International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 04 | April - 2023 Impact Factor: 8.176 ISSN: 2582-3930

Chess Engine Based on AI: A Survey

Rahul Gaikwad[1], Mahesh Namdas[2], Vaishnavi Olekar[3], Prof.Shaila Pawar[4]

[1,2,3,4] *Department of Information Technology, A C Patil College of Engineering*

Abstract - Researchers have been experimenting with several methods to create chess engines that can surpass human players, which has led to a lot of advancement in the field of chess AI in recent years. Neural networks, genetic algorithms, and machine learning are a few of the common methods. Chess engines that use self-play, automatic feature extraction, pattern recognition, and genetic player learning have been created using these methods. Researchers have also looked into various chess AI-related topics, like enhancing the game's visual appeal, creating multi-player engines, and even creating automated chess commentary. Although each strategy has its advantages and disadvantages, the area of chess AI is still developing as researchers work to create ever-more complex engines that can beat this game.

***Key Words*:** Chess, Chess AI, Chess Engine, Game ,Chess Algorithm, Negamax Algorithm, Player VS Player Game, Play with Computer,Play against AI, Play against Chess Engine.

1.INTRODUCTION

The strategy board game of chess is played between two players on a checkered board with 64 squares arranged in an 8 by 8 grid. Millions of people play chess throughout the world, and it is believed to have descended from the Indian game chaturanga before the seventh century. There are various schools of thought in chess composition, and each places a different emphasis on the difficulty of the issues. Chess studies first began to gain prominence in the 19th century. They are recognised for being difficult puzzles with sophisticated calculations and strategic patterns. They therefore provide a helpful starting point for research on artificial intelligence.

Artificial intelligence (AI), a branch of computer science, is growing swiftly and is increasingly more accessible. The social and technological impact is increasing tremendously. The top companies in the digital space, such as Google, Facebook, Amazon, and Microsoft, have been offering services and goods based on AI. This technique has been used to develop a wide range of applications, including those for video games, financial services, and machine autonomy.

One way to analyse how AI has developed is to look at the evolution of chess engines over time. Since 1997, when they beat Garry Kasparov, who was at the time the top human chess player, until the present, when Google DeepMind developed a neural network that played the game and bested the best chess engine to date after just four hours of training. It is unlikely that chess will ever be mastered because it is a difficult game. Making an artificial intelligence that can understand and play the game well is challenging.

Human players have been enthralled by the strategic board game of chess for ages. Computer programmes have evolved into formidable opponents with the development of artificial intelligence (AI), capable of competing with and even defeating the best human players in the world. We will evaluate and contrast the various AI-based chess engines that are currently on the market in this survey study. We'll talk about the development of these engines, the state of the art today, and their various design philosophies. We will also look at each engine's capability and performance in various scenarios.

There are many well known chess engines that are popular amongst the Professional Chess Players.Some of them are Stockfish , Komodo , Houdini. There is one which is not really in used but is well known as it is the first ever strong engine made, its name is Deep Blue.

Stockfish:

Stockfish is a free, open-source chess engine that uses alpha-beta search with various advanced algorithms to find the best move in a given position. It has been consistently ranked as one of the strongest chess engines and has won several computer chess championships. The advantages of using Stockfish include its accuracy, depth of analysis, and ease of use. However, one disadvantage of Stockfish is that it can be resource-intensive, requiring a powerful computer to run efficiently. In comparison to our survey report, Stockfish was also found to be one of the most popular and accurate chess engines.

Komodo:

Komodo is another popular chess engine that uses a unique search algorithm called Monte Carlo Tree Search. This algorithm allows Komodo to analyze positions much faster than other chess engines while still maintaining accuracy. One advantage of Komodo is its speed, which makes it an excellent choice for players who want quick analysis. However, the disadvantage of using Komodo is that it can sometimes miss tactical opportunities due to its reliance on fast, brute-force calculations. In comparison to our survey report, Komodo was also found to be one of the popular choices among users.

Houdini:

Houdini is a proprietary chess engine that uses a combination of traditional alpha-beta search and advanced heuristics to find the best move in a given position. It has been ranked as one of the strongest chess engines and has won several computer chess championships. The advantages of using Houdini include its accuracy, depth of analysis, and speed. However, one disadvantage of Houdini is that it is a paid engine, which may deter some users. In comparison to our survey report, Houdini was not found to be as popular as Stockfish or Komodo.

Deep Blue:

Deep Blue was the first chess engine to defeat a reigning world chess champion, Garry Kasparov, in 1997. It used a combination of advanced algorithms and brute force to analyze positions and find the best move. While Deep Blue is no longer actively developed, its victory over Kasparov marked a significant milestone in the development of chess engines. The advantages of using Deep Blue include its historical significance and its use of advanced algorithms. However, one disadvantage is that it is no longer actively developed or supported. In comparison to our survey report, Deep Blue was not found to be a popular choice among users as it is no longer relevant.

2. LITERATURE SURVEY

Murray Campbell, et al: Deep Blue - A chess-playing computer that defeated world champion Garry Kasparov in 1997. Deep Blue was a massive parallel processing computer that was capable of analyzing over 200 million chess positions per second. It utilized a combination of brute-force search algorithms and sophisticated evaluation functions to make its moves.[1]

Fogel et al: Evolutionary Computation - An approach to machine learning that is based on the principles of natural selection. Fogel and his team applied evolutionary computation techniques to create artificial neural networks that were capable of playing games like Tic-Tac-Toe, Chess and Othello at a high level.[2]

Tong Lai Yu employed the well-known open-source computer chess engine Beowulf for our project, which serves as our chess engine. Despite being single threaded, Beowulf is simple to use and comprehend. It used the single-threaded Beowulf chess engine. Beowulf scans a game tree using the Negamax Search algorithm. Being a text-based engine and missing a genuine multimedia user interface with images are disadvantages of this.[3]

To create chess commentary writings in a variety of areas, Hongyu Zang et al study an innovative technique for automated chess commentary generation. (e.g., description, comparison, planning, etc.). We integrate a neural chess engine with text generation models to help in encoding boards, predicting moves, and evaluating scenarios. By simultaneously training the neural chess engine and the generation models for multiple categories, the models are enhanced. In a benchmark Chess Commentary data set, we run studies on five distinct categories, and both automatic and manual evaluations produce positive results.[4]

The self-play method is used by the chess engine Giraffe, created by Matthew Lai, to acquire all of its domain-specific knowledge with a little amount of hand-crafted programming expertise. In contrast to past attempts using machine learning merely to perform parameter-tuning on customised evaluation functions, Giraffe's learning system performs autonomous feature extraction and pattern recognition. The trained evaluation function performs on par with the evaluation functions of contemporary chess engines, which all include thousands of lines of painstakingly hand-written pattern recognizers that have been refined over many years by both human and computer chess experts. Giraffe is the name of the most successful end-to-end machine learning chess programme to date.[5]

Dmitri Gusev et al converted an open-source chess engine to Android using the Native Development Kit (NDK), compared it against rival engines, and participated in competitive play. During the porting process, a number of issues and disclosures surfaced, some of which might be common to other mobile application ports and others of which are presumably unique to the chess game. They learned that chess engines' architecture, which is based on the universal chess interface (UCI) protocol, allows for the quick adoption of a sophisticated user interface and that only a few modifications are necessary to create a working engine.[6]

Diego Real et al co. created the duchess chess system,each game allows for a maximum of six players, plus the addition of a seventh coin called as the "Duchess" that can be used to move the bishop and knight. The computer, who serves as a third player in this engine, can help out if necessary and can replace a player who loses the game. It was formerly difficult to operate this engine due to a lack of effective technology, but now it is possible thanks to the Tree Strap Algorithm. The tree search can be made faster by searching cumulatively or by only searching the board where the changes have occurred. As a result, even with six players online at once, the game still plays rapidly.[7]

Rahul A. R. et al study chess from the perspective of genetic algorithms.We fully train a genetic player using only a few lernable factors. Multi-niche crowding is used to optimise Positional Value Tables (PVTs), which are widely employed in chess algorithms to evaluate the quality of a position. With a relatively simple setup and only 1000 generations of development, the player becomes an International Master. After the position assessment has been made, the material difference between the player positions is returned. Because it depends on the appraisal of the pieces, it cannot understand why the opponent would want to give up the piece in exchange for a positional advantage.[8]

Genetic algorithms, as opposed to other conventional chess engines, can be utilised to overcome brute force and improve AI to the point where it can outperform a human player, according to Omid E. David et al. With each iteration of the algorithm, the fitness of the population grows, increasing the possibility that a machine would surpass a person. In the beginning, chess engines could only make one move at a time, but as algorithms improved, they allowed them to carry out tens of lines of searches at once during a match.[9]

D.M. Raif, et al.'s paper 's main objective was to improve the game's visual appeal by using ceramic material into the artwork. Cartoons are regarded to be the finest way to send a message to the audience of the game, so this paper turns the traditional chess pieces into ceramic cartoonist chess pieces for increased visual appeal. Each 3D character is constructed from a number of 2D photos that are used to alter its motions and build it according to the specified model. To distinguish it from other engines, each component is given a distinct form, texture, size, and movement.[10]

3. METHODOLOGY

The evaluation function is the most significant component employed in chess AI development. The result of this module, which accepts a chess position as input, is a number. If this figure gives us an evaluation of 0, it means that the odds are equal for both participants. When the

positive number is higher, white people benefit more; when the negative number is lower, black people benefit more. Before producing an evaluation, the algorithm searches through numerous data. This features a hardcoded imitation of the great mastermind. Consider this to get a basic understanding of chess and chess engines.

A heuristic function weighs each measurement of the chessboard before calculating the value of each player's advantage in terms of numbers. In this case, the worth of each chess piece is being evaluated. The following values apply. 5: Rooks, 3: Knights 3: Bishops, 1: Pawn, and 9: Queens.

Some common used algorithms in Chess AI engines are Min Max Algorithm and Alpha Beta Pruning.

Min-Max Algorithm:-

The Minimax algorithm works by exploring all possible moves that can be made from the current board position. For each move, the algorithm simulates a series of moves, creating a tree of possible future states. It assigns a score to each possible future state based on how advantageous it is to the computer, assuming that the opponent will make the best possible moves to counter the computer's moves.The algorithm works by alternating between maximizing the score (i.e., finding the best move for the computer) and minimizing the score (i.e., assuming the opponent will make the best possible moves to counter the computer's moves). The algorithm continues to evaluate each node of the tree until it reaches a specified depth or a terminal state (i.e., a checkmate or draw).Once the algorithm has evaluated all the possible moves and their resulting future states, it chooses the move that leads to the best score for the computer. This move is then played by the computer.The Minimax algorithm is a fundamental building block of many Chess AI engines, and it is often combined with other algorithms to improve the overall performance of the system.

Nega-Max Algorithm:-

NegaMax is a variant of the Minimax algorithm that is commonly used in Chess AI. Like Minimax, NegaMax is a recursive algorithm that evaluates possible moves in a game tree and selects the best move for a given player. However, NegaMax simplifies the algorithm by using the fact that the score for the opposing player is simply the negative of the score for the current player.In NegaMax, each move is assigned a score based on a static evaluation function that estimates the value of the board position after the move is made. The algorithm then recursively evaluates the positions resulting from each possible move and selects the one with the highest score. The score of each position is computed as the negative of the score of the opponent's best move.To speed up the search, NegaMax uses alpha-beta pruning, which cuts off parts of the search tree that are not relevant to the final score. Alpha-beta pruning uses two parameters, alpha and beta, to represent the best score found so far for the current player and the best score found so far for the opponent, respectively. If the score for a position is worse than alpha or better than beta, it is not necessary to search further because it will never be chosen.

NegaMax is a powerful algorithm that has been used in many strong Chess AI programs. However, it can be slow and memory-intensive for deep searches, and improvements such as transposition tables and iterative deepening are often used to overcome these limitations.

Alpha-Beta Pruning:-

Alpha-beta pruning is a technique used in chess AI to improve the efficiency of the search algorithm by reducing the number of nodes that need to be evaluated. It works by keeping track of two values during the search: alpha and beta.Alpha represents the maximum value that the maximizing player can achieve so far, while beta represents the minimum value that the minimizing player can achieve so far. During the search, if a node's value exceeds beta for a minimizing player or falls below alpha for a maximizing player, that node and all its descendants can be pruned since they will not affect the final result.This pruning technique helps reduce the number of nodes evaluated during the search, resulting in faster and more efficient chess AI. However, the effectiveness of alpha-beta pruning depends on the order in which the nodes are evaluated. If the best moves are evaluated first, more nodes can be pruned, leading to even greater efficiency.

4. CONCLUSION

In conclusion, the survey reveal that chess engines have become an essential tool for chess players and enthusiasts of all levels. Different chess engines use various algorithms and approaches to find the best move in a given position, each with their advantages and disadvantages. However, the most critical factors for users in selecting a chess engine are accuracy, depth of analysis, and ease of use.Regarding the features of chess engines, we found that most users prioritize accuracy and depth of analysis. Other important features include ease of use, speed, and the ability to learn from past games.Chess engines provide objectivity and faster analysis than human players, making them a valuable tool in identifying and correcting mistakes, improving gameplay, and gaining a competitive advantage. As technology continues to evolve, it is likely that chess engines will also continue to improve, leading to exciting new developments in the world of chess.

ACKNOWLEDGEMENT

We are thankful to guide of IT department, **Prof. Shaila Pawar** for helping, giving guidance & giving information in “Chess Engine based on AI :A Survey”.

REFERENCES

1. Murray Campbell, A. Joseph Hoane Jr. b, Feng-hsiung Hsu, Deep Blue, Elsevier2002, Artificial Intelligence 134 (2002) 578.

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, IEEEComputer Society, 2009.

4. Hongyu Zang and Zhiwei Yu and Xiaojun Wan Institute of Computer Science and Technology, Peking University The MOE Key Laboratory of Computational Linguistics, Peking University Center for Data Science, Peking University "Automated Chess Commentator Powered by Neural Chess Engine".

5. Matthew Lai Imperial College london Departnent of Computing "Giraffe: Using Deep Reinforcement Learning to Play Chess".

6. Corey Abshire and Dmitri Gusev Indiana University, Purdue University Columbus, IN, U.S.A., Columbus, IN, U.S.A "Firenzina: Porting a Chess Engine to Android".

7. Diogo Real and Alan Blair, Learning a Multi-Player Chess Game with TreeStrap,2016 IEEE Congress on Evolutionary Computation.

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

9. 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.

10. 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.