ECS 170: Problem Set 3

January 25, 2017

Your answers should be succinct - our solutions for each written problem are no more than a couple sentences.

For this homework you will need to modify a game trees. Because you are submitting this electronically, we've attached a separate .png of the images. For these questions, please modify the .png (e.g. using a program like MSPaint or one of many online programs). Please put everything into a single .pdf.

Do not submit a photo of handwritten work - we will not grade it.

1. Minimax assumes both players are rational. However, suppose one player is not rational. Our goal is to write a program that makes rational moves against an unrational player (i.e. doesn't necessarily make the optimal move). Without loss of generality, let's say we're MAX, our opponent is MIN, we have identical game trees but we don't necessarily know which move MIN will make.

   a) If our program fully explores the game tree, (i.e. no alpha-beta pruning or evaluation function), would we need to rerun minimax if MIN makes a suboptimal move? Why?

   b) Given a fully explored game tree, if MIN makes a suboptimal move, will MAX get a lower or higher score? Why?

   c) Suppose alpha-beta pruning was used (with no evaluation function). Would we need to rerun it if MIN makes a suboptimal move? Why?

   d) Suppose we use minimax without an evaluation function. However, our program is constrained on how much time it can take to explore the game tree, so our program will instead focus on exploring branches of the game tree that are likely to be reached based on some prior knowledge of what moves MIN is likely to make. How might this be problematic if our prior knowledge is incorrect?

2. For a node n, give a concise intuition behind what n's associated α and β represent while alpha-beta pruning is running.

3. Consider the following min-max game tree. All the leaf nodes are terminal nodes whose values are the numbers beneath it.

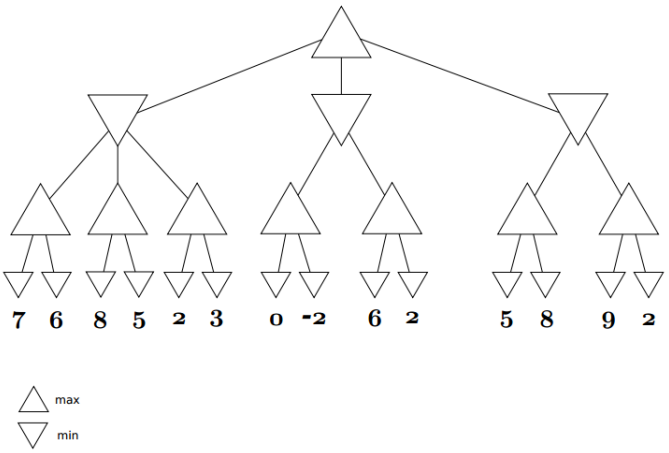
a) No. Minimax is done from bottom to top. Because MAX's moves are always rational and the whole tree(every node) is explored, that guarantees whatever move MIN makes, MAX will always make the best (most beneficial to himself) move.

b) If MIN makes a suboptimal move, MAX will get an even higher score. This is because the nash equilibrium (w/ respect to MAX) guarantees that MAX will get at least the amount of payoff if both player act perfectly. If not MAX will get more.

c) Yes we do.
For example, on MIN's turn, given two states with values a and b where a < b. If being rational, MIN will choose a over b. Therefore, the algorithm will pruning the b branch. However, if MIN's suboptimal move chooses b, then the pruning algorithm needs to be rerun for making decisions up the tree.

d) If our prior knowledge isn't correct and MIN makes an unpredicted move, we will need to expand some other node. Following that expansion, we would need to expand more nodes (more search space), and because the prior knowledge is not correct, we don't know how to prune the tree. There would be more search space and also the time constraint might fail.

2)Suppose n is MAX, then alpha records its current best choice and determines if the MIN choices down the tree needs to be pruned or not. Beta is from its parent node, it determines if n itself still needs to be expanded or not. The situation is reversed when n is MIN.
In short, if n is MAX, alpha is the lower bound on its children and beta is the upper bound on itself.
If n is MIN, alpha is the lower bound on itself and beta is the upper bound on its children.

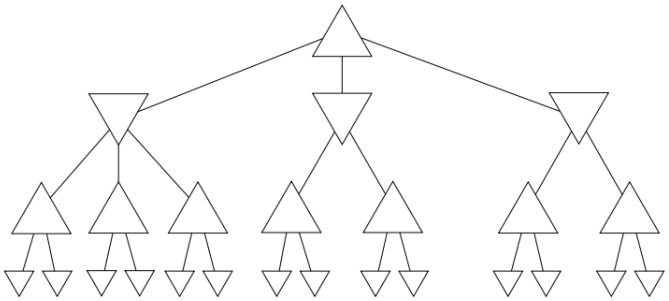


a) Fill out the tree by manually running the minimax algorithm (modify the included .png). i.e. your submission should have the single value associated with each nonterminal node when minimax finishes.

b) Fill out the same tree by manually running alpha-beta pruning where nodes are expanded left to right (again, modify the attached .png). i.e. your submission should have the α and β values associated with each visited, nonterminal node when the algorithm finishes. To make things concise, please use the format $[\alpha, \beta]$. Mark any nodes that would not be visited with an X (including terminal nodes).

c) How would you reorder the first moves that max can make in order to maximize the number of nodes pruned by alpha-beta pruning? Redraw the tree using this move ordering, filling in α and β values and marking nodes that would not be visited with an X. Assume all other nodes are still expanded from left to right.

4. Consider the following empty min-max game tree



For this problem you will modify the game tree such that alpha-beta pruning will not be able to prune any branches. Modifying the attached blank game tree, fill in the values you would use in the terminal nodes and fill in the game tree with α and β values as in the previous problem. For the terminal node values you must use the integers 1-14 with no repeats.