

Analysis of Minimax Algorithm Using Tic-Tac-Toe

Swaminathan B ^a, Vaishali R ^b and subashri T S R ^b

^aProfessor, Dept of CSE, Rajalakshmi Engineering College, Chennai

^bUG Student, Dept of CSE, Rajalakshmi Engineering College, Chennai

Abstract. The game industry has been on exponential growth, has different businesses of varying size, ethos, scope and beyond. Success of these video-games comes from a lot of labor-intensive work by developers. Every little nuance of each character, the objects within a character's environment must be hand-coded. Repetitive work takes up a significant part of development time, which leads to an increase in glitches and logical flaws. Artificial intelligence has been used to simulate human players in software games, provides an opportunity for game developers to create unique experiences and different outcomes for each player. Computer chess players are well-known examples, wherein modern chess programs are trained to defeat best human players. AI based algorithms that can be implemented for games, but a need for optimal solutions is on a rise. We require a comparative analysis of multiple algorithms for understanding the most efficient and ideal one. In our work, through use of a game Tic-Tac-Toe various algorithms will be carried out with its prototype compared in terms of effective rate and optimality.

Keywords. Artificial Intelligence, Comparative Analysis, Optimal, Effective Rate, Minimax

1. Introduction

Human since ancient times have been inextricably intertwined with games and are an essential part of life. A recreational way of socializing as well as imaginative expressionism came out through games. The classification can be done multiple ways such as, Outdoor vs Indoor, Chance vs Strategic, Cooperative vs Non-Cooperative. With the evolution of games and technology together, software games encompass not only traditional indoor ones but also provide outdoor games without having to move from a place. Football, Cricket can be played from the ease of the player's room [1]. For years now, artificial intelligence has been a rising innovation in video games. The video game industry is one of the niches of development where programmers are forced to dedicate a great deal of time to researching what AI is or is not capable of achieving.

¹Swaminathan B, Professor, Dept of CSE, Rajalakshmi Engineering College, Chennai,
Email: swamikb@gmail.com

For game creators who have access to technological limits and many applications inside stories for simulation, the only obstacles to making full use of this advancing technology are the possible time and money constraint [2]. With two teams, tic-tac-toe is a game of paper and pencil, the players take turns numbering the gaps in a grid of 3. In order to win either of the rows must have three marks X or O. It is used as a pedagogical tool to inculcate good sportsmanship, since tic-tac-toe is very easy [3]. It's common to build computer program to play tic-tac-toe perfectly or to list the 765 positions that are essentially different or the 26,830 possible games up to rotations and reflections on this space [4]. In the tic-tac-toe game, a player tries to ensure two cases: Maximize a player's own chances of win. Minimize a block failure. Minimize the chances of winning the adversary, by maximizing profit. The fork or win technique can be used to maximize profits. Block if the opponent has two 'x' or 'o' in a row then block it, or block the opponent's fork. Fork means player must initially build a chance where he can succeed in two directions [5]. Win means if there are two same X or O in a row, then play the third in a row to get three. It is a two-player game.

2. Related Work

Roopali and Deva [6] tic-tac-toe is implemented using the concept of combinational game theory. Using mini-max algorithm this algorithm estimates only one step ahead. An ideal scenario allows a player to quantify all the possibilities to ensure the success. It's a tiny 3x3 game, so the created state space tree will be small. Anurag, Pratul and Kalyanmoy [7] used Genetic Algorithms to discover not one but as many no-loss solutions as possible, so that we can continue and unravel what allows such a strategy not to fail and design new ways to evaluate a solution by MATLAB matrix analysis. Carl^[3] uses AB pruning to solve the Tic-Tac-Toe problem. The research idea was to reduce the created portion of game tree. Plamenka and Milena [8] in their research investigate the efficiency of parallel minimax algorithms for search in a game tree. Tic- Tac-Toe is used in their research to understand the efficiency [9-12]. The suggested parallel computational model exploits tree partitioning at width for each level of the game tree and is based on combination of the parallel algorithmic paradigms. Siby, Kajal and Isac [13] their model is implemented using game theory, minmax and optimal strategy. All possible winning strategies are analyzed in order to ensure the success not only by blocking the other player's success but also ensuring that blocking the opponent will not give more vulnerabilities. They used the discussion by Al-Khateeb which proposed artificial neural networks are suggested to be used as function evaluators for the creation of tic-tac-toe game play strategies [14].

2.1. Framework of Alpha-Beta Pruning Algorithm

Alpha-beta pruning is a search algorithm that seeks to decrease the number of nodes that are evaluated by the minimax algorithm in its search tree . It stops evaluating a

move when at least one possibility has been found that proves the move to be worse than a previously examined move.

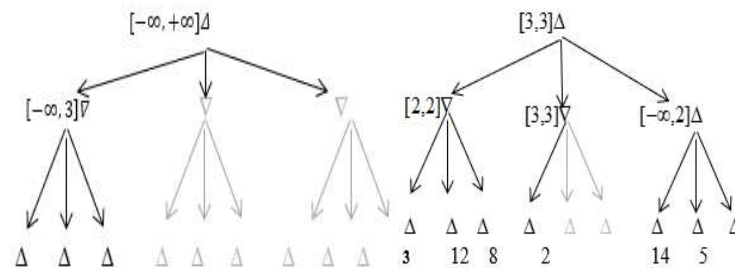


Figure 1. Final Stage AB Pruning

3. Proposed Framework

3.1. Tic-Tac-Toe Brute Force flow

Brute Force Algorithms refers to a programming method that does not include any tricks to improve performance, but depends instead on brute computational power to try out all options until a problem is solved.

3.2. Tic-Tac-Toe Alpha-Beta Flow

Alpha-Beta Minimax algorithm is used to cut the branches of the tree which won't contribute positively to the goal state that needs to be arrived at. Alpha – Beta Pruning allows the space to be used in a compact manner as well as use time efficiently. The purpose of Tic-Tac-Toe is to win multiple games as well as efficient use of the system.

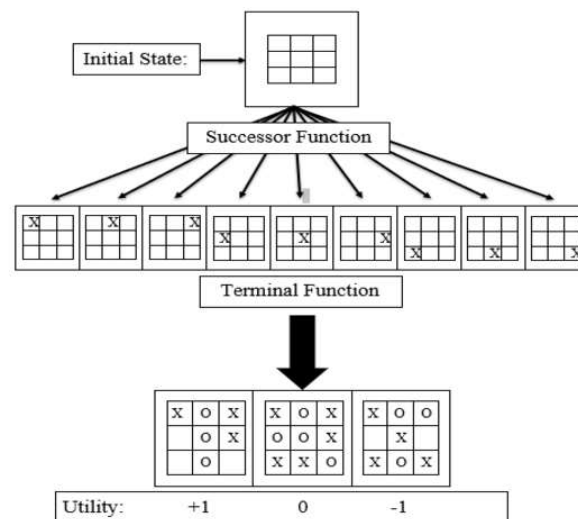


Figure 2. Alpha - Beta Pruning Utility Flow

4. Analysis

Alpha-Beta as presented in the Carl Felstiner's research found that randomness had an insignificant impact in Tic-Tac-Toe with forced steps turned on. In Tic-Tac-Toe there are so many unwanted moves that pruning doesn't do much. Randomness does not benefit without pruning, because there is no chance of tree short-circuiting. When the simulation was performed without the features of forced movement, randomness reduced the number of nodes tested over 1000 trials by about 6-10 per cent. The complexity for the Alpha-Beta Pruning is:

$$O((nM)^N). \quad (1)$$

For understanding the working requirement of Tic-Tac-Toe, we construct a PvC game with Brute force framework. This involves applying the marker at any randomly available square on the 3x3 grid. This not only decreases the chances of the System winning by 80% but has high space complexity. We can understand this by, the analysis of code used. For each loop used the complexity added is: N. For each loop nested within: N is multiplied by outer loop. For each if-else statement complexity added is: 1.

$$\text{The complexity of brute force: } O((N \times 2) \times 10 + 20 * N + N^2). \quad (2)$$

Equation 2 is brute force algorithm time complexity. Execution is recursive execute the same function again and again, it takes more time to reach the goal state. Although the complexity seems to be low in time, when compared to the optimal winning strategies achieved it is rendered useless. The algorithm is required to be not only efficient but also accurate. Improvised AB Algorithm for tic tac toe game Understanding that winning strategies begin with taking the corners, our Algorithm is set such that we take up the ends. Gradually as the player plays, the algorithm cuts down the state space by pruning and time with minimal recursive calls. The complexity calculated is:

$$O(bM) \quad (3)$$

The Equation 3 is improved alpha-beta algorithm time complexity is less time due to the number(b) of time's function(M) is to be executed.

5. Conclusion

The tic tac toe game is implemented using the alpha-beta algorithm of c++ and brute force. The outcome is evaluated with two players, player and machine, based on an actual time setting. Randomized Alpha-Beta and Revamped Alpha-Beta were adopted for game 3x 3. Enhanced Alpha-Beta has more optimal performance dependent on

Alpha-Beta production. In order to test machine and player, the result in the enhanced Alpha-Beta is better based on time complexity and brute force out. Even with a heuristic approach the difficulty of time is high and the condition of target is maximum. The discussion is based only on the complexity of time, which also includes complexity for analytical purposes in the feature work space.

References

- [1] <https://en.wikipedia.org/wiki/Tic-tac-toe>
- [2] Roopali Garg and Deva Prasad Nayak, Game of tic-tac-toe: simulation using min-max algorithm. International Journal of Advanced Research in Computer Science, (2017), Vol.8, No. 7, pp. 1074-1077. DOI: <http://dx.doi.org/10.26483/ijarcs.v8i7.4409>
- [3] Anurag Bhatt, Pratul Varshney and Kalyanmoy Deb .Evolution of No-loss Strategies for the Game of Tic-Tac-Toe.Department of Mechanical Engineering, Indian Institute of Technology Kanpur. <https://www.iitk.ac.in/kangal/papers/k2007002.pdf>
- [4] Carl Felstiner,Alpha-Beta Pruning. <https://www.whitman.edu/Documents/Academics/Mathematics/2019/Felstiner-Guichard.pdf>
- [5] Plamenka Borovska and Milena Lazarova, “Efficiency of Parallel Minimax Algorithm for Game Tree Search”, International Conference on Computer Systems and Technologies, CompSysTech 2007, Rousse,Bulgaria, June 14-15, (2007), DOI: 10.1145/1330598.1330615;
- [6] Siby Samuel, Kajal Mahawar and Isac João França, Improved technique in Tic-Tac-Toe game to minimize the condition of draw using min-max over optimal strategy.International Journal of Scientific& Engineering Research Volume 10, Issue 3, (2019) 663 ISSN 2229-5518
- [7] Using Tic-Tac-Toe for Learning Data Mining Classifications and Evaluations by Chen-Huei Chou <http://www.ijiet.org/papers/314-k010.pdf>
- [8] The Analysis of Alpha Beta Pruning and MTD(f) Algorithm to Determine the Best Algorithm to be Implemented at Connect Four Prototype by Lukas Tommy, Mardi Hardjianto and Nazori Agani; DOI : <https://doi.org/10.1088/1757-899X/190/1/012044>
- [9] Analysis of the alpha-beta pruning algorithm by Samuel H. Fuller John G. Gaschnig and Gillogly; DOI : [https://doi.org/10.1016/S0004-3702\(83\)80020-4](https://doi.org/10.1016/S0004-3702(83)80020-4)
- [10] Al-Khateeb, B., An Evolutionary Tic-Tac-Toe Player.International Journal of Reasoning based Intelligent System, Vol. 4, No. 4, pp.182-185, (2012).
- [11] https://www.researchgate.net/scientific-contributions/70323727_Jr-Chang_Chen
- [12] https://www.researchgate.net/publication/270724220_Job-Level_Alpha-Beta_Search
- [13] <https://intellipaat.com/community/26314/minmax-trees-when-min-can-win-in-two-steps>
- [14] <https://www.codeproject.com/Articles/7344/A-brute-force-search-algo>