Alpha-Beta Pruning

Background Concepts

Minimax algorithm performs depth first exploration of the game tree. It computes minimax decision from the current state. It uses a simple recursive computation of the minimax values of each successor state. The recursion proceeds all the way down to the leaves and then the minimax values are backed up through the tree. The time complexity of minimax algorithm is b^m where b represents the number of actions for each state and m denotes the depth of the tree.

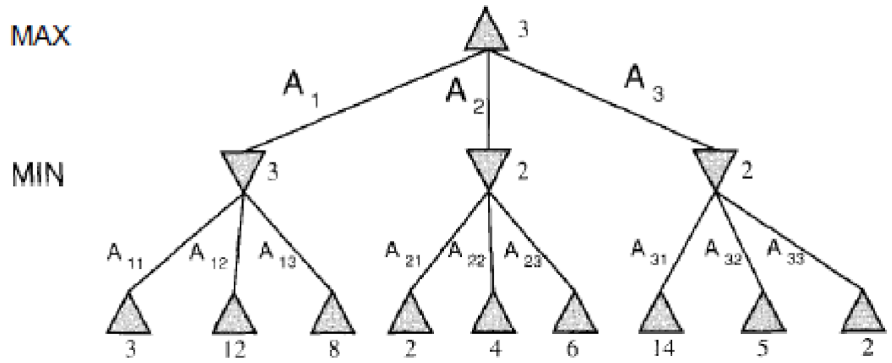

Figure 1.1

Figure 1.1 traces the tree for two players. It recurses down to the left bottom three nodes and MIN chooses minimum value 3 from the leaves 3,12,8 and backwards up this value to left most node. Similar process will be applied for middle node (2) and right most node (2). Finally, the maximum value of 3 is taken from 3, 2, 2 for the root node. Time-complexity for this scenario is 3^2 = 9.

Alpha-Beta pruning technique reduces the number of comparisons of minimax algorithm. It prunes away the branches which cannot influence the final result. Following are the 2 parameters which describe the bounds for backed-up values of the tree.

α = the highest-value discovered so far at any choice point along the path for MAX.

β= the lowest-value discovered so far at any choice point along the path for MIN.

Note that ***Alpha-beta*** search updates **α** and **β** as it goes along and prunes the remaining branches at a node (i.e., terminates depth first call) as soon as the value of current node is known to be worse than the current alpha or beta for **MAX** or **MIN**, respectively. Figure 1.2 depicts the scenario after applying *alpha-beta* pruning strategy in the problem simulated in Figure 1.1. Time-complexity for this scenario is **7**.

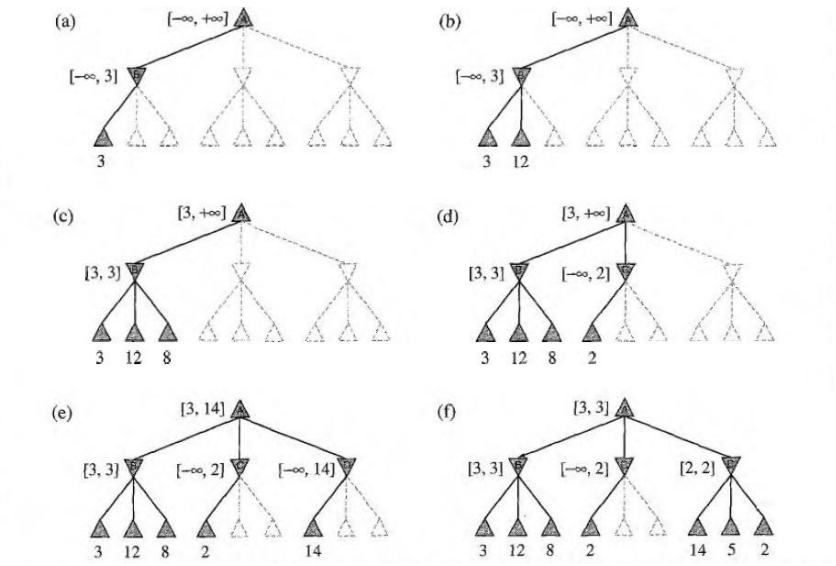

Figure 1.2

Problem

Assume Arko and Riyad are two friends. Riyad is engaged in collecting funds for the upcoming fresher's orientation party. Map the above mentioned minimax algorithm to solve the problem. Assume each of them gets fair chance to optimize the amount. For example, if Arko gets 2 chances to minimize the amount then Riyad also gets the same number of chances in order to maximize the amount. Arko or Riyad has to select 1 note from 3 notes (given) at a time. Sample input and output is provided below.

Sample Input

1	#Number of turns for Riyad [Assume both of them get equal chance, this should determine the depth of the tree]
3	#Number of notes from which the choice has to be made at certain time (Branches per each node)
1 20	#Minimum and Maximum value for the range of notes. (Value of leaf nodes)

Sample Output

Depth:2	=>2*1
Branch:3	=>Given!
Terminal States (Leaf Nodes): 9	=>3^2
Maximum amount: 3	=>Maximum amount collected by Riyad

Comparisons: 9 Comparisons: 7	=>Before Alpha-beta Pruning =>After Alpha-beta Pruning

Reading Materials

Artificial Intelligence, A modern approach [Norvig, Russel], 2nd/3rd Edition