

AI lecture, September 29, 2006

AIMA, Chapter 5, Sections 1–2

© 2006. Gene B. Chase

A. Course Red Tape.

1. Did you turn an evaluation of Eugene Rohrbaugh in to Jean McCauslin, Angela Hare or me? G Cook, Alan G Melious, Ethan G Prendergast, Michael "Joe"
2. Submit team partner requests, for exam, labs, and maybe project, on a card.
3. For Monday, October 2, finish reading Chapter 5. Pay special attention to items in bold in the outline from class, below. You are ultimately only responsible for class lectures and the parts of those that are expounded on in AIMA. Note for example below that a page outside of your Chapter 5 reading is referenced about Truth maintenance. Try Exercises 5.11 and 5.13 for class discussion.

B. Demos from Q:\InstructorFiles\Chase_Gene\ai\.

1. 8 Queens demos

- Genetic Algorithm: Chess.exe (C++ has the :: qualifier; the interface goes in *.h)
- DFS, BFS, DLS, IDS, Simulated Annealing*, Hill Climbing*: Java
SuccessorFunction.java, HillClimbingSearch.java (Java 1.5 has < > and ArrayList)

2. 8-puzzle demo

- Depth-limited search (DLS)
- Iterative DLS
- Greedy best first, MisplacedTileHeuristic
- Greedy best first search, ManhattanHeuristic
- AStar search, MisplacedTimeHeuristic
- AStar search, ManhattanHeuristic
- Simulated annealing *

* Simulated annealing didn't work, possibly because of cooling schedule. Hill climbing didn't work because it found a local rather than a global minimum of the objective function.

C. Chapter 5: **Constraint Satisfaction Problems (CSP's)**.

1. We may want any consistent solution; we may want an optimal one (min or max).
Example from today's OPL exam: TK!Solver solves a system of equations by constraint.
2. Examples.
 - Linear programming.
Continuous variables, but Dantzig's theorem gives us a finite number of places to try to find minima or maxima.
 - Simulated annealing. Also continuous variables.
 - Integer solutions to $x^2 - dy^2 = k$ (Pell's Equation), for $k=1, 4$. [Diophantine Equations]

- Map coloring (color Australia with 3 colors, one per province). [4 Color Theorem]
—*Prototypical*, because in general constraints will form a **constraint graph**, where the nodes represent the variables, the coloring represent the values, and the arcs represent the constraints. How would we represent a constraint like $X+Y+Z=5$, which is among three variables? Answer: For any CSP, an equivalent CSP can be created by defining new variables on which all constraints are binary.
 - Sudoku.
 - Cryptarithmic (e.g. SEND + MORE = MONEY).
 - N Queens.
 - N-puzzle. (I did not explain in lecture how it would be framed as a constraint problem. Here's a place for you to think in preparation for the next exam.)
 - Scheduling classes and professors and rooms and students.
 - Logic: $p(x) \wedge q(y) \rightarrow r(x)$, with x and y from some **domain**.
 - Crossword puzzle constructing. (You could google many free examples.) [p. 158]
 - Dr. Chase's "Richard Nixon problem"
Work of Jon Doyle on truth maintenance by retraction is a CSP solution. [p. 360]
 - Constraint programming languages.
Prolog for logic (You saw monkeys and bananas.); TK!Solver for equations.
3. CSP's for discrete variables can be solved by any search algorithm.
- Problem: Exponential number of possible solutions; in fact "**NP-complete**" even on finite domains. (We will give details about NP-completeness later.)
4. Heuristic improvements over simple backtracking: In assigning unassigned variables,
- a. Single step look-ahead.
 - Pick the most constrained variable (**Minimum Remaining Values Heuristic**); or,
 - Pick the node of maximum degree in the constraint graph; or,
 - Pick the value of a variable that results in min number of conflicts with other variables. (**Min Conflicts Heuristic**. Think scheduling a new course by the Registrar.)
 - b. More generally, **propagate constraints** more than a single step.
 - Problem: Propagating constraints is like forward chaining, so how is that any faster than blind search?
 - Answer: If we propagate constraints fully, it is like forward chaining, and so slow. If we don't propagate, we don't take advantage of look-ahead to avoid dead ends, and so it is slow. Depth of propagation is another parameter to adjust in defining a class of problem solvers!
- * 5. A discrete variables CSP can usually be cast as a "**SAT**" logic problem: Where we want to find a set of True and False assignments to boolean variables (a model of the real world) in an "open sentence" (one with variables) that makes the sentence true.
- We say that **the model "satisfies" the sentence**.
 - Even with only 3 variables (3-SAT), the CSP logic problem is NP-complete.
" Stephen Cook (1971) proved that SAT is NP-complete. [p. 235]
 - Because SAT is *also prototypical* for CSP's, we're jumping to Chapter 7 next.
" The syllabus says that, and we're quite ahead in the syllabus except for the labs.