Research Review

My reading material: **Multi-player Alpha-Beta Pruning**, Richard E. Korf

Overview

This paper proposes a generalized algorithm of minimax search with alpha-beta pruning algorithm - **Multi-player alpha-beta pruning**. This algorithm is based on **maxn search algorithm** which is the generalized multi-player version of two-player **minimax search algorithm**. The paper come up with that if an upper bound on the sum pf the evaluations for each players and a lower bound on each individual evaluation can be found, the shallow alpha-beta pruning will be possible. However, deep pruning is not suit for multi-player game in all the cases except two-player version.

Techniques

Minimax Search Algorithm

At first, the author introduces the two-player version of the minimax search with alpha-beta pruning algorithm. In this case, deep pruning can be applied to the searching procedure and it refers to pruning a node based on a bound inherited from its great-grandparent or any more distant ancestor. However, deep pruning does not generalize to more than two players.

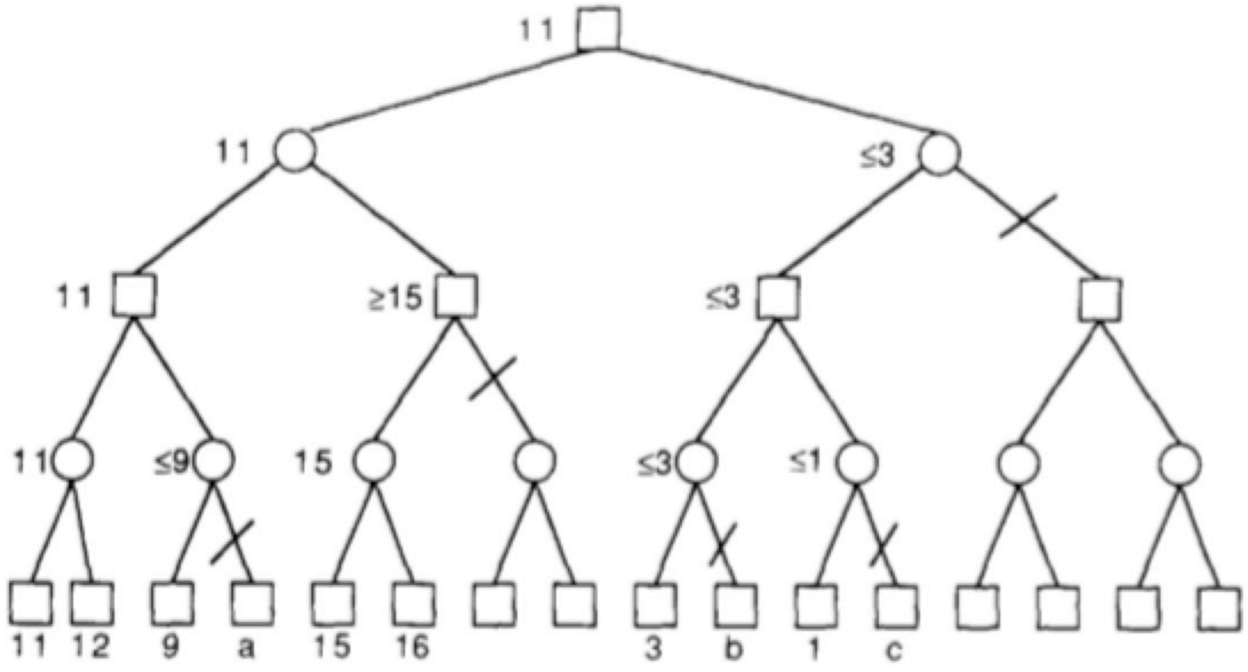

Fig. 1. Two-player alpha-beta pruning.

Maxn Search Algorithm

Then, the author introduces the generalized multi-player version of two-player minimax search algorithm - maxn search algorithm which was proposed by Luckhardt and Irani. This algorithm estimates a node by maximize the corresponding component of active player in value tuples returned by its successors and choose the move whoese corresponding compent of active player in the value tuple is a maxmum.

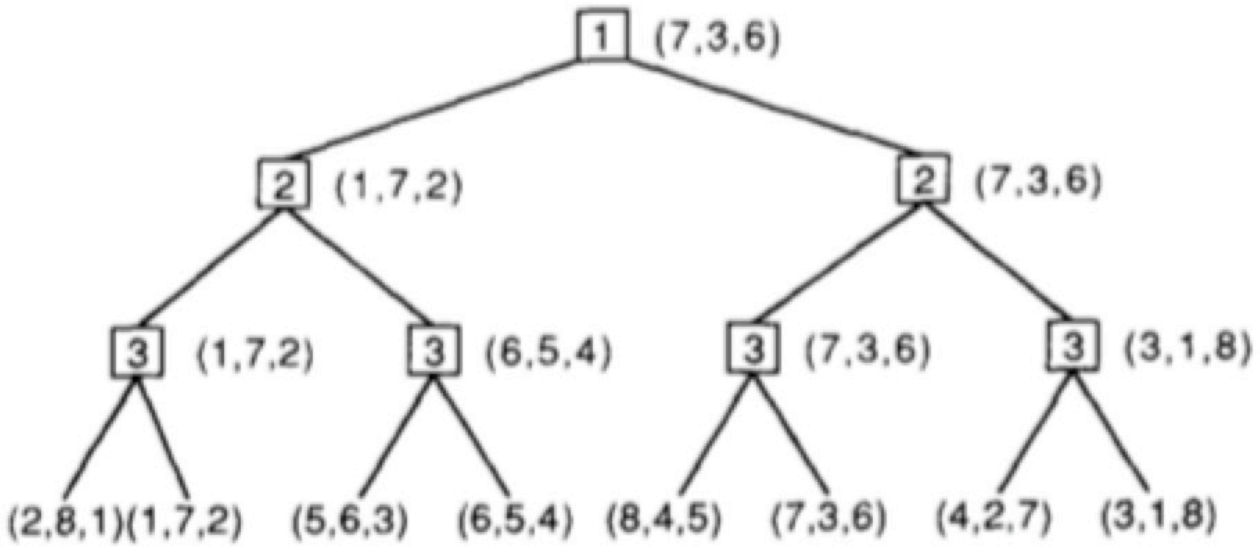

Fig. 2. Three-player maxn game tree.

Multi-player Alpha-Beta Pruning

After that, the author proposes the alpha-beta pruning in multi-player games, which consist of **Immediate Pruning** and **Shallow Pruning**. Alpha-beta in multi-player games is possible to be applied to maxn search algotithm, when both an upper bound on the sum of all components of a tuple and a lower bound on the values of each component are found. And this condition is the basic assumption of **Multi-player Alpha-Beta Pruning**.

- Immediate Pruning
  Immediate Pruning is the simplest kind of pruning which occurs when a player is to move and, the corresponding component of the tuple value of one of its successors equals to the upper bound on the sum of all components. And in this case, all remaining successors can be pruned.
- Shallow Pruning
  Shallow Pruning is a more complex version of traditional two-player alpah-beta pruning. In general, if the correspond component of a player in the value tuple of a successor is higher than SUM minus the correspond component of its parent in the value tuple, the remaining successors can be pruned.

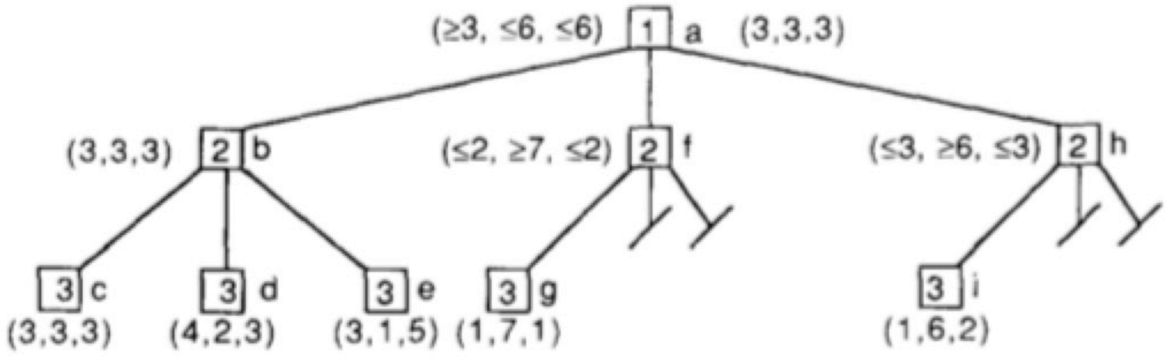

Fig. 3. Shallow pruning in three-player game tree.

This procedure can be written in a python-like pseudo code:

```
def Shallow(Node, Player, Bound):
    if Node is terminal:
        return Score
    Best = - inf
    for Successor in remaining successors:
        if Best[Player] >= Bound:
            return Best
        Value = Shallow(Successor, next Player, SUM - Best[Player])
        if Value[Player] > Best[Player]:
            Best = Value
    return Best
```

Proofs

Correctness of shallow pruning procedure

Theorem 1. If the evaluation of any position for any player is non-negative, and the sum of all player's evaluations in any given position is less than or equal to sum, and ties are broken in favor of the leftmost node, then Shallow(a, p, sum) = M(a, p), for any node a and player p.

Failure of deep pruning

Even though the value of a successor of a node cannot be the maxn value of the node's parent, it can affect the maxn value of the node's parent.

Optimality of shallow pruning

Theorem 2. Every directional algorithm that computes the maxn value of a game tree with more than two players must evaluate every terminal node evaluated by shallow pruning under the same ordering.

Optimality of alpha-beta

Theorem 3. Every directional minimax search algorithm must evaluate every leaf node evaluated by alpha-beta under the same ordering.

Performances

Best-case performance

The solution to the general recurrence has an asymptotic branching factor of $\frac{1}{2}\ (1 + \sqrt{4b - 3}))$, where b is the branching factor. For large values of b, this approaches $\sqrt{b}$ which is the best-case performance of full two-player alpha-beta pruning.

Average-case performance

Knuth and Moore determined that in the average case, the asymptotic branching factor of two-player shallow pruning is approximately $b/\log b$. They assumed independent, distinct leaf values. Under the average-case model described in the paper, the asymptotic branching factor of shallow pruning with more than two players is simply b, the brute-force branching factor.