CS 380 Artificial Intelligence, Spring 2019
Sect: 002
Homework 3 - Informed Search

by JL Popyack

Due Date: Wednesday, May 15, 2019, 11:55 PM

PURPOSE:

This assignment provides experiences with Graphsearch, heuristics and A*, and also preliminary experiences with adversarial search and the Minimax algorithm.

THE ASSIGNMENT:

This assignment consists of one section:

1. Written problems to be turned in.
2. A program to be completed.

Click here to return to top of this page.

---

1. **Written Problems to be turned in**:
   A. **Algorithm A* *(25 points)*:** Problem 3.23 *from 3rd Edition* of the text. It makes use of Figure 3.2 (p. 68) and Figure 3.22 (p. 93).
   B. **Algorithm A* (8-Puzzle)** ***(25 points)***

   Consider the familiar 8-Puzzle problem – a 3x3 sliding block puzzle featuring 8 numbered pieces and a blank cell is to be rearranged to place the tiles in a specified order. We have examined two heuristics (referred to below as heuristics h_1 and h_2). We introduce some other heuristics appropriate for the 8-Puzzle.

   $h_0(node)$: 0
   $h_1(node)$: number of tiles out of place
   $h_2(node)$: sum of distances out of place
   $h_3(node)$: 2**DT(node)* – defined below
   $h_4(node)$: $h_2(node)$ + 3**S(node)* – see below
   $h_5(node)$: $h_1(node) + h_3(node)$
   $h_6(node)$: $h_2(node) + h_3(node)$
   $h_7(node)$: maximum of all *admissible* heuristics in $\{h_1(node), h_2(node) \ldots h_6(node)\}$

2	3	4
8	1	6
7		5

Sample A

1	2	3
8		4
7	6	5

Goal State

2	1	3
8		4
7	5	6

Sample B

   DT(node) counts "direct tile reversals" – in which two adjacent tiles must be exchanged to be in their respective proper positions (*DT(node)=2* for sample state B, since 1-2 and 5-6 are direct reversals.)

   S(node) is a "sequence score" – for every tile around the edge, count 2 for every tile not followed by its proper successor and 0 for every other tile. If there is a piece in the center, add 1.

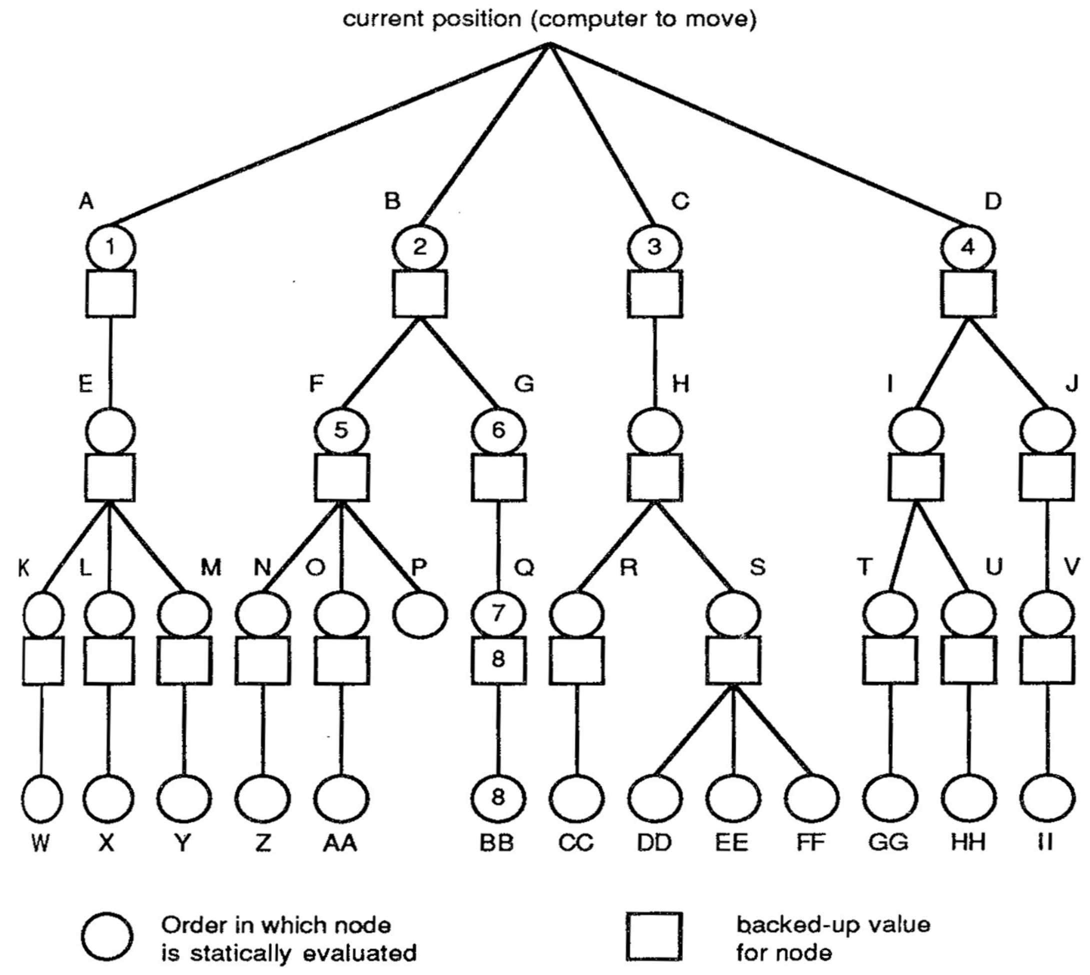
   For each of these heuristics, determine whether it is *admissible*, and justify your answer.

   For the *admissible* heuristics above, show the precedence from least to most informed¸ e.g., $h_i(node) \leq h_j(node) \leq \ldots$

   C. **Minimax, Alpha-Beta *(from Tanimoto) (25 points)* :** The game tree in the figure below illustrates the possible moves, to a depth of 4, that can be made from the current position (at the root) in a hypothetical game between a computer and a human.

The evaluation function is such that the computer seeks to maximize the score while the human seeks to minimize it. The computer has five seconds to make its move and four of these have been allocated to evaluating board positions. The order in which board positions are evaluated is determined as follows: The root is "searched." A node which is at ply 0, 1, 2, or 3 is *searched* by

- generating its children,
- statically evaluating the children,
- ordering its children by static value, in either ascending or descending order, so as to maximize the probability of alpha or beta cutoffs during the searches of successive children,
- searching the children if they are not in ply four, and
- backing up the minimum or maximum value from the children, where the value from each child is the backed-up value (if the child is not in ply four) or the static value (if the child is in ply four).


current position (computer to move)
A
B
C
D
1
2
3
4
E
F
G
H
I
J
5
6
K
L
M
N
O
P
Q
R
S
T
U
V
7
8
8
W
X
Y
Z
AA
BB
CC
DD
EE
FF
GG
HH
II
Order in which node is statically evaluated
backed-up value for node

Static values for nodes: A 4, B 15, C 13, D 10, E 20, F 9, G 8, H 10, I 10, J 8, K 5, L 20, M 3, N 7, 0 6, P 0, Q 9, R 12, S 10, T 15, U 10, V 9, W 7, X 22, Y 2, Z 7, AA 5, BB 8, CC 15, DD 12, EE 13, FF 13, GG 20, HH 22, II 18.

The computer requires 1/7 seconds to statically evaluate a node. Other times are negligible (move generation, backing up values, etc.). The computer chooses the move (out of those whose backed-up values are complete) having the highest backed-up value.

a. Give the order in which nodes will be statically evaluated. Hint: the first eight nodes have been done for you. Be sure to skip the nodes that alpha-beta pruning would determine as irrelevant. Indicate where cutoffs occur.
b. Determine the backed-up values for the relevant nodes. Node Q has been done for you.
c. Keeping in mind that it takes 1/7 seconds per static evaluation, what will be the computer's move?
d. Now assume that it takes 1/8 seconds per static evaluation. What will be the computer's move?

Click here to return to top of this page.

2. **Programs to be turned in**:
   A. ***Search: Backtracking - Pegboard (25 points)***: This is a continuation of the programming exercises from the previous assignment: (`https://www.cs.drexel.edu/~jpopyack/Courses/AI/Sp19/assignments/HW2/index.html#Prog`) :
      a. In the previous assignment, you used the backtracking algorithm to find a solution to the *Pegboard* problem, however no attempt was made to either find an efficient solution or to find a solution efficiently. For this assignment, you are to:
         i. devise a heuristic *h(s,r)* which estimates the desirability of applying rule *r* when the system is in state *s*, and returns a numerical value, where a low number is considered more desirable than a high number. (Your heuristic may base its decision entirely on the rule *r*, or on the new state *s'* produced after *r* is applied to *s*, or even something else).
         ii. Use this heuristic in **`applicableRules`** so that the list of rules returned are arranged in order from most desirable to least desirable.
         iii. Solve the problem, using your heuristic.
         iv. Output should report on the number of calls to **`backTrack`**, both with and without use of your heuristic, and number of failures before finding the solution.

         It is likely you will need to experiment with several heuristics before finding one that provides suitably improved performance. Describe all of the heuristics you used, and report on the results of using them.

WHAT TO SUBMIT:

All homework for this course must be submitted electronically using Bb Learn. ***Do not e-mail your assignment to a TA or Instructor!*** If you are having difficulty with your Bb Learn account, ***you*** are responsible for resolving these problems with a TA, an Instructor, or someone from IRT, before the assignment it due. It is suggested you complete your work early so that a TA can help you if you have difficulty with this process.

For this assignment, you must submit:

- A PDF document with your answers to the "Written problems to be turned in".
- Source code, documentation and results for your program.

Click here to return to top of this page.

ACADEMIC HONESTY:

You must compose all program and written material yourself, including answers to book questions. All material taken from outside sources must be appropriately cited. If you need assistance with this aspect of the assignment, see a consultant during consulting hours.

Click here to return to top of this page.