Constraint Satisfaction

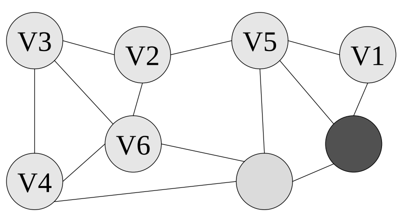
Introduction to
Artificial Intelligence
COS302
Michael L. Littman
Fall 2001

Administration

- HW 1 due, HW 2 assigned (3 question parts!)

Example CSP

- 3-coloring: color each node R,B,G. Connected pairs must differ.



Formal Definition

- Constraint satisfaction problem (CSP): {V, D, C}.
- V is set of variables V1…Vn.
- Each variable assigned a value from the domain D.
- C is set of constraints: each a list of vars, legal assignment set.

CSP for Graph Problem

- V = {V1, V2, V3, V4, V5, V6}
- D = {R, G, B}
- C = {(V1, V5) : { (B,R), (B,G), (R,B), (R,G), (G,B), (G,R) },
  (V2, V5) :{ (B,R), (B,G), (R,B), (R,G), (G,B), (G,R) },
  … }

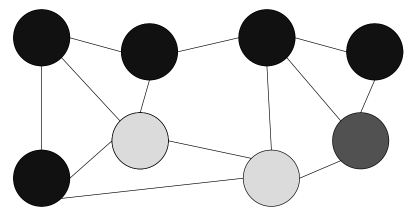
Solve via Search

- In a state, variables V1 through Vk are assigned and V{k+1} through Vn are unassigned.
- Start state: All unassigned.
- G(s): All assigned, constraints satisfied.
- N(s): One more assigned

Search Example

- s_0 = ??????
- $N(s_0)$ = {R?????, G?????, B?????}
- What search algorithms could we use?
- BFS? DFS?

DFS Looks Dumb



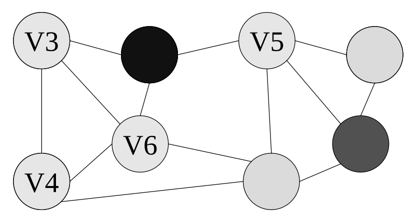
Making DFS Smarter

- Like A*, we can use a little bit of generic knowledge to guide the search process. Idea?

Don't peek. ☺

Consistency Checking

- Still flails a bit…

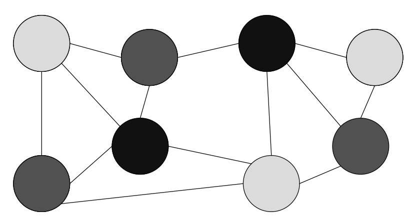


Forward Checking

- Track set of (remaining) legal values for each variable. Fail if any variable's set goes empty.
- How express this as a heuristic function h(s)?

Let's Try It

- Yay!



Computational Overhead

How would you implement forward checking?

Could the “empty domain” calculation be performed incrementally?

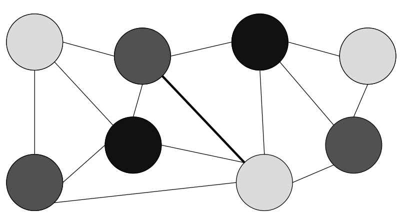
Constraint Propagation

As values are eliminated from a variable’s domain, this can start a cascade effect on other variables.

Expensive operation, so often performed only before search begins. Can be quite powerful…

Constraint Propagation

Solved without search



Example CSPs

Graph coloring

- Real applications include hundreds of thousands of nodes

VLSI board layout

8 queens, cryptarithmetic

Visual scene interpretation: Waltz

Minesweeper…

Minesweeper Example

0	0	1			
0	0	1			
0	0	1			
1	1	2			

Number is count of bombs in 8 adjacent cells

Minesweeper CSP

0	0	1	V1		
0	0	1	V2		
0	0	1	V3		
1	1	2	V4		
V8	V7	V6	V5		

V = { V1, …, V8 }, D = {B, S}

C = { (V1, V2) : { (B,S), (S,B) },

(V1, V2, V3): {(S,S,B), (S,B,S), (B,S,S)}, …}

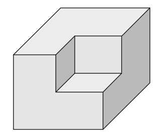
Using CSP to Decide

0	0	1	V1		
0	0	1	V2		
0	0	1	S		
1	1	2	V4		
V8	V7	S	V5		

How can we check if a value is forced?

Waltz Algorithm

Each "Y" intersection can be either concave or convex. Global interpretation is key.



Scheduling

Big, important use of CSPs.

- Makes multi-million dollar decisions in many industries.
- Used in space mission planning.
- Military uses.

Large state spaces

Generic CSP Algorithm

- If all values assigned and no constraints violated, done
- Apply consistency checking
- If deadend, backtrack
- Select variable to be assigned
- Select value for the variable
- Assign variable and recurse

Search Heuristics

Have some freedom in what variable to assign next

- Most constrained variable
- Most constraining variable

Freedom in value to assign to that variable

- Least constraining value

Implementing CP

Fundamental question:

- Given a constraint involving variables V1…Vk and possible values P1…Pk, remove impossible values

```
For i = 1 to k
  for each val in Pi
     if not possible(V1…V{i-1} { val }
                     V{i+1}…Vk, legalset)
          Vi -= { val }, restart
Return P1…Pk
```

What's "possible"?

Possible(V1…Vk, legalset)
For I1…Ik in legalset
  fail = 0
  For i = 1 to k
    if Ii not in Vi, fail = 1
  if fail == 0, return TRUE
Return FALSE

Formalizing Other CSPs

- Cryptograms
- Paint by Numbers
- Crossword Puzzles
- Four on the Floor
- Battleships

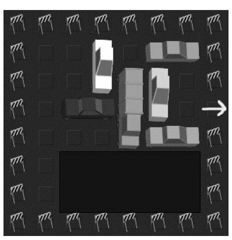
What to Learn

- Definition of a CSP: relationship to search.
- How to formalize problems as CSPs
- Use of improvements like consistency checking and forward checking

Homework 2

1. Consider the heuristic for Rush Hour of counting the cars blocking the ice cream truck and adding one. (a) Show this is a relaxation by giving conditions for an illegal move and showing what was eliminated. (b) For the board on the next page, show an optimal sequence of boards *en route* to the goal. Label each board with the f value from the heuristic.
2. Describe an improved heuristic for Rush Hour. (a) Explain why it is admissible. (b) Is it a relaxation? (c) Label the boards from 1b with the f values from your heuristic.

Rush Hour Example



Homework 2 (cont.)

3. Consider the 3-coloring problem on a graph with n nodes with the additional constraint that there *not* be exactly 2 nodes colored blue. (a) The most direct constraint for this involves all n nodes. How large is the corresponding legal assignment set for this constraint? (b) Explain how to specify this constraint more compactly by breaking it down into a set of simpler constraints. How many constraints do you add and how large is the legal assignment set for each one?