Title:	Tree Clustering for Constraint Networks
Authors:	R. Dechter & J. Pearl
Journal:	AIJ
Vol.:	38
Pages:	353-366
Year:	1989

Details:	Tsang, Section7.6, pages 212-234.

Theory and Practice of Constraint Satisfaction
CSCE 990-05, Fall 1999
www.cse.unl.edu/~choueiry/CSE990-05/

Contributions:

- Tree Clustering (T-C) scheme $\surd$
- Adaptive Consistency (A-C) $\times$
- Relating T-C & A-C $\times$

Two basic mechanisms for solving CSPs:

1. Local consistency algorithms (local constraint propagation)
2. Backtrack search (answer producing)

Using the same set of constraints for a long stream of queries (find many solutions)?
$\longrightarrow$ invest more effort and space in (1) to facilitate (2)

Main idea:

Restructure the CSP as a tree: group variables in clusters such that interactions between clusters are tree-structured

Tree?

$\rightarrow$ Local consistency ensures global consistency

$\rightarrow$ Backtrack-free search

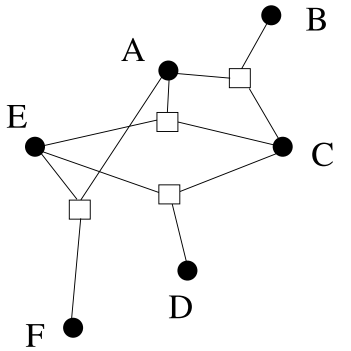
Steps?

- Decide which variables to cluster
- Find internally consistent values in each cluster
- Process clusters as singleton variables in a tree

$\longrightarrow$ general and systematic method

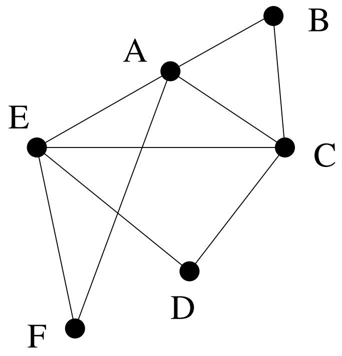
Non-binary CSPs

Constraint Hyper-Graph



variables
values
non-binary constraints

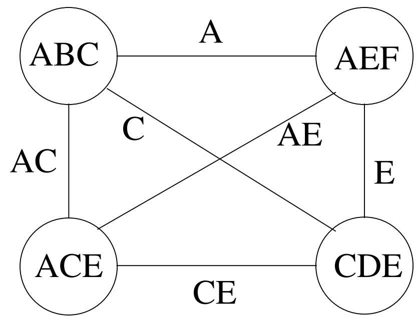
Primal Constraint Graph



variables?
values?
binary constraints: projection

NON-CONSERVATIVE

Dual Constraint Graph

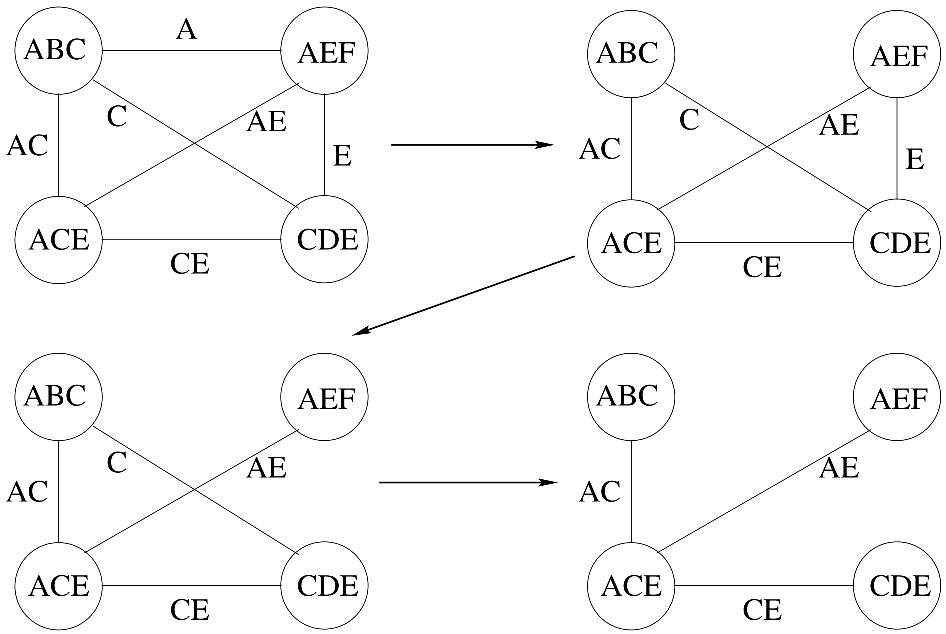


variables?
values?
binary constraints?

If dual graph is a tree, ...

Dual graph: Equality constraints → redundant → remove

⟶ **Join Graph:** for any two nodes that share a variable there is at least one path of labeled arcs, each containing the shared variable



⟶ **Join Tree**

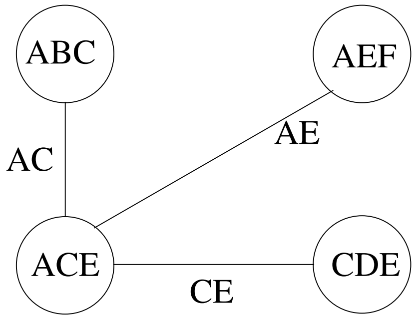
⟶ A CSP that has a join tree is called an **acyclic CSP**

⟶ Efficient procedures exist for finding a join tree in acyclic CSPs

A CSP is acyclic iff its primal graph is chordal and conformal [Bieri et al. 83]

- chordal: any cycle of length at least 4 has a chord
- conformal: each maximal clique corresponds to a constraint in the original CSP

Given a join tree:



Global consistency in $\mathcal{O}(pl^2)$, where:

- p is #constraints (c-variables, nodes in the join tree)
- l maximum #subtuples per constraint

If tuples are ordered lexicographically, matching 2 tuples: $\mathcal{O}(\log l)$
Global consistency in $\mathcal{O}(p.l.\log l)$

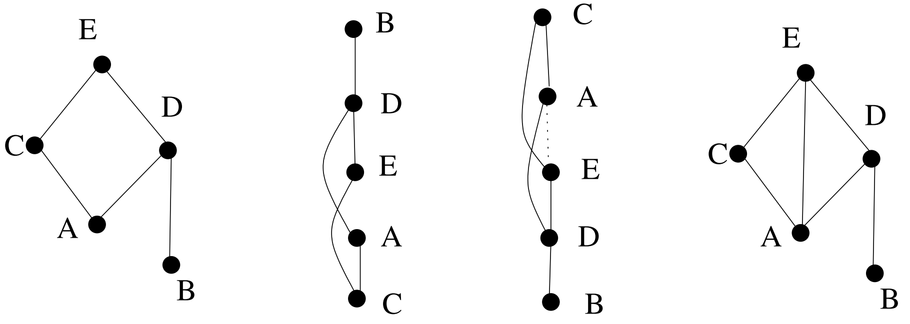
$\longrightarrow$ What if a join tree does not exist?

Triangulate primal graph

- Triangulate the primal graph (*e.g.*, Tarjan):

  1- Compute an ordering using a maximum cardinality ordering

  2- Reverse, moralize the graph

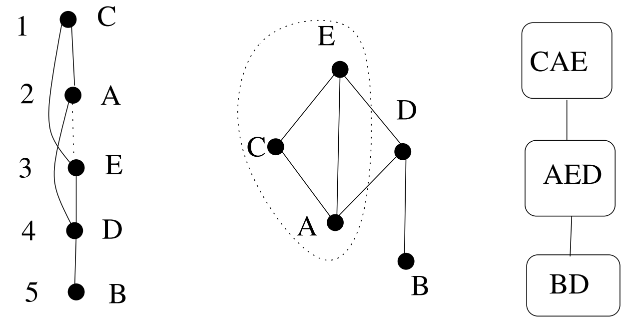


- Identify maximal cliques in linear time

If primal graph is chordal, no fill-in edge is added

Clustering Scheme

- Triangulate the primal graph
- Find maximal cliques. Index them by the rank of their lowestnodes



- Form the dual graph, and identify one of its join tree (if necessary) by connecting each cluster to ancestor with whom it shares the largest set of variables
- Solve subproblems in each cluster
- Solve tree treating clusters as variables (DAC on the join tree, solve backtrack-free)

Conclusions

Tree clustering: systematic way of regrouping constraints into hierarchical structures capable of supporting information retrieval without backtracking.

$\rightarrow$ Effective database, to be amortized over many problem instances

$\rightarrow$ Resistant to local changes (changes in a given clique)