# **CPS331** Lecture: Constraint Propagation as an Alternative to Search

last revised February 22, 2012

## **Objectives:**

- 1. To introduce constraint propagation
- 2. To show how the human eye evidently does a form of constraint propagation

## Materials:

- 1. Projectable and handout of Garden Puzzle
- 2. Prolog generate and test + "human" solution process
- 3. Projectable of Sudoku Puzzle
- 4. Sudoku solver program; demo.sudoku puzzle
- 5. Projectable of partial line drawing showing ambiguity when only part seen
- 6. Projectable of complete line drawing to be labeled
- 7. Image Editor project (used in the past in CPS112 and 122) with parrots image
- 8. Projectable of Nilsson (1998) Figure 6.16
- 9. Projectables of progressive labeling of #6 using Waltz procedure
- 10.Projectables of two interpretations of line drawing with shading
- 11.Handout problem with possible vertex labelings and figure to label

## I. Introduction

- A. Recall that, in our introduction to search, we mentioned that a key problem in search is something called "combinatorial explosion". Basically, what we are dealing with is that state space size can grow very rapidly - e.g. we showed that, for a search with 13 steps and 3 alternatives at each node the state space contains over 1 million nodes! If we go to just 20 steps, the size increases to over 3 billion! When we hit 21 steps, the search space size is greater than the present population of the world.
- B. Although heuristic methods can help a great deal by helping us to focus on the alternatives that are most likely to lead to a solution, good heuristics are not necessarily easy to find.
- C. In certain cases, we can minimize search or even avoid it altogether by using a strategy called constraint propagation. While the problems to which this applies are limited in number, the benefits to be gained are enormous.

- D. One weakness of standard AI search techniques, like those we've been looking at, is that they are often quite different from the ways humans solve the problem.
  - 1. Recall what we saw in the case of solving the "Garden Dilemma Puzzle" using a search.

HANDOUT + PROJECT

a) The Prolog program we looked at used a naive generate and test strategy.

PROJECT

- b) Recall that we saw that, while this works for a small problem, it quickly falls prey to combinatorial explosion (e.g. with our program a problem with 15 people would take 25 times the age of the universe to solve!)
- c) For this particular sort of problem, I'm not aware of any good heuristics that would allow us to use an informed search (and I suspect this is not a hot research area!)
- 2. Of course, a human wouldn't solve the problem this way at all.

PROJECT human approach

That is, a human makes use of <u>constraints</u> to fairly quickly home in on a solution. For example, since we are told that each person bought a different tool, we can eliminate that tool as a possibility for the other four people once we learn which person bought it.

3. When a problem lends itself to the application of constraints, it is often possible to arrive at a solution very quickly.

### **II. Constraint Propagation in Sudoku**

A. We will use, for further examples, the familiar Sudoku puzzle.

- 1. Explain the puzzle
- 2. Solving a Sudoku can be regarded as a search problem in which the states correspond to partially filled in puzzles. The start state is the given initial state
  - a) The goal state is a state in which all squares are filled in
  - b) The operators are "write a particular number in a particular square"
  - c) Example: consider a typical puzzle:

PROJECT Sudoku Solver program with demo puzzle

				3	7		2	
	4	2						
9	1					7		5
	6		5			3		
			9	2	6			
		5			3		9	
5		6					1	8
						2	7	
	2		6	7				

(1) The available operators (consistent with the rules of the game) are

Put a 6 in the upper-left corner Put an 8 in the upper-left corner (1-5, 7, and 9 are ruled out by others in same row/column/ block) ... Put a 3 in the lower-right corner Put a 4 in the lower-right corner Put a 9 in the lower-right corner (1, 2, and 5-8 are ruled out by others in same row/column/ block)

(2) In fact, for this particular state, there are 175 legal operators!

- 3. However, if we tried to use a conventional search technique to solve this puzzle, we would quickly run into combinatorial explosion
- B. The key to solving a puzzle like this is to recognize that, in a legal puzzle, at any time, there will always be one or more squares whose value is constrained to a single value.

Example: In this puzzle, the third square in the first row is constrained to be an 8 by the fact that no other value is possible.

- -3, 7, 2 are eliminated by other values in the same row
- 5, 6 are eliminated by other values in the same column
- 4, 1, 9 are eliminated by other values in the same block

Example: In this puzzle, the second square in the first row is constrained to be a 5 by the fact that this is the only square in the block where a 5 can go. (5 cannot go in the other vacant squares in the block due to a 5 elsewhere in the same column)

Example: In this puzzle, the first square in the second row is constrained to be a 7 by the fact that this is the only square in the block where a 7 can go. (7 cannot go in the other vacant squares in the block due to a 7 elsewhere in the same row.)

- C. This leads to the following strategy for solving this puzzle:
  - 1. Initial setup:
    - a) Associate with each square the set of legal values that can appear in that square, based on constraints propagated from other squares in the same block, row, and column.

Example: The legal values for the vacant squares in the upperleft block are

{ 6, 8 }	{ 5, 8 }	{ 8 }
{ 3, 6. 7. 8 }	-	-
-	-	{ 3, 8 }

b) Associate with each block a list of squares where each value that has not been used in that block can occur [ which can be extracted from the above ]

Example: In the upper-left block

- 1 has been used
- 2 has been used
- 3 { middle-left, lower-right }
- 4 has been used
- 5 { upper-middle }
- 6 { top-left, middle-left }
- 7 { middle-left }
- 8 { top-left, top-middle, top-right, middle-left, lower-right }
- 9 has been used
- c) Do something similar for each row
- d) Do something similar for each column
- 2. Now perform the following process repeatedly until the puzzle is solved

- a) Select a set which contains just one element (if there is none, we're stuck)
- b) Give the corresponding square the appropriate value
- c) Propagate constraint resulting from this choice to other cells
  - (1) The value just given to the cell can be removed from the sets of possible values for other cells in the same row, column, and block
  - (2) The square can be removed from the sets of possible locations for unused values in the row, column, and block of which it is a part
- 3. DEMO: Run Sudoku Solver program with demo.sudoku
  - a) Click OK to get to point where solution has been set up
  - b) Click Show Details to show possible values for each cell Observe:

- If a cell has only one possible value, it is shown in yellow. (This includes cells which represent the only possible location for the value in a row, column, or block.)

c) Step through first few steps of solution, showing how constraint is propagated

Observe:

- The cell whose value is being fixed is highlighted in red

- Cells with a single possible value are selected in the order in which this fact was discovered - hence the 7 in the second row being the first to be fixed (only possible location for 7 in its block) - A square which is about to have its set of possible values reduced is highlighted in purple and then on the next step is reduced to new value.

D. Actually, this process is not totally sufficient

Demo: Click solve - note how solution is stuck. (This was actually a fairly challenging puzzle).

- This puzzle can be solved by noting another constraint. In the middle block, the only place where an 8 can occur is in the top row. Hence, one of these two squares will eventually be an 8 - in which case the 8 in the middle-right block top row is not actually possible, forcing this square to be a 4.
- 2. Manually enter and show how puzzle can now proceed to solution.
- 3. This sort of constraint could also be incorporated programatically but I didn't choose to do so in this program.

### **III.Constraint Propagation in Vision**

- A. One could argue that the Sudoku puzzle is still something of a "toy problem". So let's look at another example of constraint propagation this one seeming to reflect something that actually occurs as a part of human vision.
- B. Consider the following line drawing: (PROJECT)



1. Is point "N" closer or farther than point "I"?

ASK

Be sure to get both possible answers

2. Now consider a more complete drawing of which this was a part.

# PROJECT



- a) Now, the correct answer should be clear.
- b) However, if you focus your attention on the portion we looked at earlier alone, does N seem to leap out of the page at you?

ASK

What's going on?

ASK

3. It appears that our visual system is particularly sensitive to the presence of edges.

Examples?

ASK

4. One thing that is often done in computer vision is to convert raw data to a line drawing. There is actually a fairly straight-forward way to do this computationally, which seems (based on experiments done on frogs) to be similar to what animal eyes actually do.

DEMO: Image Editor project with parrots drawing - convert to edges

- 5. To interpret a line drawing, it is necessary to interpret the individual lines
- C. One of the earliest uses of constraint propagation in AI was in conjunction with interpretation of line drawings. We will consider a simplified version of this procedure here. (Even the full procedure as originally developed by David Waltz has limitations, which subsequent work has addressed).
  - 1. Lines occur in images for a number of reasons
    - a) Actual physical edges
      - (1) A boundary between an object and the background
      - (2) A convex edge in the interior of the object
      - (3) A concave edge in the interior of the object
    - b) Pseudo-edges
      - (1)Crack
      - (2) Shadows
    - c) Markings on the surface of the object which we may choose to treat as real edges
  - 2. To interpret a line drawing, we need to interpret the lines. For simplicity, the example we will develop here will consider only lines corresponding to actual physical edges. Our ultimate goal is to interpret each line as corresponding to one of the four kinds of physical edge.



Example: The two ways of "seeing" the relationship between points N and I correspond to two ways of interpreting the edges



The "correct" interpretation (the one on the right) is actually determined by constraint propagated from the surrounding context.

- 3. Waltz's procedure turns the problem of labeling lines into a problem of labeling junctions. It relies on two physical constraints to make the problem tractable:
  - a) Though many combinations of line labelings at a junction are are combinatorially possible, only certain combinations are physically possible.

PROJECT: Nilsson Figure 6.16. This gives a set of of junction labelings in a restricted environment where we impose the following limitations:

- (1) No shadows or cracks are allowed; only real edges. (Thus, only the four line labels we mentioned earlier are needed.)
- (2) All junctions are formed from at most three faces. (The pyramids of Egypt are ruled out.)
- (3) The viewing angle is not singular; what we see would not be drastically altered by a slight movement of position.

(These classes of junctions are called, respectively, V's, W's, Y's, and T's)

- (4) Note that real vision systems must deal with a much more complex set of conditions, including shadows, cracks, and junctions formed from more than three faces. The set of possible labelings of junctions would be very hard to enumerate by hand, but has been done by computation.
- b) Each line connects two junctions. The labelings at the two junctions must both assign the same label to the line.
- 4. The problem of finding a set of consistent line labelings could be viewed as a search problem, with a particular assignment of labels being a state. In this case, the goal would be to find a physically consistent set. However, the size of the search space would be very large e.g. with just the 4 labels and 10 lines, we would have 4^10 (> 1 million) states to consider; and with 20 lines, we would have over a trillion. Waltz's procedure dramatically reduces the size of the search space by using constraint propagation, as we did in the Sudoku example. (In fact, when I first started solving Sudoku's I recognized that the way to approach a solution was to use this technique that I already knew about!)
  - a) At any given time in the procedure, each junction has associated with it a set of possible labelings. (Ultimately, the set associated with each junction will become a singleton.)

- b) Initially, we label the outside edges as boundaries. Then we associate with a junction the set of all possible labelings for junctions of that type (L, fork, arrow etc.) which are consistent with these labeling. (In some cases, this will mean that all possibilities for a junction are available.)
- c) Constraints propagated from neighbors allow us to eliminate certain elements from the set of possible labels for the junction, until we are (hopefully) eventually left with just one.Constraint is propagated as follows: whenever we change the set of labelings on a junction (even just by making it smaller), we examine each of its neighbors.
  - (1) Let J be a junction we have just learned something new about
  - (2) Let E be an edge connecting it to a neighbor (N).
  - (3) The set of labels for J determines a set of possible labelings for E.For example, if none of the possible labels for J has E labeled as a "+", then "+" is not one of the possible interpretations for E.
  - (4) We eliminate from N's set of labelings any label which requires an interpretation for E not allowed by J's labeling.

(5)Eventually, the set of possible labelings for J or N determines a unique label for E. At this point, we mark E.

d) Whenever the set of labelings for a junction's edges eliminates all possibilities for the junction save one, we can label the junction. D. Example - labeling the figure we used earlier.

## PROJECT AGAIN

- 1. First, label all outside boundaries with an arrow going clockwise starting with A.
- 2. The W's at B, D, and F must be labeled with +'s on their third barb.
- 3. The Y at G therefore becomes all +'s, which is one of the possibilities for a Y.

PROJECT



- 4. This finishes the outer edges. We now must plunge into the interior.
  - a) If we start with H, all 6 "V" labelings are initially possible.



b) Likewise, for I all 3 "W" labelings are initially possible, since each is consistent with some labeling for H.



c) At J, we can eliminate 2 of the 6 "V" labelings as inconsistent with the set of labelings for I. (There is no labeling for I that has a boundary edge going from J to I.) That leaves the following possibilities:



d) At K all 4 "T's" are possible



e) But now, we can propagate some constraint back to J. The two "V's" having a boundary from J to K are out, as is the "V" having the J-K edge a -, since K constrains the J K edge to be a K to J boundary. Thus, the labeling for J has been fixed as:



f) Further, the fixing of J's labeling forces the labeling for I, which in turn forces the labeling for H, leading to:

PROJECT



g) At L, only two labelings are possible consistent with K:







- i) However, since all of M's labelings have ML a boundary edge, this eliminates one possibility for L, forcing the remaining one as the only possibility.
- j) Finally, at N only two labelings are consistent with I:



- k) This, in turn, reduces possible labelings of K and M to two each.
- 5. At this point, our labeling looks like this

PROJECT

Α

F



We can go no further. The Waltz procedure has uncovered a genuine a genuine ambiguity - is the center area (bounded by K,L,M,N) a solid bottom or a hole? Further data would be needed to settle this. (E.g. if this area were a different color or shading from the background, it's probably a bottom; if the same as the background, probably a hole.

To see this, consider these two different versions of our drawing with shading:

#### PROJECT



E. Exercise to do in class:

Waltz's procedure problem

HANDOUT with possible labelings and figure to label.