Homework 3

October 13, 2018

1 Homework #3

Problem 1: Consider a genetic algorithm problem having the following properties. Each state is a string of 6 digits, so a state can be any string from "000000" to "999999". The initial population is a set of states {s1,...,sn}, where n is a constant. Reproduction is done by selecting two states s and t from the population, and creating a new state that contains the first i digits of s followed by the last 6 – i digits of t, where i is a random number in {0,1,...,6}. Mutation is not allowed.

(a) How many different possible states are there?

10^6

(b) If p is an initial population, let D(p) be the set of all possible descendants of p. What is the maximum possible size of D(p)?

(Largest Range of Population)^6

(c) Find an initial population p such that D(p) includes every possible state. The size of p should be as small as possible.

{s1,s2}

(d) To create p, suppose we choose 10 states chosen randomly and independently (hence there may be duplication) from the set of all possible states. What is the probability that D(p) includes all possible states?

9/10

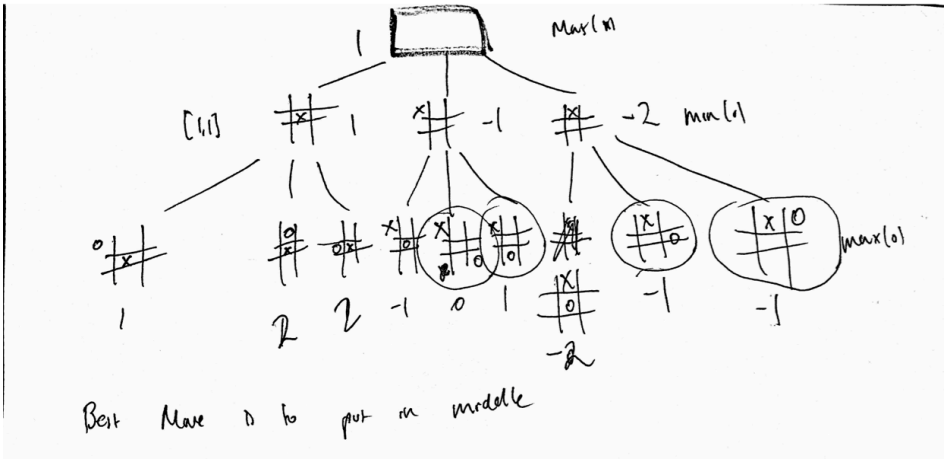
5.9 This problem exercises the basic concepts of game playing, using tic-tac-toe (noughts and crosses) as an example. We define Xn as the number of rows, columns, or diagonals with exactly n X's and no O's. Similarly, On is the number of rows, columns, or diagonals with just n O's. The utility function assigns +1 to any position with X3 = 1 and 1 to any position with O3 = 1. All other terminal positions have utility 0. For nonterminal positions, we use a linear evaluation function defined as Eval(s) = 3X2(s)+X1(s)(3O2(s)+O1(s)).

a. Approximately how many possible games of tic-tac-toe are there?

$$9! = 362,880$$

b. Show the whole game tree starting from an empty board down to depth 2 (i.e., one X and one O on the board), taking symmetry into account.
c. Mark on your tree the evaluations of all the positions at depth 2.
d. Using the minimax algorithm, mark on your tree the backed-up values for the positions at depths 1 and 0, and use those values to choose the best starting move.

e. Circle the nodes at depth 2 that would not be evaluated if alpha–beta pruning were applied, assuming the nodes are generated in the optimal order for alpha–beta pruning.



5.10 Consider the family of generalized tic-tac-toe games, defined as follows. Each partic- ular game is specified by a set S of squares and a collection W of winning positions. Each winning position is a subset of S. For example, in standard tic-tac-toe, S is a set of 9 squares and W is a collection of 8 subsets of W: the three rows, the three columns, and the two diag- onals. In other respects, the game is identical to standard tic-tac-toe. Starting from an empty board, players alternate placing their marks on an empty square. A player who marks every square in a winning position wins the game. It is a tie if all squares are marked and neither player has won.

a. Let N = |S|, the number of squares. Give an upper bound on the number of nodes in the complete game tree for generalized tic-tac-toe as a function of N.

Upper Bound =

$$\sum_{i=0}^{N}(N-i)^i$$

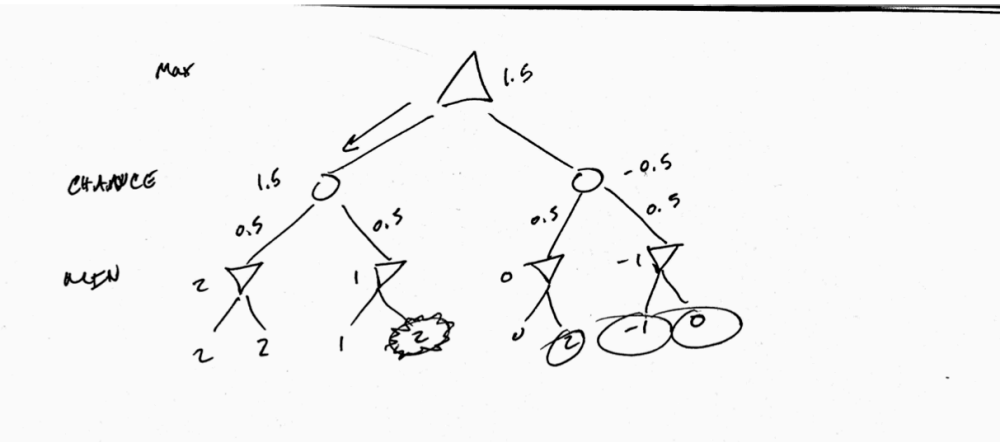
b. Give a lower bound on the size of the game tree for the worst case, where W = { }.

If the collection of winning positions is empty, then we can assume that in order for a state to be a terminal position, all of it's square must be filled.

Lower Bound:

$$N!$$

d. Assume that it is possible to generate a new board and check whether it is a winning position in 100N machine instructions and assume a 2 gigahertz processor. Ignore memory limitations. Using your estimate in (a), roughly how large a game tree can be completely solved by alpha–beta in a second of CPU time? a minute? an hour?



hw3_6

5.16 This question considers pruning in games with chance nodes. Figure 5.19 shows the complete game tree for a trivial game. Assume that the leaf nodes are to be evaluated in left- to-right order, and that before a leaf node is evaluated, we know nothing about its value—the range of possible values is to .

a. Copy the figure, mark the value of all the internal nodes, and indicate the best move at the root with an arrow.

b. Given the values of the first six leaves, do we need to evaluate the seventh and eighth leaves? Given the values of the first seven leaves, do we need to evaluate the eighth leaf? Explain your answers.

If we do not know the bounds of the utility of the tree then we need to evaluate all leaves. This is because the chance nodes won't have any bounds and it's ability to select children to expand is unhindered.

c. Suppose the leaf node values are known to lie between –2 and 2 inclusive. After the first two leaves are evaluated, what is the value range for the left-hand chance node?

$$[-2, 2]$$

d. Circle all the leaves that need not be evaluated under the assumption in (c).