Constraint Satisfaction Problems

CHAPTER 6
COSC 370
SPRING 2013
ALAN C. JAMIESON

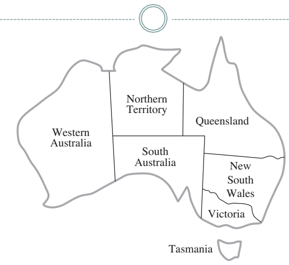
SOME SLIDE CONTENT FROM RUSSELL & NORVIG PROVIDED SLIDES

- CSP
- Backtracking Search
- Problem Structure and Decomposition
- Local Search

CSP

- Class of problems that feature three distinct components:
  - X – a set of variable symbols.
  - D – a set of domains for each variable.
  - C – a set of constraints that specify allowable combinations of values.
- Our goal – create an assignment of variables such that it does not violate any of the constraints.
- Frequently - multiple complete assignments.
- Occasionally, we need to look at complete assignments and evaluate them.

Example: Map Coloring



Variables: {WA, NA, Q, SA, NSW, V, T}
Domains: D_i = {red, green, blue}
Constraints: adjacent regions must have different assignment (WA != SA)

Example Job-Shop Scheduling

- Consider a car assembly factory, part of which has 15 tasks: install axles (front and back), affix all four wheels (right and left, front and back), tighten nuts for each wheel, affix hubcaps, and inspect the assembly.
- Our axles have to be in place before the wheels are put on, which takes 10 minutes each. Affixing each wheel takes 1 minute, which must happen before tightening the nuts (2 minutes), and finally we can attach the hubcap (1 minute).
- Suppose we also add that everything must be done in 30 minutes, and that the Inspection happens after everything, but takes 3 minutes. We also have 4 people to assign to these jobs at particular minute long timeslots. How do we formulate?

Example: Job-Shop Scheduling

- Variables: {AxleF, AxleB, WheelRF, WheelRB, WheelLF, WheelLB, NutsRF, NutsRB, NutsLF, NutsLB, CapRF, CapLF, CapRB, CapLB, Inspect}
- Domain: D_i = {1, 2, 3, … , 27}
- Constraints: AxleF + 10 <= WheelRF
  AxleF + 10 <= WheelLF
  AxleB + 10 <= WheelRB
  AxleB + 10 <= WheelRF
  WheelRF + 1 <= NutsRF
  WheelLF + 1 <= NutsLF
  WheelRB + 1 <= NutsRB

… and so on.

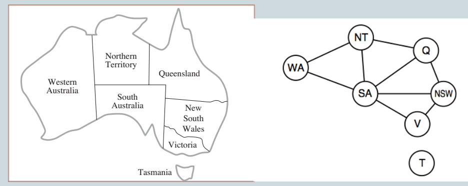
Problem Formulation Exercise

The Whitt Window Company is a company with only three employees which makes two different kinds of hand-crafted windows: a wood-framed and an aluminum-framed window. They earn $180 profit for each wood-framed window and $90 profit for each aluminum-framed window. Doug makes the wood frames, and can make 6 per day. Linda makes the aluminum frames, and can make 4 per day. Bob forms and cuts the glass, and can make 48 square feed of glass per day. Each wood-framed window uses 6 square feet of glass, and each aluminum-framed window uses 8 square feet of glass. Maximize the total profit.

Credit: Hillier and Lieberman. "Introduction to Operations Research, 9th Edition".

Constraint Graphs

- Graph that shows relationships between variables.
  - Vertices – variables
  - Edges - constraints



Varieties of CSPs

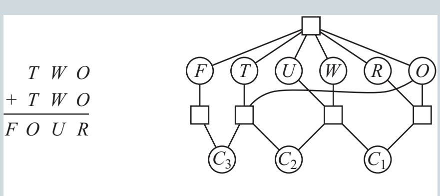
- Discrete variables –
  - finite domains like Boolean CSPs (SAT)
  - infinite domains (ints, strings, etc.) like job scheduling – need a constraint language (see our job scheduling example from before). Linear constraints solvable, nonlinear undecidable.
- Continuous variables – like start and end times with positioning for a telescope, solvable by LP methods.

Varieties of Constraints

- Unary constraints – single variable, single constant: e.g. SA != green
- Binary constraints – pairs of variables: e.g. SA != WA
- Higher order constraints – three or more variables.
- Preference constraints (soft constraints): e.g. red is better than green, representable by a cost

Example – Cryptarithmetic

- Each character in an equation represents a distinct digit. The aim – find a substitution for letters such that the resulting sum is correct.



Real World Examples

- Assignment problems
- General timetabling problems
- Hardware configuration
- Transportation scheduling
- Factory scheduling
- Floorplanning

Standard Search Formulation

- States defined by values assigned as the algorithm proceeds.
- Initial state – empty assignment
- Successor function – assign a value to an unassigned variable that does not conflict with current assignment.
- Goal test – the current assignment is complete.
- Problem? SIZE!

Backtracking Search

- Begin to consider changes to single nodes at a time.
- Depth first search with single value assignments = backtracking.
- The core, basic uninformed algorithm for CSPs.
- Can solve n-queens for n approx. 25.

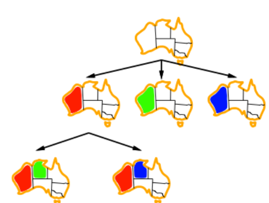
Backtracking Search

```
function BACKTRACKING-SEARCH(csp) returns solution/failure
   return RECURSIVE-BACKTRACKING({ }, csp)
function RECURSIVE-BACKTRACKING(assignment, csp) returns soln/failure
   if assignment is complete then return assignment
   var ← SELECT-UNASSIGNED-VARIABLE(VARIABLES[csp], assignment, csp)
   for each value in ORDER-DOMAIN-VALUES(var, assignment, csp) do
       if value is consistent with assignment given CONSTRAINTS[csp] then
           add {var = value} to assignment
           result ← RECURSIVE-BACKTRACKING(assignment, csp)
           if result ≠ failure then return result
           remove {var = value} from assignment
   return failure
```

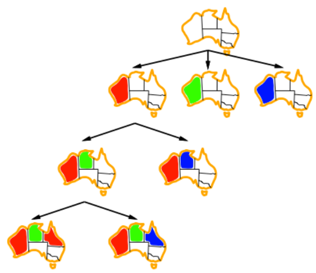
Example

Example

Example



Example

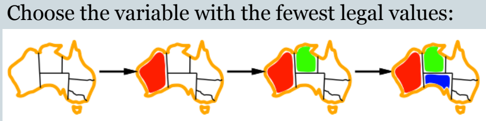


Improving Efficiency

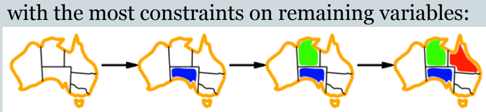
- Which variables should be assigned next?
- In what order should its values be tried?
- Can we detect inevitable failure early?
- Can we take advantage of problem structure?

Minimum-Remaining-Values (MRV)

- Choose the variable with the fewest legal values:

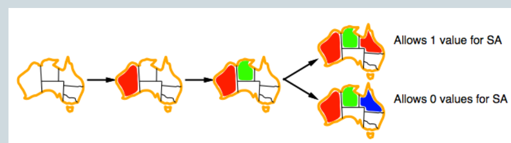


- Tiebreaker: degree heuristic – choose the variable with the most constraints on remaining variables:



Least-Constraining-Value (LCV)

- Given a variable, choose the least constraining value: the one that rules out the fewest values in the remaining variables:



Problem Structure

- Look for subproblems – connected subgraphs in the constraint graph.
- Sometimes we can exploit the subgraph structure (split the problem!) and get great results:
  - Suppose each subproblem has c variables out of n total variables: regular search 2^{80} – 4 billion years at 10 million nodes/second; $4*2^{20}$ – 0.4 seconds at 10 million nodes/second.

Tree Structure

- Theorem: if the constraint graph is a tree, the CSP can be solved in $O(n*d^2)$ time.
- General CSPs – $O(d^n)$
- Basic algorithm – choose a root, order the variables from root to leaves such that every node's parent precedes it in the ordering. For j from n down to 2, remove any inconsistencies in assignment. For j from 1 to n, assign X_j consistently with Parent(X_j).

Nearly-Tree Structure

- Conditioning: pick a variable, assign it a value and prune its neighbors' domains.
- Cutset conditioning: pick variables and assignments such that what's left is a tree.
- Basically – take some variables out of the equations!

Local Search Techniques and CSP

- We can still use our previous techniques for searching with CSPs! They're just not as efficient.
- To apply: allow states with unsatisfied constraints, we "move" to variables to new values. Heuristic?
- Variable selection – randomly select any conflicted variable.
- Value selection by min-conflicts – choose value that violates the fewest constraints.

Exercise

- 6.2 a-c