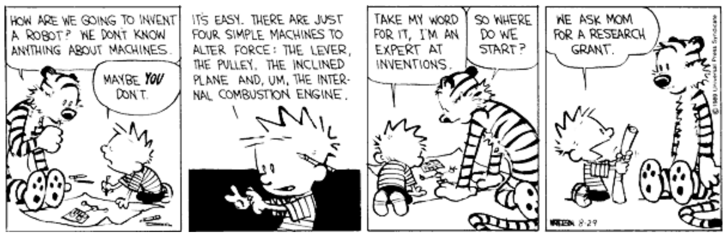
Constraint Satisfaction Problems

- CSP examples
- General search applied to CSP
- Backtracking
- Forward Checking
- Heuristics for CSPs



Constraint Satisfaction Problems (CSPs)

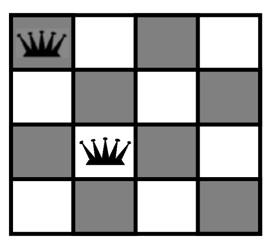
- Standard Search Problem:
  - State is a 'black box' – any data structure that supports goal test, evaluation, and successor
- CSP
  - State and goal test conform to a simple representation (problem specific)

CSP

- Definition:
  - Variables: $X_1, X_2, \ldots, X_n$
  - Constraints: $C_1, C_2, \ldots, C_m$
  - Each domain D_i for a variable X_i has a set of possible values $D_i=\{v_i^1, v_i^2, \ldots, v_i^k\}$
  - Each C_i specifies a subset of variables, and limits the combinations of values for that subset.
  - A state is then an assignment of values to some or all of the variables $\{X_i=v_i^1, X_j=v_j^3, \ldots\}$
  - A solution is a complete assignment that satisfies all constraints
    - Sometimes an objective function must be maximized/minimized as well

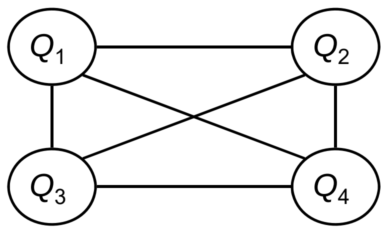
Example: 4-Queens

- Assume one queen in each column
- Which row does each go in?
  - Variables: Q_1, Q_2, Q_3, Q_4
  - Domains: $D_i = \{1,2,3,4\}$
  - Constraints:
    - $Q_i \neq Q_j$ (cannot be in same row)
    - $|Q_i - Q_j| \neq |i-j|$ (or the same diagonal)
  - Translate each constraint into set of allowable values for its variables
  - E.g., values for (Q_1, Q_2) are:
    (1,3) (1,4) (2,4) (3,1) (4,1) (4,2)



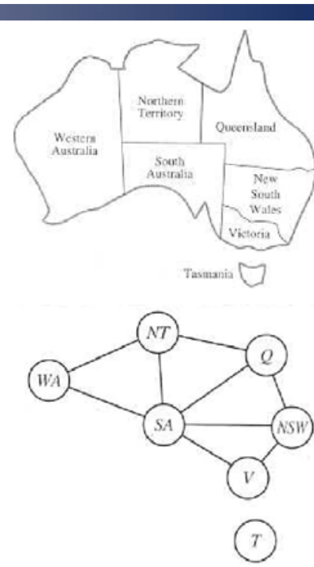
Constraint Graph

- Binary CSP - each constraint relates at most two variables, a binary constraint ($Q_2 \neq Q_4$)
- Higher order constraints can all be reduced to binary constraints with the introduction of auxiliary variables
- Constraint Graph – nodes are variables, arcs show constraints



Example: Map Coloring

- Color a map so that no adjacent territories have the same color
  - Variables: territories C_i
  - Domains: {Red, Green, Blue}
  - Constraints: $C_1 \neq C_2$, $C_2 \neq C_5$,…



Other Examples

- Assignment Problems
  - Who teaches which class
- Timetable Problems
  - Which class is offered when and where
- Hardware Configuration
- Spreadsheets
- Transportation Scheduling
- Factory Scheduling
- Floor Planning

- Note: Real world problems involve real-valued or continuous variables and sometimes preferences (represented as cost)

Applying Standard Search

- Start with straight-forward (dumb) approach
  - States are defined by the values assigned so far
  - Initial State: all variables unassigned
  - Operators: assign a value to an unassigned variable
  - Goal Test: all variables assigned, no constraints violated

Implementation

- CSP state keeps track of which variables have values
- Each variable has a domain and a current value
- Constraints may be represented
  - Explicitly – as a set of allowable values, or
  - Implicitly – by a function that tests for satisfaction

- Datatype CSP-State
  - Components: Unassigned, set of variables not assigned
    Assigned, set of variables that have values
- Datatype CSP-Var
  - Components: Name, for I/O purposes
    Domain, a list of possible values
    Value, a current value (if any)

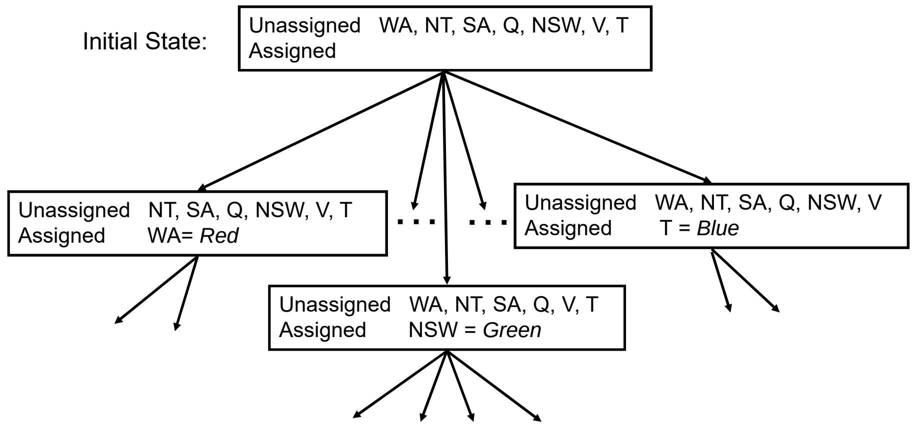
Searching for Map Coloring

Initial State:

Unassigned	WA, NT, SA, Q, NSW, V, T
Assigned	

What are the next states?

Searching for Map Coloring



Complexity of Simple Search

- Search algorithm to use?
  - Any uninformed search algorithm
- Maximum depth of space (m)
  - m = n = $|\boldsymbol{X}|$
- Depth of solution space (d)
  - $d = n$
- Branching factor (b)

$b_0 = nd$
$b_1 = (n\text{-}1)d$ leaves $= n(n\text{-}1)d^2$
$b_2 = (n\text{-}2)d$ leaves $= n(n\text{-}1)(n\text{-}2)d^3$
…
$n!d^n$ leaves in the tree

Making Improvements

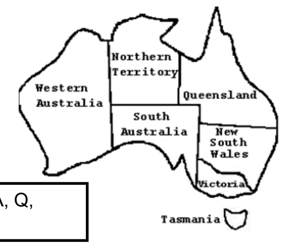
- Search for this type of problem can be improved significantly by noting
  - The order of the assignment variable assignment doesn't matter (commutative)
  - The same states appear many times in the tree
- If we want to improve this, we should enforce an ordering constraint to limit the repeated work.

Backtracking Search

- A modified depth-first search
  - Fixed order of assignment
    - Always make assignments in the same order
    - Reduces branching factor from $\sum_{i=1..n}|D_i|$ to $|D_i|$
  - Check for violated constraints
    - Don't create successors that violate a constraint
    - Don't expand a state that violates a constraint

Backtracking Search Example

- Coloring the map of Australia
  - Domain: {Red, Green, Blue}
  - Instantiate variables in order: (WA, NT, SA, Q, NSW, V, T)



WA, NT, SA, Q, NSW, V, T

WA=Red, NT, SA, Q, NSW, V, T

WA=Grn, NT, SA, Q, NSW, V, T

WA=Bl, NT, SA, Q, NSW, V, T

~~WA=Red, NT=Red, SA, Q, NSW, V, T~~

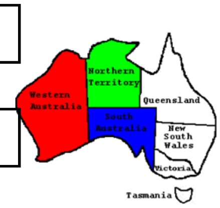
WA=Red, NT=Grn, SA, Q, NSW, V, T

WA=Red, NT=Bl, SA, Q, NSW, V, T

~~WA=Red, NT=Grn, SA=Red, Q, NSW, V, T~~

~~WA=Red, NT=Grn, SA=Grn, Q, NSW, V, T~~

WA=Red, NT=Grn, SA=Bl, Q, NSW, V, T



Problems with Backtracking Search

- If we select the wrong branch, we can explore all possible paths even if there is no way to succeed.

- Backtracking Search is basically uninformed search for CSPs
- Can solve up to 15-Queens in a reasonable time

Forward Checking

- An additional improvement is to consider the variable constraints for selection as far in advance as possible.
  - Whenever a variable *X* is assigned, delete from variable *Y*'s domain any value inconsistent with the value chosen for *X*.

- Can solve up to 30-Queens in a reasonable time

Forward Checking

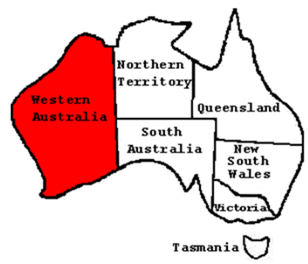
- Choose a color for Western Australia

	Red	Grn	Blue
Western Australia			
Northern Territory			
South Australia			
Queensland			
New South Wales			
Victoria			
Tasmania			

Forward Checking

- Choose a color for Western Australia

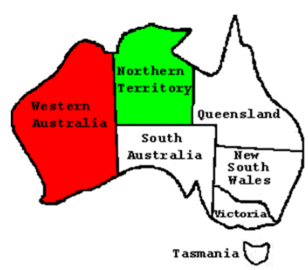
	Red	Grn	Blue
Western Australia	√	X	X
Northern Territory	X		
South Australia	X		
Queensland			
New South Wales			
Victoria			
Tasmania			



Forward Checking

- Choose a color for the Northern Territory

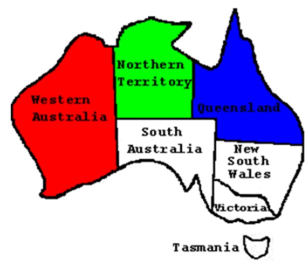
	Red	Grn	Blue
Western Australia	√	X	X
Northern Territory	X	√	X
South Australia	X	X	
Queensland		X	
New South Wales			
Victoria			
Tasmania			



Forward Checking

- Choose a color for Queensland

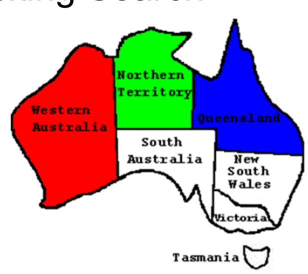
	Red	Grn	Blue
Western Australia	√	X	X
Northern Territory	X	√	X
South Australia	X	X	X
Queensland		X	√
New South Wales			X
Victoria			
Tasmania			



Forward Checking

- Because South Australia is overconstrained, we must back up as in Backtracking Search

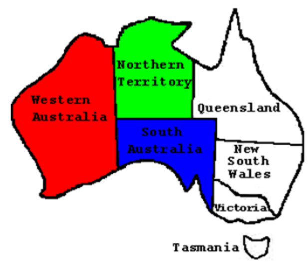
	Red	Grn	Blue
Western Australia	√	X	X
Northern Territory	X	√	X
South Australia	X	X	X
Queensland		X	√
New South Wales			X
Victoria			
Tasmania			



Forward Checking

- Choose a color for the South Australia

	Red	Grn	Blue
Western Australia	√	X	X
Northern Territory	X	√	X
South Australia	X	X	√
Queensland		X	X
New South Wales			X
Victoria			
Tasmania			

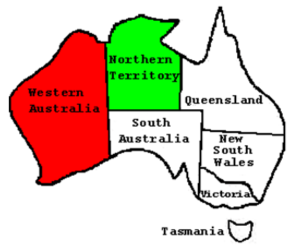


Help!

- Forward checking can help out some.
  - Went from solving 15-Queens to 30-Queens
- In the real-world, problems are much more complex
- The answer is?

Heuristics for CSPs

- We can make more intelligent decisions on
  - Which value to choose for each variable
  - Which variable to assign next
- Given WA = Red, NT = Green choose Q?



- Given WA = Red, NT = Green, choose what?
- Can solve n-Queens for n = 1000

Heuristics for CSPs

- We can make more intelligent decisions on
  - Which value to choose for each variable
  - Which variable to assign next

- Given WA = Red, NT = Green choose Q?
  - Q = Red: the least constraining value
- Given WA = Red, NT = Green, what next?
  - Select for SA: the most constrained state
- Can solve n-Queens for n = 1000

Heuristic Forward Checking

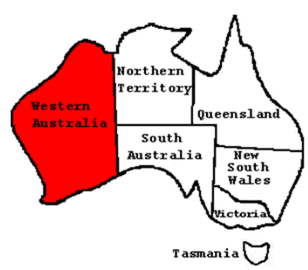
- Select most constrained state first, Western Australia

	Red	Grn	Blue
Western Australia			
Northern Territory			
South Australia			
Queensland			
New South Wales			
Victoria			
Tasmania			

Heuristic Forward Checking

- Choose a color for Western Australia

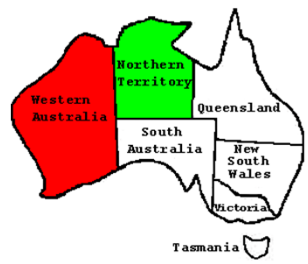
	Red	Grn	Blue
Western Australia	√	X	X
Northern Territory	X		
South Australia	X		
Queensland			
New South Wales			
Victoria			
Tasmania			



Heuristic Forward Checking

- Choose a color for the 'most' constrained state, Northern Territory

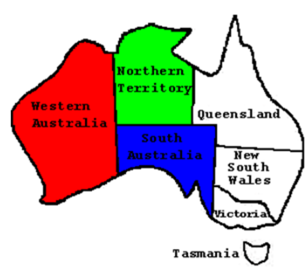
	Red	Grn	Blue
Western Australia	√	X	X
Northern Territory	X	√	X
South Australia	X	X	
Queensland		X	
New South Wales			
Victoria			
Tasmania			



Heuristic Forward Checking

- Choose a color for the 'most' constrained state, South Australia

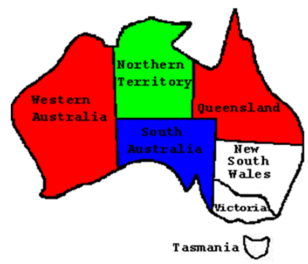
	Red	Grn	Blue
Western Australia	√	X	X
Northern Territory	X	√	X
South Australia	X	X	√
Queensland		X	X
New South Wales			X
Victoria			X
Tasmania			



Heuristic Forward Checking

- Choose a color for the 'most' constrained state, Queensland

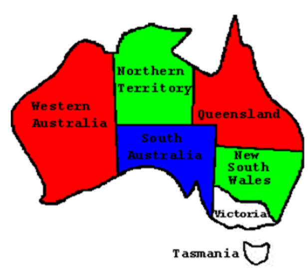
	Red	Grn	Blue
Western Australia	√	X	X
Northern Territory	X	√	X
South Australia	X	X	√
Queensland	√	X	X
New South Wales	X		X
Victoria			X
Tasmania			



Heuristic Forward Checking

- Choose a color for the 'most' constrained state, New South Wales

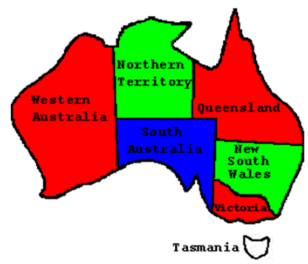
	Red	Grn	Blue
Western Australia	√	X	X
Northern Territory	X	√	X
South Australia	X	X	√
Queensland	√	X	X
New South Wales	X	√	X
Victoria		X	X
Tasmania			



Heuristic Forward Checking

- Choose a color for the 'most' constrained state, Victoria

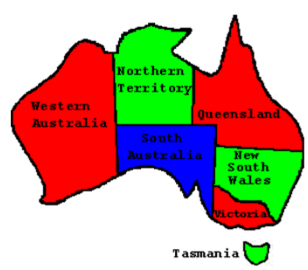
	Red	Grn	Blue
Western Australia	√	X	X
Northern Territory	X	√	X
South Australia	X	X	√
Queensland	√	X	X
New South Wales	X	√	X
Victoria	√	X	X
Tasmania	X		



Heuristic Forward Checking

- Choose a color for the 'most' constrained state, Victoria

	Red	Grn	Blue
Western Australia	√	X	X
Northern Territory	X	√	X
South Australia	X	X	√
Queensland	√	X	X
New South Wales	X	√	X
Victoria	√	X	X
Tasmania	X	√	X



Search Based CSP

- Dumb search can solve CSPs but does not take advantage of knowledge of states
- Backtracking search fixes the order of expansion to prune the search
- Forward checking search attempts to prune paths that can never reach a goal state
- Heuristics such as most constrained variable or least constrained value yield dramatic improvements

Iterative Algorithms for CSPs

- Hill-climbing and Simulated annealing typically work with “complete” states, i.e., all variables assigned
- To apply to CSPs
  - Allow state with unsatisfied constraints
  - Operators reassign variable values
- Variable selection
  - Randomly select any conflicted variable

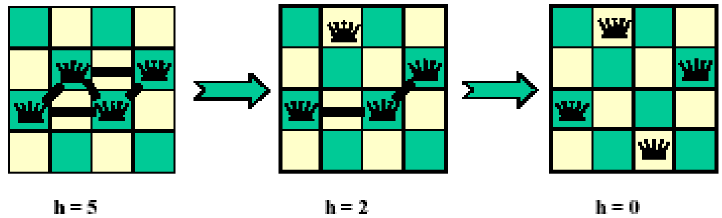
min-conflict Heuristic

- Choose value that violates the fewest constraints, i.e., hill-climb with

- $h(n)$ = # violated constraints

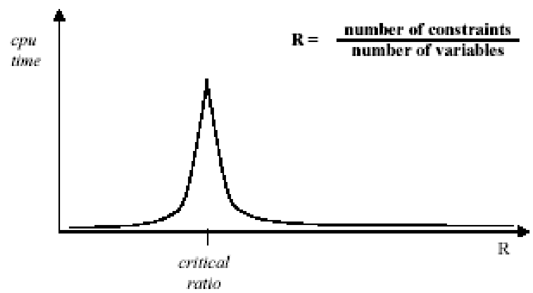
Example: 4-Queens

- States: 4 queens in 4 columns
- Operators: move queen in column
- Goal test: no attacks
- Evaluation: h(n) = number of attacks



min-conflicts Performance

- Given a random initial state, can solve *n*-queens in almost constant time for arbitrary *n* with high probability
- The same appears to be true for any randomly generated CSP except in a narrow range of the ratio



CSP Summary

- CSPs are defined by
  - A set of variables V_i with values from domain D_i
  - Set of constraints specifying allowable combinations of values for subsets of variables
- Traditional search is not efficient
  - Order of assignment is irrelevant
  - Adding assignments cannot correct a violated constraint

CSP Summary

- Backtracking search is depth-first search with a fixed order of assignment that checks for violated constraints
- Forward checking tracks legal values for unassigned variables and ends when a variable has no legal values
- Heuristics make a huge improvement in speed
- Iterative algorithms can work with *min-conflict* heuristic

Next Time

- Game Playing
  - Can we use what we have learned about search to develop computer programs that beat the worlds best human players at their own game?