DBCY Publication
Journal of Information Technology and Society
Volume: 01 Issue: 01 June 2021 Page No.32-37
ISSN: XXX-XXXX

Game Optimization Using Minimax and Alpha-Beta Pruning

(Making the Optimal Move)

R. Manimegalai[1], A. Sheeba[2]
[1,2]Department of Computer Science and Engineering, PSG Institute of Technology and Applied Research
Coimbatore, India.

ABSTRACT - Gaming theory is all about the mathematical study of optimizing agents. Ever wondered how a computer can make moves so intelligently in a turn based game. Isn't is fascinating to know that a high level concept of Artificial Intelligence starts with just a simple algorithm and that now everyone can design their game and make the system play optimally. Optimization is a very important concept in gaming theory. Minimax Algorithm is a decision making algorithm which aims in making optimal decisions assuming that the opponent also plays optimally. This work highlights the advantage of minimax algorithm over other gaming algorithms. In other to support minimax algorithm in gaming theory, concepts like Backtracking, Evaluation function and Alpha-Beta pruning play a major role. This work also gives a brief description about the related terminologies. The scope of this work is to make decision rules in various areas like decision theory, game theory, statistics and so on and make optimal gaming an easy concept. Finally in order to substantiate the work, a two-player turn based game of tic-tac-toe is programmed using the concepts of minimax algorithm**.**

KEYWORDS- Alpha-Beta Pruning, Game theory, probability theory, enhancement in gaming, maximization, minimization, Evaluation function, Backtracking, Recursion, Tic-Tac-Toe, Minmax.

I. INTRODUCTION

Game theory is applied in number of fields like business, economics, entertainment, political science and so on. **The Prisoner's Dilemma** was one of the popular and basic game theory strategy. Later, lots of game theory strategies like the matching pennies, deadlock, cournot competition, coordination and so on were invented. Most of these involved only two-player game at the basic level. **Backtracking** [4][2] is one of the most important concept as it contributes in efficient decision making in game theory. Taking the turn based two player games into consideration, one of the efficient algorithm to make optimal move and play optimally is the **Minimax algorithm.** Minimax is a backtracking algorithm used in decision making and in game theory to find the optimal move. In order to make it more efficient and to decrease the time complexity, the concept of **alpha-beta pruning** [9] comes into play. This work combines all these concepts and presents a simple and an efficient code for a two-player turn based game of Tic-Tac-Toe.

II. LITERATURE SURVEY

Michael Saks and Avi Wigderson have made a detail study on Probabilistic Boolean decision trees and the complexity of evaluating game trees [8]. They have made use of the concepts of Boolean decision tree model where the input variables are the leaves and they may take values from a large set and thus the nodes are replaced with MIN/MAX nodes. They have computed the cost with the help of number of leaves probed by the algorithm and have proved the alpha beta pruning as the best method for these cost computations. Abdallah Saffidine, Hilmar Finnsson, Michael Buro have made a study on Alpha-Beta Pruning for Games with Simultaneous Moves [9] in which they have referred alpha-beta pruning as the most powerful and fundamental minimax search improvements. They have also given a validation on how empirical data shows considerable savings in terms of expanded nodes and computation time as compared to the naïve depth first move without and alpha-beta pruning. Xiyu Kang, Yiqi Wang, Yanrui Hu have given a brief description about the minimax

Journal of Information Technology and Society
Volume: 01 Issue: 01 June 2021 Page No.32-37
ISSN: XXX-XXXX

algorithm and the effectiveness of alpha-beta pruning and evaluation function in game optimization in their work Research on different Heuristics for Minimax algorithm insight from Connect-4 game [10]. In order to substantiate the work, they have introduced a self-developed heuristics supported by well demonstrated result.

A Literature survey on Artificial Intelligence by Nishika Gupta deals with the endless exciting new researches on the field of Artificial intelligence. Clear history and advancements of AI starting from AI in Machine learning to AI in Gaming is presented in [11]. Mayanda Oudah, Talal Rahwan, Tawna Crandeall and Jacob W. Crandall [12] have clearly elaborated regarding the artificial intelligence and the role it plays in gaming.

III. BACKTRACKING

Backtracking is an algorithmic technique for solving problems recursively by trying to build solution incrementally [1-3], one piece at a time, removing those that fail to satisfy the constraints of problem at any point. In the game of tic-tac-toe, entries are made accordingly, stepwise. If the move comes out to be unfavorable, then the solution is backtracked and a new of entries are made from that point.

IV. MINIMAX ALGORITHM

4.1 Introduction

Minimax algorithm basically comes from the Minimax Theorem, proposed by John Von Neurmann in the year 1928. Ever wondered how a computer can make intelligent moves in a game. It all narrows to the concept of Artificial intelligence. But what if things are made more easier. We can design powerful games very easily using the minimax algorithm [5-6] and this paper deals with the brief explanation of making gaming easier and more optimized.

4.2 The Algorithm

Minimax or the Minmax is a recursive or backtracking algorithm that helps in choosing the most optimal move in a turn based two player game assuming that the other player also plays optimally. The **game tree** is the collection of all possible moves in that game and it resembles an inverted tree. The leaf node contains all possible moves. The **game state** denotes the present board situation. With every move, the game state changes and the game tree gets updated height wise.

4.3 Terminologies

The main terminologies related to the Minmax algorithm are [6]:

- MAX (Maximizer): Player whose chances of winning is maximized.
- MIN (Minimizer): Player whose chances of loosing is to be maximized.
- Initial State: Denotes the initial state of the game board.
- Terminal State: Denotes the final state of the game board. It can be a win, loss or draw.

4.4 Working

The main idea of this algorithm is to **maximize the chance of winning for the maximizer**. Every node of the game tree is assigned with weights [1]. Higher the weights, higher the chances of winning.

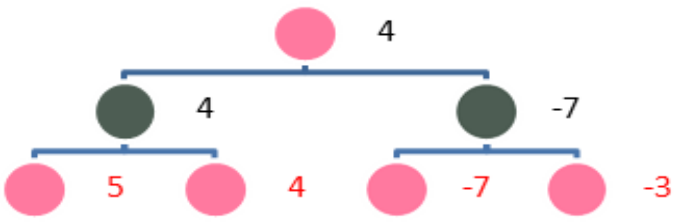


Fig 1. The Optimal Decision Tree

Journal of Information Technology and Society
Volume: 01 Issue: 01 June 2021 Page No.32-37
ISSN: XXX-XXXX

4.5 Applying Minimax in Tic-Tac-Toe

An in-between state of a Tic-Tac-Toe board is taken into consideration. We can see that in fig 2, the current turn is that of the CPU and the CPU is considered to be the maximizer as we need to maximize the chances of CPU winning the game, assuming that the opponent (user) also plays optimally. Now, the CPU can make any of the 3 moves as mentioned in the figure. After the CPU's turn, it's the turn of the user and there are four more further moves that can be made by the user. Now, we backtrack the moves in order to find the most optimal move and it is evident from the figure that the middle board in the 2nd row is the most optimal move for the CPU to win the game. Since it's the most optimal move, that move is assigned a weight of +10 in order to notify that it is a positive move. Similarly, the cases or the moves in which the user wins is marked with a negative weight of -10 which denotes that it's a forbidden move and is not a favorable one.

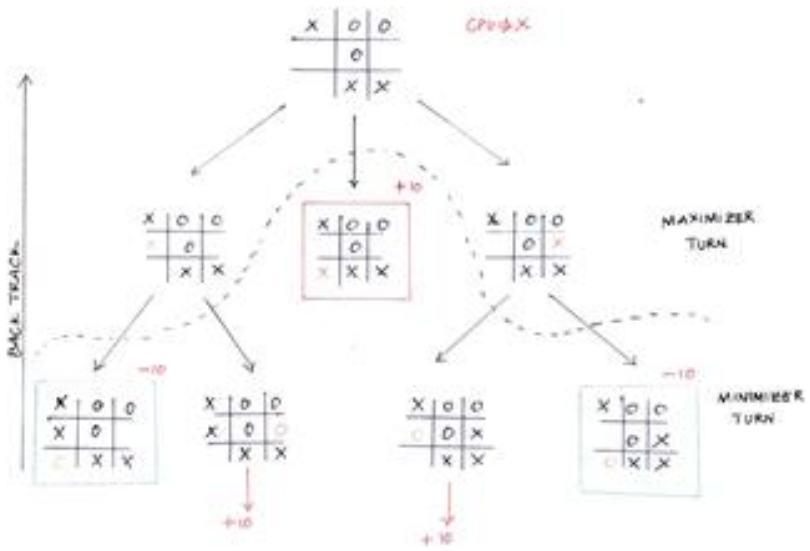


Fig 2. Applying Minimax in Tic-Tac-Toe (process of backtracking)

V. EVALUATION FUNCTION

An evaluation function is also termed as the heuristic evaluation function or the static evaluation function. It is a function used in gaming to estimate the value of goodness of a position in a game tree. The evaluation function evaluates the board and gives the evaluated result. Here, for the game of tic-tac-toe, the evaluation function checks if there is any column-wise, row-wise or diagonal-wise same entry.

VI. ALPHA-BETA PRUNING

Alpha-Beta pruning is is an optimization technique for minimax algorithm [9]. It helps in reducing the computation time by a very huge factor. It allows the user to traverse faster and deeper into the tree. Alpha in alpha-beta pruning denotes the maximizer and the Beta in alpha-beta pruning denotes the minimizer. It aims on decreasing the number of nodes visited for computation. It stops evaluating when at least one possibility has been found that proves the move to be worse than the previously examined move.
There are two cases while implementing alpha-beta pruning.

- Ideal Ordering
- Worst Ordering

6.1 Ideal Ordering

This is the case in which the alpha-beta pruning actually comes into play. In this case, all nodes need not be evaluated. Only particular nodes are evaluated and the alpha-beta pruning is implemented to the fullest.
In case of ideal ordering, the timing complexity comes out to be O(b^(m/2)) where, 'b' is the number of legal moves at each point and 'm' is the depth of the tree.

Journal of Information Technology and Society
Volume: 01 Issue: 01 June 2021 Page No.32-37
ISSN: XXX-XXXX

```
//Minimax function.
int minimax(bool flag)
{
    int max_val=-1000;
    int min_val=1000;
    int i,j;
    int value=1;
    if(eval_cpu()==1)
    {
        return 10;
    }
    else if(eval_user()==1)
    {
        return -10;
    }
    else if(isFull()==1)
    {
        return 0;
    }
    int score[9]={1,1,1,1,1,1,1,1,1};
for(int i=0; i<9; i++)
{
    if(board[i]=='*')
    {
        if(min_val>max_val) //pruning condition
        {
            if(flag==true)
            {
                board[i]='X';
                value=minimax(false);
            }
            else
            {
                board[i]='O';
                value=minimax(true);
            }
            board[i]='*';
            score[i]=value;
        }
    }
}
```

Fig 3. Evaluation function for alpha-beta pruning

6.2 Worst Ordering

In case of worst ordering, there is no chance of eliminating any computation and in this case the concept of alpha-beta pruning does not come into play. Here, the timing complexity is computed as O(b^m) where, 'b' is the number of legal moves at each point and 'm' is the depth of the tree [9]. And thus, it is evidential that how the concept of Alpha-Beta Pruning helps in optimization.

VII. IMPLEMENTATION OF TIC-TAC-TOE

7.1 Evaluation function

Eval_function ( )

if any column-wise entry is found:
  return 1
if any row-wise entry is found:
  return 1
if any diagonal-entry is found:
  return 1
if none of the condition is satisfied:
  return 0

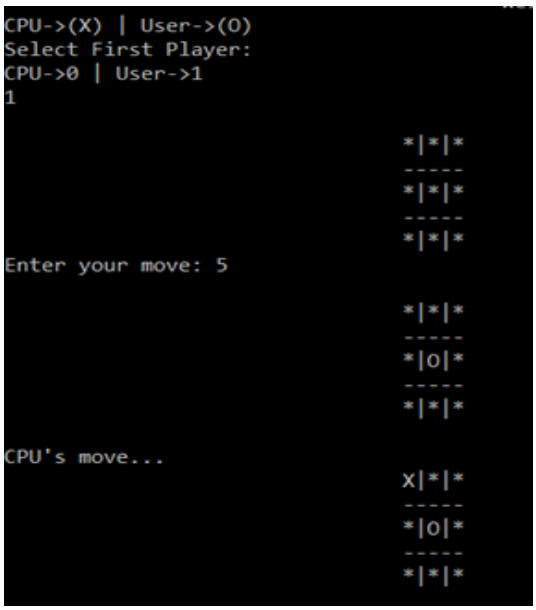
Journal of Information Technology and Society
Volume: 01 Issue: 01 June 2021 Page No.32-37
ISSN: XXX-XXXX

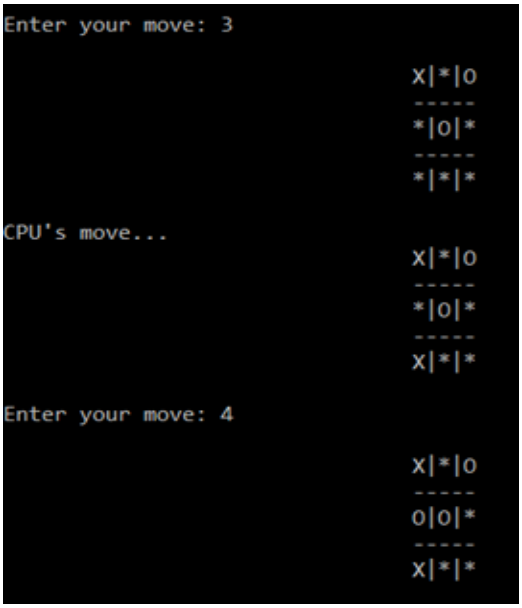
7.2 Minimax Function

Minimax (bool flag)

Set max_val=-1000, min_val=1000, value=1
if(eval_cpu()==1)
  then return a positive weight
else if(eval_user()==1)
  then return a negative weight
else if(isFull()==1)
  then return a neutral weight
initialize score[9] as {1,1,1,1,1,1,1,1,1};
iterate i from 0 to 9
if the board is empty and min_val>max_val then
  check if flag equals True
if true then set ith position of the board to 'X'
call the function minimax again by passing "False" as parameter and store it in a variable value.
otherwise,
set ith position of the board to 'O'
call the function minimax again by passing "True" as parameter and store it in a variable value.
now re assign the ith position of board to '*' and assign value to ith position of score.
If flag holds true then assign -1000 to max_val
Iterate j from 0 to 9
Check if jth index of score > max_val and not equal to 1
  If true, assign value at jth index of score to max_val and set index to j
    return max_val
if flag holds false then assign 1000 to min_val
iterate j from 0 to 9
Check if jth index of score < min_val and not equal to 1
  If true, assign value at jth index of score to min_val and set index to j
    return min_val;

VIII. EXPERIMENTAL RESULTS





Journal of Information Technology and Society
Volume: 01 Issue: 01 June 2021 Page No.32-37
ISSN: XXX-XXXX

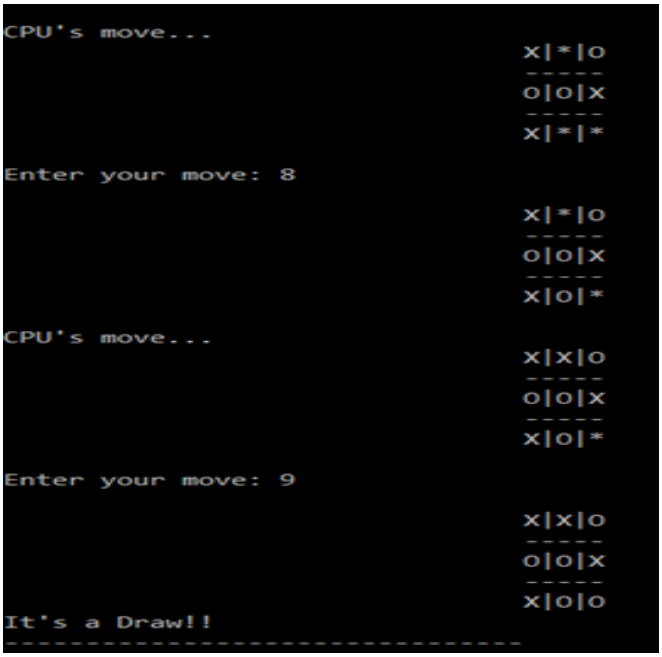


Fig 3. Tic-Tac-Toe Moves using alpha-beta pruning

IX. CONCLUSIONS

Thus, a simplest and most efficient way of developing an optimized game has been explained clearly in this work with the help of minimax algorithm and the concepts related to it. It is also made clear that minimax algorithm is more efficient than the other gaming algorithms like the expectimax algorithm [7] because it is only in case of minimax algorithm, the player is considered to be playing optimally [6] whereas in case of algorithms like expectimax, only the CPU plays optimally without considering the case that the opponent (user) also plays optimally. The study about the concepts of Evaluation function, Alpha-Beta Pruning are also made successfully and it is concluded that alpha-beta pruning decreases the time complexity [9] of the code from O(b^m) to O(b^(m/2)) where b is number of valid moves at that point and m is the depth of the tree.

REFERENCES

[1] Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", Third Edition.
[2] Anany Levitin, "Introduction to the Design and Analysis of Algorithms", Second Edition.
[3] NarasimhaKarumanchi, "Data Structures and Algorithms Made Easy: Data Structures and Algorithmic Puzzles".
[4] Ellis Horowitz, SartajSahini,"Fundamentals of Data structures".
[5]https://towardsdatascience.com/how-a-chess-playing-computer-thinks-about-its-next-move-8f028bd0e7b1
[6]https://en.wikipedia.org/wiki/Minimax
[7]https://www.geeksforgeeks.org/expectimax-algorithm-in-game-theory.
[8] Michael Saks and Avi Wigderson, "Probabilistic Boolean decision trees and the complexity of evaluating game trees", IEEE transactions on 27th annual symposium on Foundations of Computer Science.
[9] Abdallah Saffidine, Hilmar Finnsson, Michael Buro, "Alpha-Beta Pruning for Games with Simultaneous Moves", Proceedings of the 26th AAAI Conference (AAAI 2012).
[10] Xiyu Kang, Yiqi Wang, Yanrui, "Research on different Heuristics for Minimax algorithm insight from Connect-4 game", Proceedings of Journal of Intelligent learning systems and applications, Vol 11 No.2 May 2019.
[11] Nishika Gupta, "A Literature survey on Artificial Intelligence", IRJET 2018 Vol 5- Issue 19.
[12] Mayanda Oudah, Talal Rahwan, Tawna Crandeall and Jacob W. CrandallConference Short Name:WOODSTOCK'18, "How AI Wins Friends and Influences People in Repeated Games with Cheap Talk".