Literature Survey of Chess Engines

Anushka Nair
Department of Computer Engineering,
Atharva College of Engineering.
Mumbai, India.

Kanksha Marathe
Department of Computer Engineering,
Atharva College of Engineering.
Mumbai, India.

Prof. Suvarna Pansambal
Department of Computer Engineering,
Atharva College of Engineering.
Mumbai, India.

***Abstract* - Chess has come a long way since making its appearance on a computer and every year there is a new chess engine created that works better than the previously existing ones. But all these engines have one thing in common, they all work on some variant of the MinMax algorithm. The game tree is made shorter and more efficient and ways to play around with the graphical user interface are invented. The search tree is then pruned to save the time that the machine needs to compute the next best move. Thanks to all this, chess engines have come a long way and are now known all around the world to defeat some of the best Grandmasters of the world with their own strategies. But the AI in chess still has a long way to go before it can be declared as the fastest and the best chess playing machine.**

Keywords: Chess, Engine, MinMax Algorithm, Pruning.

I. INTRODUCTION

This paper focuses on not only the software part of the engines discussed here, but also advancements in the hardware and graphical user interface of a chess engine. Since most engines use a variant of the MinMax search tree, it is considered as an efficient algorithm for running the engine. Each variant is then improved using new searching or pruning techniques. The hardware is updated accordingly to help the machine carry out the calculations faster.

The GUI is improved using various tools available. Making the GUI interactive makes the user feel more and more close to playing the actual game but at the convenience of portability. 2D as well as 3D graphics and other media like background sound is added to give the game a little bit of personality and edge over the original orthodox game.

Also, now more and more variants of the game itself are being made to attract the number of players of chess. The Surakarta chess and the hexagonal chess are some of the better known versions that are gaining popularity amongst players of the game. These engines work on the same idea except the chessmen move in ways different to the original chess. The board is also constructed in a different way to give players the feeling of playing an entirely different game.

Each engine is constructed keeping in mind that it should run on the average machine and no extra processing units should be needed unless it is required for computing a very complicated game tree or adding very high definition graphics to it. Also, a lot of games are made purely to be played on a network between two connected client machines, which means players can play against each other or the computer.

II. DEFINITIONS

1. MinMax Algorithm: is a decision based rule used in a game tree for minimizing the loss for a worst case scenario. Originally formulated for two-player zero-sum games like chess, it has evolved into various variants to work in more complex versions of chess as well as other games.
2. Pruning: In a game tree, when the minimum node or the maximum node is already the value it needs to be and further traversing is not required, then that particular branch of the tree is pruned or cut off from the traversal.
3. Surakarta Chess [5]: This is a Chinese version of battle chess where the square chessboard has six horizontal lines and six vertical lines. On the thirty-six crossing points of horizontal lines and vertical lines are chess pieces. All these lines are linked by eight arc lines. Each side to play the game has twelve chess pieces.
4. Hexagonal Chess: In hexagonal chess, the board is in the form of a hexagon with 91 cells. The best known is Gliński's variant, which is a symmetric hexagonal board. The rules and movements of the chessmen vary according to the maker.

III. LITERATURE SURVEY

Murray Campbell et al [1] contributed in the making of this chess engine which was attempted twice because the first time it lost to the chess grandmaster Garry Kasparov in 1996 but later went on to defeat him in 1997. It was the first chess machine to defeat a grandmaster in tournament play. It uses a single chip move generator. Deep Thought 1 and Deep thought 2 were intermediate stepping stones to the Deep Blue, but Deep Thought 2 was played in a lot of public events from 1991-1995. Deep Blue I, which was initially defeated by Kasparov, made it clear that it had certain shortcomings which had to be fulfilled. Thus, a new chip was designed which went from 6400 features to 8000 features. Basically, the chip combines a software search in C language as well as a hardware search on the silicone chip and generates the best move set possible. It implements the null window alpha beta algorithm and even without pruning, it is a very effective search.

Fogel et al [2] made this engine, where the flaws of machine learning are overcome. Since credit assignment system has its flaws, the evolutionary system is more efficient. The evolutionary algorithm uses three artificial neural networks to evaluate the worth of potential positions in the chessboard that are alternate. All three players start with neural networks and play in random variation to evaluate the fittest of them all through the quality of the moves. Then, the fittest players generate the off spring for the next generation. This algorithm is very useful as the evolution of a high-level chess program suggests broader applications to problem solving and also gives us the possibility to attempt solving problems to which known solutions do not exist.

Tong Lai Yu et al [3] used the single threaded chess engine Beowolf in this project. Platform independent SDL libraries are used inorder to make it more efficient so as to incorporate graphics and other media into the game. Beowulf searches a game tree using the Negamax Search algorithm, which is a simply a variation of the minimax search that accounts the zero sum game. The game tree is simplified by Alpha-Beta pruning. The 2D graphics and sound is added with the help of SDL libraries that are written in C/C++. The 3D graphics are implemented with the help of open source tools such as Blender 3D. Thus, this paper establishes the fact that a chess engine with amazing graphics, or any other grid game can be created with the help of open source tools.

D. M. Raif, et al [4] In this paper, the main objective was to make the game more aesthetically appealing by the addition of art constructed by ceramic material. Cartoons are considered the best way to deliver a message across to the game’s audience and this paper converts the usual chess pieces into cartoon ceramic chess pieces for more visual appeal. Every 3D character is a set of 2D images used to construct it according to the planned model and altering its movements. Thus, each piece is given its own shape, size, texture, movement and personality to give the game of chess an edge over the other engines.

Liqun Zhang et al [5] introduce the program of Surakarta chess battle platform in computer game is a network program. It is played on a server by both sides on the client side and the game starts after both sides login successfully. The server has to be a special server since the game itself judges who’s winning and human intervention only proves to slow down the engine. Unlike regular chess, this version of chess follows the rule of moving pieces one square vertically or diagonally to a position that is not occupied by another piece unless it is a capture. Also, the board is represented in a different way. The Surakarta chess is the most efficient way to hold large machine-machine games which is a great way to compare which machine works the best when it comes to a game of chess.

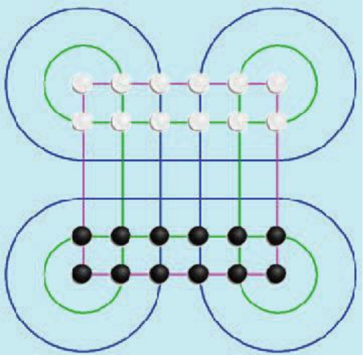

Fig. [6] Surakarta Chess Board

Omid E. David et al [7] put forward the idea that unlike the conventional chess engines, genetic algorithm can be used to skip the brute force and actually improve the AI to such a potential that it can surpass a human at the game. The algorithm starts with a population, and with each generation, the fitness improves, which improves the chances of a computer winning against a human. In the initial stages of chess engines, only 1-ply searches were possible, but due to the algorithm’s evolutionary nature, now it is possible to make tens of plies searches in a tournament game. This is tested out in this paper with the help of chess as well as checkers.

Nathaporn Karnjanapoomi et al [8] in this paper brings out the possibility of throwing away the orthodox idea of a two player chess games and allowing 3 players to play in the same game. Any player that captures another payer’s King, is declared a winner in this game. Thus, this game not only gives an opposing edge, but also creates the possibility of cooperation of two players to defeat the third one. This is implemented by the use of Nondominated Adversarial Search algorithm which makes it necessary for the player to select the quantity of good moves over quality of moves.

Rahul A R et al [9] describes in this engine, that most game engines are beat by Grandmasters very easily since the engine does not calculate the future feects of a certain move made currently. This engine thus, evaluates the position evaluation and returns the material difference between the player positions. This is important as in a game of chess, sometimes a side willingly sacrifices a coin in order to layout a more elaborate trap. For the computer to this, it must evaluate the position and the moves in advance and niching and multi-niching is used to defeat the player. This engine also allows the players to use advance tactics such as castling, which has been possible only in the real life board game till recently. Thus, this engine has the closest, most efficient AI for the game of chess.

Diego Real and Alan Blair [10] made the Duchess chess engine work with efficiency. In this paper, the duchess chess engine is explained which allows upto six players in each game and also adds an additional coin of the ‘Duchess’, which can perform the Bishop and Knight’s moves. This engine also adds one additional player i.e. the computer, for any additional help that is required as well as to replace a player if they lose the game. This engine was initially impossible to run because of the lack of efficient technology but is now possible with the help of the Tree Strap Algorithm.The tree search is sped up by searching cumulatively, i.e., only searching the board where the changes have occurred. This helps the game run fast despite having six players simultaneously.

Wu Gui et al [11] in this paper, has tried to improve the game search algorithm to make it more practical for the game engines created now. This Chinese chess engine solves the frequently faced problems in the search tree. The board is represented as a 16*16 matrix, thus solving it as a 256 byte system. This helps the system evaluate the board in one dimensional arrays quickly and efficiently. Multiple search algorithms are then compared and evaluated to see which one works thebest and is practical to be implemented on all kinds of host systems. The rules of Chinese chess are used for these experiments.

IV. DISCUSSION

Each system has the flaw of having a relatively lower AI in the beginning but evolving as and when it loses a game. Then the algorithm creates a new population that is better than the previous and avoids the mistakes that the engine made initially. Learning from each iteration, the algorithm improves itself in game play to reach the levels of a grandmaster and is then used in public events to determine whether the engine is successful or not. Most systems also use databases with grandmaster entries, following which the machine makes moves that previously winning grandmasters have made. This is the closest an engine has got to depicting human moves in the game depending on the move’s success rate.

V. CONCLUSION

Since the algorithm side has a lot of research scope, the focus is on the graphics end on how to make the chess game visually appealing enough that players find it unique to the real life chess board. The GUI can be improved using light bit boards and the chessmen can be reinvented into whatever style intended for the game. Thus, this paper gives the new graphic solution of implementing Wizard’s Chess from the Harry Potter Series in the 2D form. This game is made with the intention of encouraging more and more players to play chess as the number of chess players has greatly decreased in the past decade. The engine will use a GUI made of arrays instead of the otherwise used bit boards as it makes the job of understanding the code easier and to improve the algorithm at a later time.

VI. REFERENCES

[1] Murray Campbell, A. Joseph Hoane Jr. b, Feng-hsiung Hsu, “Deep Blue”, Elsevier 2002, Artificial Intelligence 134 (2002) 57–83

[2] *Fogel, Hays, Hahn, and Quon, “*A Self-Learning Evolutionary Chess Program*”,* Proceedings of the IEEE, Vol. 92, no. 12, December 2004.

[3] Tong Lai Yu, “Chess Gaming and Graphics using Open-Source Tools”, IEEE Computer Society, 2009.

[4] D. M. Raif, R. Anwar, N. A. Ahmad, Z. Zakaria and M. F. A. Jalil, “Revision on Cartoon Character Integrate with Chess Concept for Industrial Ceramic Artware”, 2013 IEEE Business Engineering and Industrial Applications Colloquium (BEIAC).

[5] Liqun Zhang, Lili Ding, Zhenlai Li, “The Design of Surakarta Chess Battle Platform in Computer Game”, 2013 25th Chinese Control and Decision Conference (CCDC), IEEE paper.

[6] Figure of the Surakarta Chess Board taken from the “The Design of Surakarta Chess Battle Platform in Computer Game” paper.

[7] Omid E. David, H. Jaap van den Herik, Moshe Koppel, and Nathan S. Netanyahu, “Genetic Algorithms for Evolving Computer Chess Programs”, IEEE Transactions On Evolutionary Computation, Vol. 18, No. 5, October 2014.

[8] Nathaporn Karnjanapoomi, Pongpol Pramanpol, Voratep Lertratsamewong, Torsakuln Chacavarnkitkuln, Vazuthorn Rattanajongjittakorn and Chaicharn Thavaravej, “A Nondominated Adversarial Search Algorithm for a Three-player Chess Game”, 2013 International Computer Science and Engineering Conference (ICSEC).

[9] Rahul A R and G Srinivasaraghavan, “Phoenix: A Self-Optimizing Chess Engine”, 2015 IEEE International Conference on Computational Intelligence and Communication Networks.

[10] Diogo Real and Alan Blair, “Learning a Multi-Player Chess Game with TreeStrap”, 2016 IEEE Congress on Evolutionary Computation.

[11] Wu Gui and Tao Jun, “Chinese Chess Algorithm Design and Implementation in the Computer Games”, Proceedings of the 35th Chinese Control Conference July 27-29, 2016, Chengdu, China.