Title: A Sufficient Condition for Backtrack-Free Search

Author: E.C. Freuder

Journal: Journal of the ACM

Year: 1982

Foundations of Constraint Satisfaction

CSCE421/821, Fall 2001

`www.cse.unl.edu/~choueiry/F01-421-821/`

Berthe Y. Choueiry (Shu-we-ri)
Ferguson Hall, Room 104
`choueiry@cse.unl.edu`, Tel: (402)472-5444

- **Motivation:** Backtrack effort so far evaluated by experiments
- **Goal:** Develop an analytical understanding of backtrack effort
- **Assumptions:** binary finite CSPs
- Backtrack depends
  1. on order of exploration in tree
  2. interactions between variables, through constraints
- To understand backtracking, analyze relationship between structure and orderings
- **Result:** *Relation (width, consistency level)* to predict backtrack effort in search
- **Significance:** theoretical, does not carry out in practice simply provides guidance in practice

Known facts

- Search involves two types of orderings:
  1. Vertical: order of choosing variables
  2. Horizontal: order of choosing values
- The constraint graph defines a **connective structure** structure of connections among variables

Definitions

- Backtrack-free: once an instantiation is made that is consistent with the past, it will not be undone
- An ordering is backtrack-free if it guarantees a backtrack-free search regardless of the horizontal order in search

Example

- $X = \{5, 2, 4, 6\}$
- $Y = \{2, 4, 6, 10\}$
- $Z = \{5, 2, 4, 6\}$
- Z divides X
- Z divides Y

Try search order:

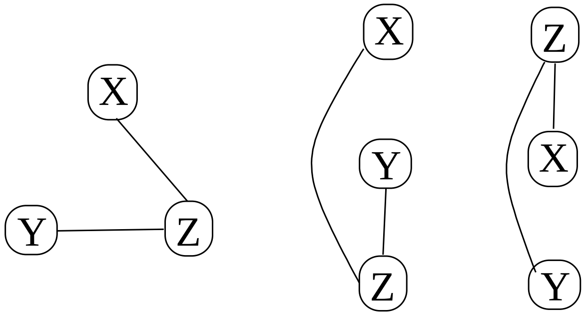
- Variables: X, Y, Z
- Values: as listed

Try search order:

- Variables: Z, X, Y
- Values: as listed

Reminders:

- Ordering of nodes (for instantiation in search) $n!$
- Width of a node in an ordering
- Width of an ordering
- Width of a constraint graph

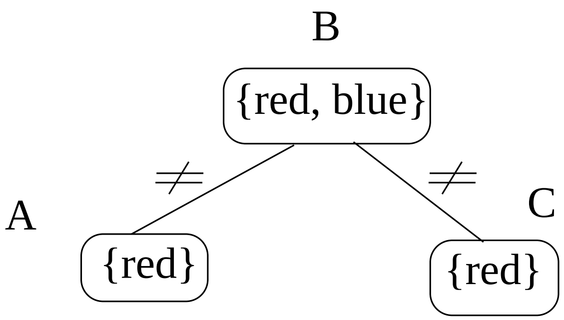


- k-consistency: any consistent assignment for $(k-1)$-variables can be extended to any k^{th} variable
- Step k of Constraint Synthesis Algorithm guarantees j-consistency for all $j \leq k$
  $\longrightarrow$ **Strong k-consistency**

Warning: k-consistency vs. strong k-consistency

Definitions

- k-consistency: any consistent assignment for $(k-1)$-variable can be extended to any k^{th} variable
- strong k-consistency: has to be j-consistent for all $j \leq k$



Is this CSP 2-consistent?

Is it CSP 3-consistent?

Theorem I

Given a constraint satisfaction problem:

1. A vertical search order is backtrack-free if the level of **strong** consistency is greater than the width of the corresponding ordering constraint graph

2. There exists a backtrack-free vertical search order of the problem is the level of strong consistency is greater than the width of the constraint graph.

Proof

- When instantiating variable V_k of width j, we must check consistency with at most j other variables (back-checking)

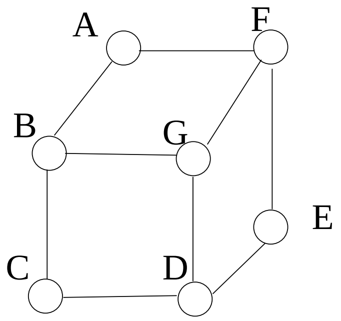
- If CSP is strong k-consistent, with $k \geq j + 1$, then all j prior choices will succeed, and no backtracking will ever occur.

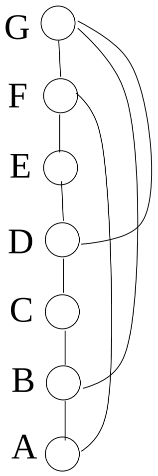
Result

- Compute width (remember??)
- ensure consistency level k greater than width value
- backtrack-free search guaranteed!

Minimal width?

Consistency level needed?





Width of a tree?? *(necessary + sufficient)*

Consistency level needed?

State result: ....

Freuder'82

Procedure Width of a graph G:

Remove from the graph all nodes not connected to any others

Set $k \leftarrow 0$

Do while there are nodes left in the graph:

    Set $k \leftarrow (k + 1)$

    Do while there are nodes not connected to more than k others:

        Remove such nodes from the graph, along
        with any edges connected to them.

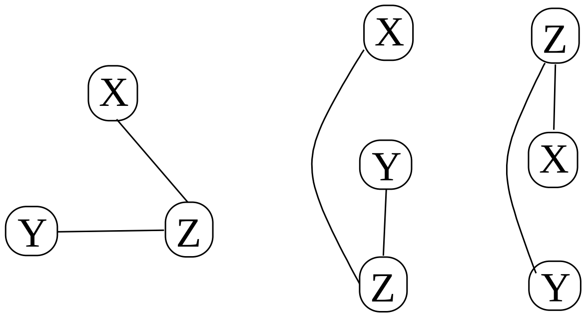
Return k

end

The minimal width ordering of the nodes is obtained by taking the nodes in the reverse order than they were removed.

Example I

- X = $\{5, 2, 4, 6\}$
- Y = $\{2, 4, 6, 10\}$
- Z = $\{5, 2, 4, 6\}$
- Z divides X
- Z divides Y



Example II: graph theory

A simple planar constraint graph has width ≤ 5

- Any simple planar graph has a vertex of degree ≤ 5 (known property)
- Use this vertex as last node in order ordering, remove it
- the remaining graph must have a has a vertex of degree ≤ 5
- repeat... etc.
- ordering of width ≤ 5

Graph coloring require strong-consistency is greater than or equal to #colors
$\longrightarrow$ Given 6 colors, we can color a graph backtrack-free
$\longrightarrow$ Graph theory: 4 colors are sufficient :—)

Conclusions

- Main result: *Relation (width, consistency level)* to guarantee backtrack-free search
- k-consistency can be quite expensive
- Also, while ensuring k-consistency we may alter graph width (*e.g.*, path consistency adds edges)
  this alters width of graph
  higher-order consistency become necessary, etc..
- Limited use in practice
- Nevertheless, important paper:
  identifies special cases of graph <u>topologies</u> that can be solved in a tractable way

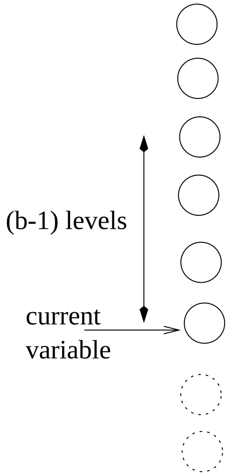
Title: A Sufficient Condition for Backtrack-Bounded Search
Author: E.C. Freuder
Journal: Journal of the ACM
Year: 1985

Foundations of Constraint Satisfaction
CSCE421/821, Fall 2001
`www.cse.unl.edu/~choueiry/F01-421-821/`

Berthe Y. Choueiry (Shu-we-ri)
Ferguson Hall, Room 104
`choueiry@cse.unl.edu`, Tel: (402)472-5444

Context

- Backtrack-free can be difficult to realize
- Generalize previous result to predict <u>bounds</u> on backtrack effort
- Definition: backtrack search is b-level bounded, if at any level of searching, we backtrack to at most $(b-1)$ previous levels



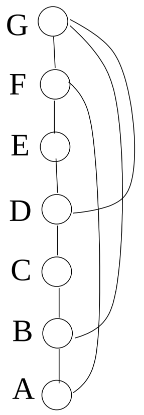
Backtrack-free, $b =$.....?

- Goal: relation between width and consistency to bound backtrack

Definitions: j-width

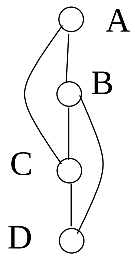
Generalize width $\longrightarrow$ j-width

- Width of j consecutive vertices in an ordering is the number of nodes preceding the group in the ordering and linked to any of the j vertices in the group



  width of a node corresponds to j = ...?

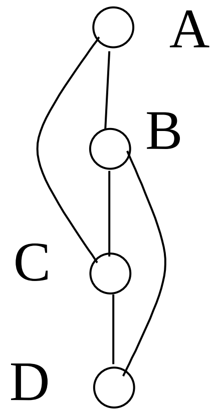
- j-width of a node v, is the minimum, for k=1 to j, of the width of the k consecutive nodes up to a including v



  3-width at D = ...
  3-width at C = ...

Definitions: j-width

- j-width of a node v, is the minimum, for k=1 to j, of the width of the k consecutive nodes up to a including v



- j-width of an ordering is maximum j-width at all nodes in the ordering

- j-width of a graph is the minimum j-width over all orderings

Definitions: (i, j)-consistency

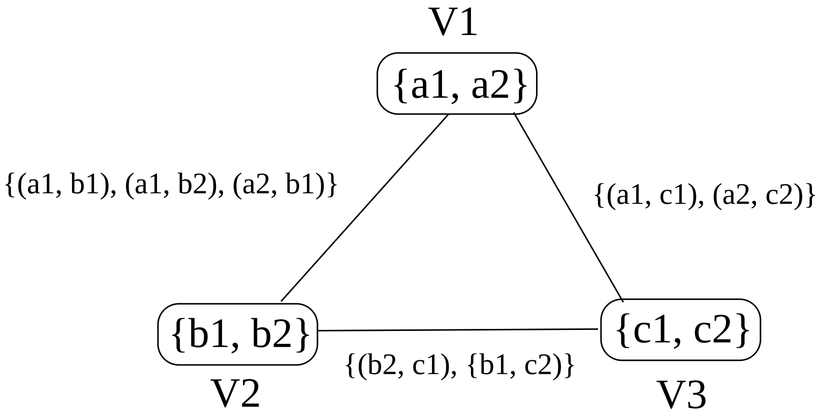
Generalize k-consistency $\longrightarrow$ (i, j)-consistency

- k-consistency: any consistent assignment for $(k-1)$-variable can be extended to any k^{th} variable
- (i, j)-consistency: given consistent values for i variables and given any other (or fewer) j variables, it is possible to find values for the j variables such that the values for i and j variables are consistent.
  k-consistency $= (\ldots, \ldots)$-consistency?
- strong (i, j)-consistency is (i', j)-consistency for all $i' \leq i$

Definitions: (i, j)-consistency

Note that (i, j)-consistency neither implies nor is implied by (i', j')-consistency where $i + j = i' + j'$

This is (1,2) consistent but not (2,1)-consistent



(given V1=a1 and V2=b1, there is no consistent values for V3)

Theorem I

Given a constraint graph for a CSP, there exists a vertical search order that guarantees j-bounded backtrack search if the graph is strongly (i, j)-consistent for i equal to the j-width of the graph

Use in 'practice'

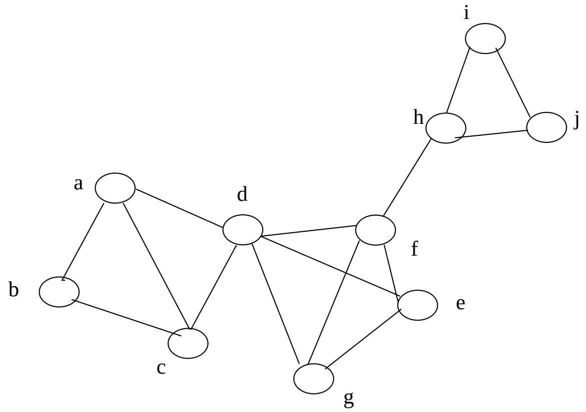
- Choose j (bound on backtrack)
- compute j-width (*e.g.*, by branch-and-bound) $\rightarrow i$
- ensure strong (i, j)-consistency (*e.g.*, by constraint synthesis)
- backtrack is guaranteed j-bounded!!

Same drawback: consistency alters width...

Definitions:

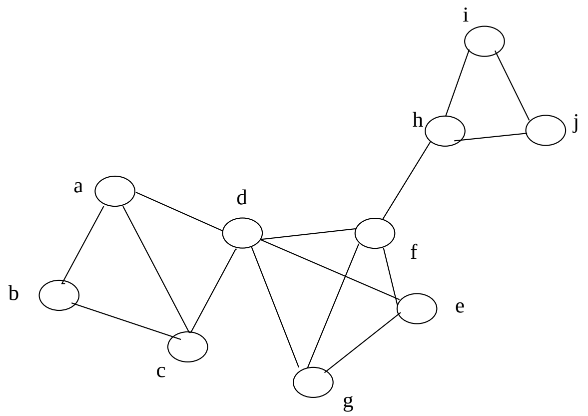
Given a connected graph:

- Articulation node: a node whose removal disconnects the graph
- Separable graph: a graph that has one or more articulation nodes
- Biconnected component: a graph that has no articulation node (remove any node and its connected edges, the graph remains connected).



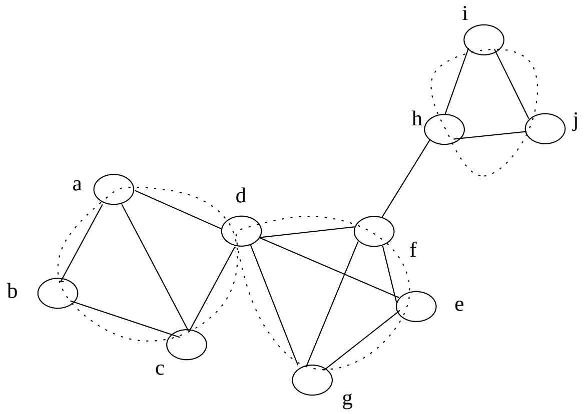
Theorem III (we skipped Theorem II)

A CSP can be solved in time exponential in b, where b is the number of nodes in the maximal biconnected component of the constraint graph.



Proof for theorem is in the paper.
Requires finding a search order of nodes whose $(b-1)$-width is equal to 1, which is done as follows:

- Find articulation nodes and biconnected components $(O(V+E))$



- Construct a tree whose nodes are biconnected components $(O(V^2))$

  Choose a component as root of tree, put in Q
  While Q is not empty
      Remove a component c from Q
      For each component d which shares an articulation point with c and is not in the tree:
          Make d a child of c in the tree
          Put d in Q

Conclusions

- Generalized strong k-consistency to strong (i, j)-consistency
- Generalized width to j-width
- Provided algorithms for computing them (terrible complexity)
- Provided relations between them to guarantee j-bounded search
- $\rightarrow$ limited practical use
- Very important result: a CSP can be searched in an effort bounded by the size of its largest bi-connected component (regardless of the number of nodes in the CSP)