Title: Search orders in CSPs

Author: Edward Tsang

Book: Foundations of Constraint Satisfaction

Chapter: 6

Foundations of Constraint Satisfaction
CSCE 990-005, Fall 2000

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

Motivation for ordering heuristics

- In lookahead:
  - failure could be detected earlier under some orderings than others
  - larger portions of the solution space can be pruned off under some orderings than others

- In BT, less of undoing-labels needs to be done under some orderings than others

- When searching for one solution, value ordering may speed up search as branches that have a better chance to reach a solution are explored first

Heuristics

Value ordering:

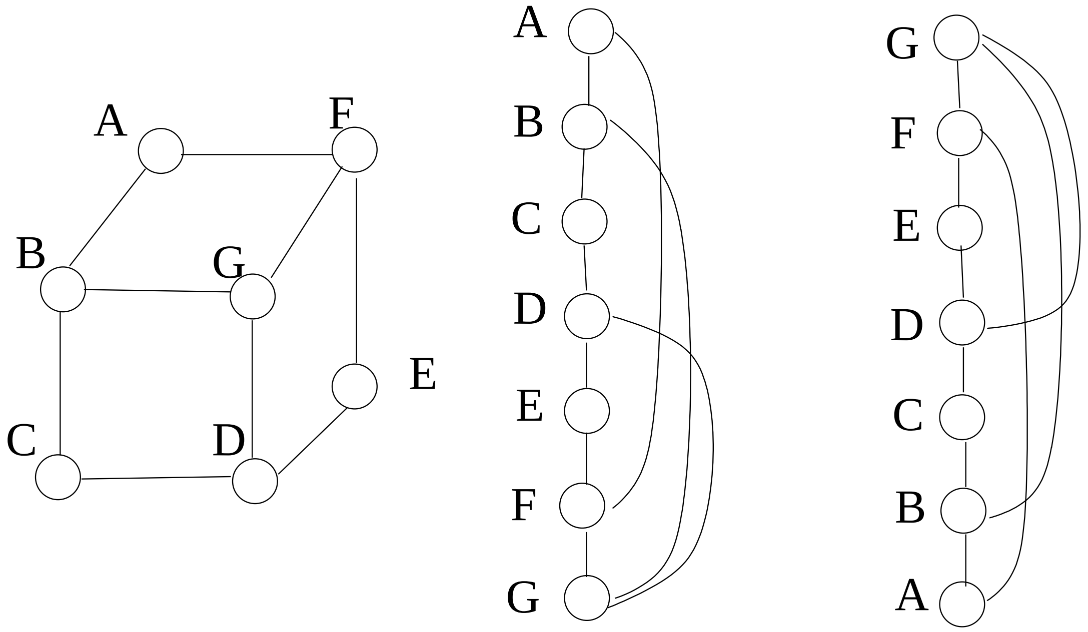
Min-conflict heuristic
orders values according to the conflicts in which they are involved with the future variables.

Variable ordering

- Minimal width ordering (MWO)
- Minimal bandwidth ordering (MBO)
- Fail first principal (FFP)
- Maximal cardinality ordering (MCO)

→ static, dynamic?

Minimal width ordering



Reduces the chance of backtracking: variables that have more unassigned values depending on them are labeled first. Variables at the front of the ordering are in general more constrained..

Finding minimum width ordering: $\mathcal{O}(n^2)$.

Minimal bandwidth ordering

Localizes/confines backtracking:
the smaller the bandwidth is, the sooner one could backtrack to relevant decisions

Finding minimum bandwidth ordering:
$\rightarrow$ NP-hard.

Is there an ordering of a *given* bandwidth k?
$\rightarrow \mathcal{O}(n^{k+1})$ (polynomial)

Fail First Principal

Aims at recognizing dead-ends as soon as possible so that search effort can be saved.

Several FFP measures
→ *e.g.* domain size (smallest domain first)

Simple, cheap and effective

Suitable for both static and dynamic ordering

Maximum Cardinality Ordering

An approximation of min. width ordering, $\mathcal{O}(n)$

Choose a node arbitrarily. Among the remaining nodes, choose the one that is connected to the maximum number of already chosen nodes, break ties arbitrarily, repeat...
Reverse the final order.

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.