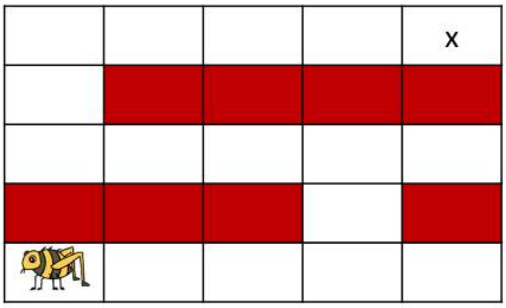
Artificial Intelligence
HW 1: Search and games

Published: 2022/02/01
Due date: 2022/02/22 10:55
In class solutions: 2022/02/22

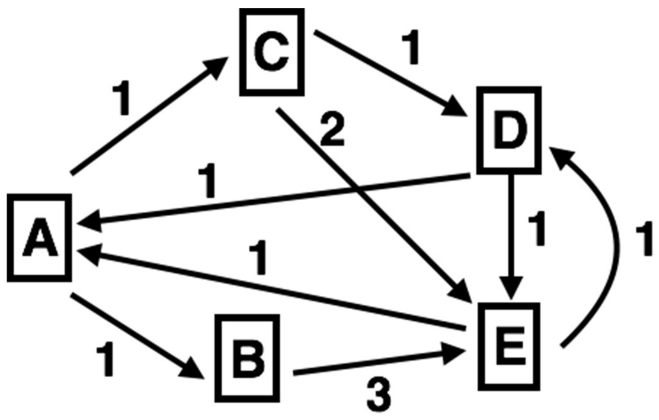
---

1. **Informed Search and A*.** A flea is in a maze with dimensions 5 × 5, as shown in the figure. At each time step, the flea can move north, east, south, or west (but not diagonally) to an adjacent square if that square is currently free. The squares may be blocked by walls (as indicated by the red squares), but the map is known. The flea starts where indicated in the figure and must end at the X. Help the flea to solve the maze and get to the X. In addition to moving through the maze as usual, the flea can jump and get on top of a wall. When it is on a wall, the flea can walk along it as it would when it is in the white squares of the maze. You can also jump off the wall and go back to the maze. Jumping to the wall has a cost of 2, while all other actions (including returning to the labyrinth) have a cost of 1. Note that the flea can only jump to walls that are in adjacent squares (either to the north, south, west or east of the flea).



a) Formalize the states, actions/operators to generate successors (indicate the order), the necessary preconditions to apply them, the resulting postconditions and associated costs in this problem. What is the S0 state? And Sf?
b) Analyze the state space corresponding to this problem: i) say what is the minimum and maximum branching factor of the nodes; ii) comment on the existence of cycles and the different ways to avoid them.
c) Propose an admissible heuristic for this problem, and justify why it is admissible
d) Is A * with elimination of repeated states optimal with the proposed heuristic?
e) Would A * be without elimination of repeated states optimal with the proposed heuristic?
f) Develop the search tree using the algorithm A * **WITH** elimination of repeated states. For each node, indicate the order of generation and the order of expansion and the cost values **g + h = f**. In case of ties, the nodes that were generated first are expanded. It is requested to expand the tree until a solution is found or when **5 nodes have been expanded** (not generated). You choose the **order** of expansion and **indicate it** and be consistent with that order throughout the generation of the tree.

2. Search with A*. Consider the following directed graph in which we wish to find the least cost path between A and E using informed search:



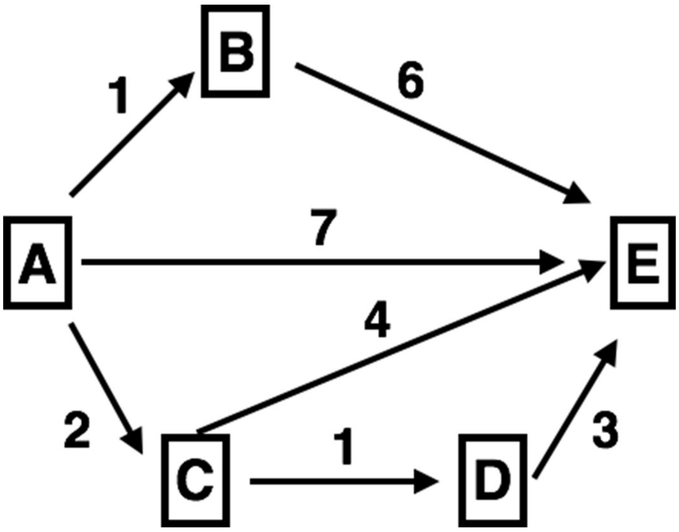
Each transition is labeled by its cost.

(i) Define a monotonic heuristic that is neither trivial (all zeros) nor equal to h *. Demonstrate that it is monotonic.

(ii) Using this heuristic, perform A * search on this graph to find a path from node **A** to node **E** using graph search (that is, eliminating repeated states). The successors of a node are generated in alphabetical order. All other things being equal, the nodes that were generated first are scanned first. Indicate in the search tree the value of the g, h and f values of each node and the order in which the nodes are expanded, including the repeated states.

(iii) Has this search found an optimal (least cost) path? Does this strategy with the proposed heuristic guarantee to find an optimal path? Why?

3. **A* Search**. Consider the following directed graph in which we wish to find the least cost path between A and E using informed search:



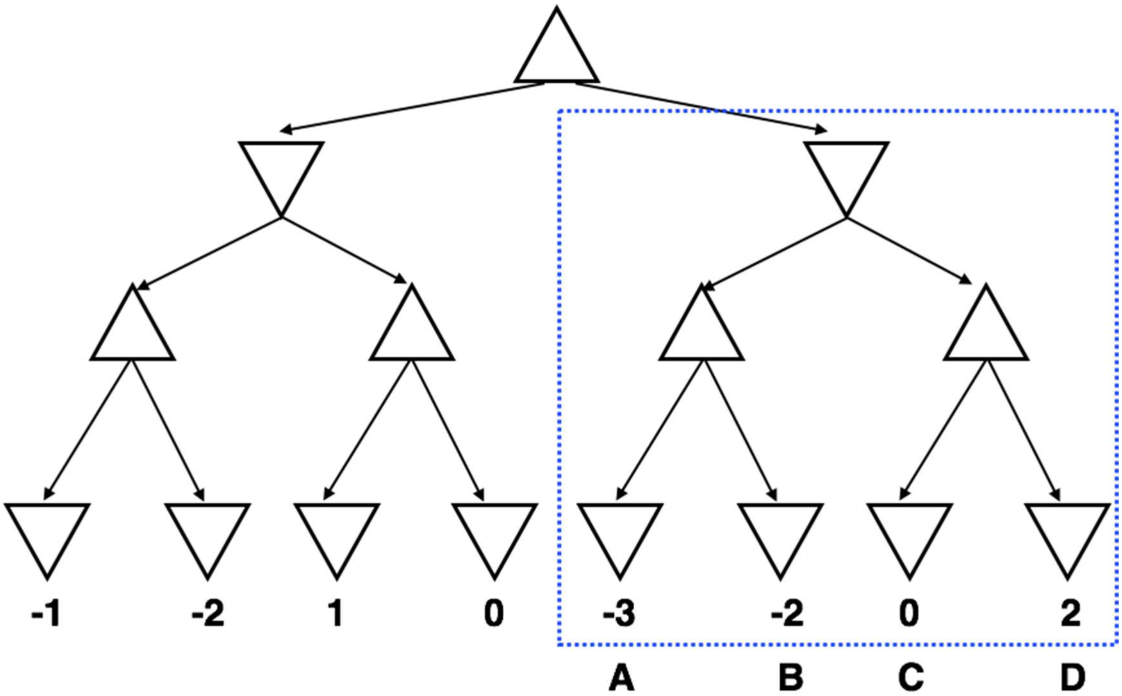
Each transition is labeled by its cost.

(i) Define an admissible and **non monotonic** heuristic. Demonstrate that it is admissible and not monotonic.

(ii) Using this heuristic, perform A* search on this graph to find a path from node **A** to node **E** using graph search (that is, eliminating repeated states). The successors of a node are generated in alphabetical order. All other things being equal, the nodes that were generated first are scanned first. Indicate in the search tree the value of the g, h and f values of each node and the order in which the nodes are expanded, including the repeated states.

(iii) Has this search found an optimal (least cost) path? Does A* generally guarantee, with elimination of repeated states and an admissible and non-monotonic heuristic, to find an optimal path?

4. Adversarial search. Consider the following game tree corresponding to a 0-sum game, where MAX is the player which has now the turn:



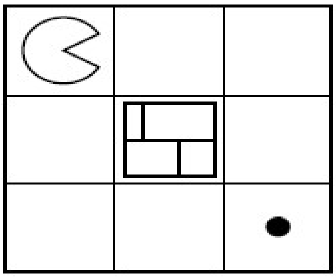
a) Calculate the minimax values of each node in the tree according to the minimax algorithm. Mark with "MIN" or "MAX" each level. What is the optimal movement for MAX, based on the values calculated by the minimax algorithm?

b) Repeat the previous analysis now using minimax with alpha-beta pruning. Draw the corresponding tree clearly indicating the steps of the algorithm, enumerating and labeling each of them with the relevant information, and clearly indicating the pruning steps. Assume that nodes at the same level are generated from left to right. What would be the minimax value assigned to the root? Based on the new analysis, what move should the player who has the turn make?

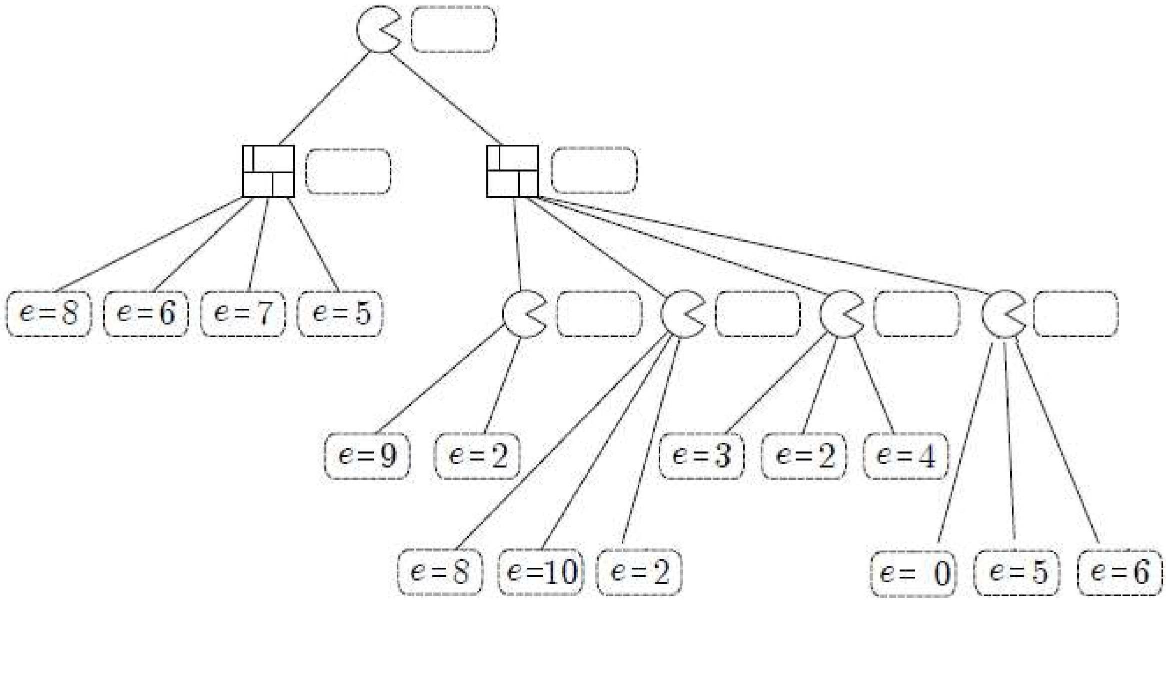
c) What values should the terminal nodes **A**, **B**, **C** y **D** have so that no pruning occurs in the part of the tree contained in the dashed box? You do not need to calculate all possible values, just give an example of these values where no pruning occurs. Explain your answer.

5. **Adversarial search.** In the game of Surrealist Pacman, Pacman plays against a moving wall. On Pacman's turn, Pacman must move in one of the four cardinal directions, and must move into an unoccupied square. On the wall's turn, the wall must move in one of the four cardinal directions, and must move into an unoccupied square. The wall cannot move into a dot-containing square. Staying still is not allowed by either player. Pacman's score is always equal to the number of dots he has eaten.

The first game begins in the configuration shown below. Pacman moves first.



a) Draw a game tree with one move for each player. Draw only the legal moves.
b) Indicate at each level if it corresponds to MIN or MAX.
c) According to the depth-limited game tree you drew above what is the value of the game? Use Pacman's score as your evaluation function.
d) If we were to consider a game tree with ten moves for each player (rather than just one), what would be the value of the game as computed by minimax?
e) A second game is played on a more complicated board. A partial game tree is drawn, and leaf nodes have been scored using an (unknown) evaluation function e. In the dashed boxes, fill in the values of all internal nodes using the minimax algorithm.
f) Cross off any nodes that are not evaluated when using alpha-beta pruning



6. Alfa-beta pruning (1.6 points). Use the minimax algorithm with alpha-beta pruning to:

a. Determine the minimax value at the root of the tree
b. The optimal play assuming the opponent is optimal.
c. Indicate the order of traversal in the tree, numbering the visited nodes together with the updates of the [α, β] values in each step.
d. Indicate who is MAX and who is MIN
e. Is this a 0-sum game? Justify your answer.
f. Does mimimax guarantee to find the optimal strategy in this game? Justify your answer.