CSC 416

Question Set 5

Jacob Peck

A set of questions based on chapter 4 of Coppin's Artificial Intelligence Illuminated. This chapter deals with search methodologies.

GRADE ≈ 90

1. Answer in concise, precise terms.
    - What is a *problem space*?
        - A problem space is a representation of a problem and all the possible solutions and paths to those solutions of said problem.
    - What does *forward chaining* refer to?
        - Forward chaining refers to starting at the initial state and working through legal transitions to a goal state.
    - What does *backward chaining* refer to?
        - Backward chaining refers to starting at a goal state and working backwards through a series of legal transitions to the initial state.
2. In English, characterize each of the following search methods according to how the list of unexplored nodes is maintained.
    - Breadth-first search.
        - A breadth-first search list of unexplored nodes is maintained as a queue, i.e., first-in, first-out.
    - Depth-first search
        - A depth-first search list of unexplored nodes is maintained as a stack, i.e., last-in, first-out.
3. Describe, in English, depth first iterative deepening search. Explain why DFID is reasonably efficient. Explain why DFID might be preferable to use than depth first search.
    - Depth first iterative deepening search is a combination of depth-first search and breadth-first search. DFID consists of applying depth-first search limited to successively deeper levels, going across the breadth of a tree. DFID is reasonably efficient for large trees because the vast majority of the nodes which have to be explored are in the last level, so little computing time is wasted doing redundant checking. DFID is optimal, guaranteeing the optimal path.
4. Explain what is meant by the following terms in relation to search methods:
    - Complexity.
        - Complexity (in reference to time or space) refers to how much computing power is actually required to perform a search. Often represented in big-O notation.
    - Completeness.
        - Completeness refers to the ability of a specific method to find a goal, if one exists.
    - Optimality.
        - An optimal search method is guaranteed to find the best solution that exists, if a solution exists.
5. Provide a definition of the word "heuristic". In what ways can heuristics be useful in search? Name three ways in which you use heuristics in your everyday life.
    - A heuristic is a rule that allows a search method to operate more efficiently, given information that is known about the problem. One example of a heuristic would be looking for an item in the grocery store—grapes aren't kept in the cereal aisle. Checking for a specific email, is another example—if you know the email was sent by professor Graci, you wouldn't be looking at emails sent by professor

Wenderholm. And when taking a shower, you wouldn't grab a textbook and a pen to bring into the bathroom with you.

6. Consider the following state of an eights puzzle:

```
6 3 1
2 B 4
7 8 5
```

Also, suppose that this particular eights puzzle has the following goal:

```
1 2 3
8 B 4
7 6 5
```

- Write down, each in the form of a two dimensional array, the four children of the given state.

```
6 B 1   6 3 1   6 3 1   6 3 1
2 3 4   B 2 4   2 4 B   2 8 4
7 8 5   7 8 5   7 8 5   7 B 5
(c1)    (c2)    (c3)    (c4)
```

- Write down, in English, heuristic H1 for the eights puzzle, as described in the text.
  - The H1 value of a node is the number of tiles that are not in the correct spot.
- Compute the value of H1 for each of the four children of the given state. Which move would be made according to heuristic H1?
  - c1 = 5
    c2 = 5
    c3 = 6
    c4 = 5
    Any move other than the move leading to c3 would be made according to H1.
- Write down, in English, heuristic H2 for the eights puzzle, as described in the text.
  - The H2 value of a node is the amount of moves (vertical and horizontal) it would take for each tile to move into the correct spot.
- Compute the value of H2 for each of the four children of the given state. Which move would be made according to heuristic H2?
  - c1 = 2 + 2 + 2 + 0 + 0 + 3 + 0 + 2 = 11
    c2 = 2 + 1 + 1 + 0 + 0 + 3 + 0 + 2 = 9
    c3 = 2 + 2 + 1 + 1 + 0 + 3 + 0 + 2 = 11
    c4 = 2 + 2 + 1 + 0 + 0 + 3 + 0 + 1 = 9
    Either the move leading to c2 or the move leading to c4 would be made according to H2.

7. In English, characterize each of the following search methods according to how the list of unexplored nodes is maintained.
   - Best first search.
     - Best first search re-organizes the list of unexplored nodes such that the "best" candidates are at the front every time nodes are added to the list.
   - Beam search.

- Beam search "weeds out" all but the best *n* nodes from the list of unexplored nodes before every expansion.

8. For each of the following approaches to optimal path search, search the Web for a site that you think provides an interesting introduction to the search technique. For each technique/site, write down the URL along with a short description of what you like about the site.
    - British Museum procedure.
        - Site: http://en.wikipedia.org/wiki/British_Museum_algorithm
        - This Wikipedia page offers an interesting theoretical example of meta-problem searching: finding the smallest possible program to solve a problem. In theory, the British Museum procedure would find the solution, but in terms of practicality, this would take many times the lifetime of the universe to calculate, and the halting problem causes an issue as well.
    - A*.
        - Site: http://www.edenwaith.com/products/pige/tutorials/a-star.php
        - This page offers a detailed explanation of what A* search is, and also provides some example code in LISP, with links to an implementation in Java.
    - Uniform cost search.
        - Site: http://www.cs.utah.edu/~hal/courses/2009S_AI/Walkthrough/UCS/ucs.html
        - This site from the University of Utah has an overview of uniform cost search, with an animated example search through a graph.
    - Greedy search.
        - Site: http://yuval.bar-or.org/index.php?item=9
        - This site presents the greedy search algorithm within the frame of the famous chess puzzle called the *n Queens* problem.
9. Investigate the file search utility on your computer.

    My linux machine has Beagle installed, which I use for my file searches. Beagle is based upon Apache Lucene, so all answers here reflect that fact.

    - Which type of search method do you think it uses?
        - Lucene uses a (heavily modified) best-first search algorithm. (http://lucene.apache.org/java/3_0_2/scoring.html#Algorithm)
    - Why do you think this particular search method was chosen?
        - This particular algorithm seems rather well suited for searching through directories and subdirectories, as it will return a file in the shortest amount of time, compared to a basic, uninformed search.
    - What problems could this approach cause it?
        - If the weighting algorithm is off, the file could be returned rather slowly, wasting processor cycles.
    - How well does it work when it is searching directories with large numbers of files in them?

- I find that Beagle (and therefore Lucene) works rather well under these conditions. Almost instant results, if the file is in the immediate directory you are searching, and only a slight slow-down if it is deeper down in the filesystem structure.