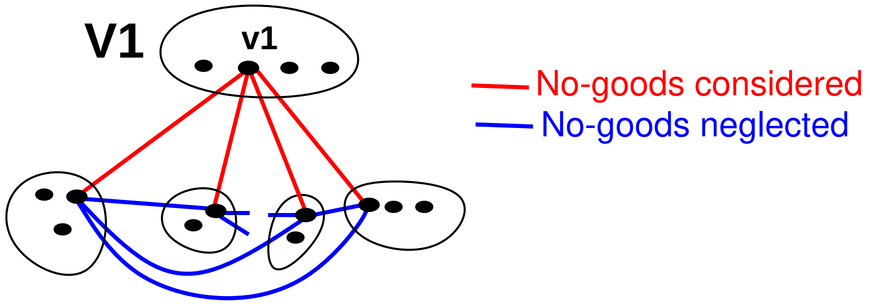
Given v a value for a variable, generate subproblems that contain either v or the values that are not consistent with v



IDC addresses minimal no-good sets but it treats all MNG of 1 vv-pair at once

IDC is not complete (some sol. are lost)
IDC is semi-complete (keeps at least 1 sol.)

Comparing co-μBD to IDC



- IDC exploits MNG-sets; co-μBD exploits CNG-sets
- IDC uses simultaneously all MNG-sets of a VVP
  co-μBD uses one CNG-sets of a VVP at any step
- Many small NG per VVP $\rightarrow$ IDC more effective?
  Few large NGs per VVP $\rightarrow$ co-μBD more effective?
- IDC: no concern about cliques
- IDC sacrifices completeness (misses solutions)

Conclusions

- co-μBD flexible wrt. the choice of Size(CNG)
  $\rightarrow$ control of #subproblems
  $\rightarrow$ control of gain vs. effort
- Nice properties & quantifiable gain
- co-μBD is efficiently applicable to RA

Future Research

- $\text{Use of no-good sets for} \begin{cases} \text{problem decomposition} \\ \text{localization of interactions} \\ \text{identification of conflicts} \end{cases}$
  (as an application of the VAD to general CSPs)
- $\text{Experimental studies} \begin{cases} \text{real} \\ \text{random problems} \end{cases}$