Chess

1 Introduction

Chess seems like a distinctly human activity, requiring intelligence and thought, so how can a computer possibly do it? Chess AI breaks down the complex game of chess into mathematical calculations and algorithms. Humans approach the game from an entirely different perspective.

If you've ever watched someone learn to play chess, you know human chess players start with very limited abilities. Once a player understands the basic rules that control each piece, they can "play" chess. Early defeats are often learning opportunities — "Oh, I didn't think about that!" or "I didn't see that coming!" are common exclamations.

The human mind absorbs these experiences, stores away different chessboard configurations, discovers certain tricks and ploys, and generally soaks up the nuances of the game one move at a time. Computers do none of this. Instead, computers don't "think" — they calculate a set of formulas that guide them to make optimal moves.

As chess engines in computers have advanced, the quality of these calculated moves have gotten better and better. AI chess calculators are now the best chess players on the planet, even though they do it totally blindly. Despite the complexity of playing chess, these engines rely solely on calculations. So how does a computer do it? Let's take a closer look.

2 Computers and Chess

Let's say you start with a chessboard set up for the start of a game. Each player has 16 pieces. Let's say that white starts. White has 20 possible opening moves:

The white player can move any pawn forward one or two chess positions. The white player can move either knight in two different ways. The white player chooses one of those 20 moves and makes it.

For the black player, the options are the same: 20 possible moves. So black chooses a move.

Now white can move again. This next move depends on the first move that white chose to make, but there are about 20 or so moves white can make given the current board position, and then black has 20 or so moves it can make, and so on. This is how a computer looks at chess. It thinks about it in a world of "all possible moves," and it makes a big tree for all of those moves, like this:

In this tree, there are 20 possible moves for white. There are 20 * 20 = 400 possible moves for black, depending on what white does. Then there are 400 * 20 = 8,000 for white. Then there are 8,000 * 20 = 160,000 for black, and so on. If you were to fully develop the entire tree for all possible chess moves, the total number of board positions is about

1,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,0
or 10120 — give or take a few.

That's a very big number. For example, there have only been 1026 nanoseconds since the Big Bang. There are thought to be only 1075 atoms in the entire universe. When you consider that the Milky Way galaxy contains billions of suns, and there are billions of galaxies, you can see that that's a whole lot of atoms. But that number is dwarfed by the number of possible chess moves. Chess is a pretty intricate game!

No computer is ever going to calculate the entire tree. What a chess computer tries to do is generate the board-position tree five or 10 or 20 moves into the future. Assuming there are about 20 possible moves for any board position, a five-level tree contains 3,200,000 board positions. A 10-level tree contains about 10,000,000,000,000 (10 trillion) positions. The depth of the tree that a computer can calculate is controlled by the speed of the computer playing the game. The fastest chess computers can generate and evaluate millions of board positions per second.

Once it generates the tree, the computer needs to "evaluate the board positions." Using a search algorithm, the AI explores possible moves and outcomes several levels ahead in the game. However, to decide which move is optimal, it must assign a value to each board position it encounters. This is where the evaluation function comes into play. For example, if the computer is playing white and a certain board position has 11 white pieces and nine black pieces, the simplest evaluation function might be:

11 - 9 = 2 Obviously, that formula is way too simple for chess — after all, some chess pieces are more valuable than others. So the formula might apply a weight to each type of piece. As the programmer thinks about it, they make the evaluation function more and more complicated by adding things like board position, control of the center, vulnerability of the king to check, vulnerability of the opponent's queen, and tons of other parameters. No matter how complicated the function gets, however, it's condensed down to a single number that represents the "goodness" of that board position.

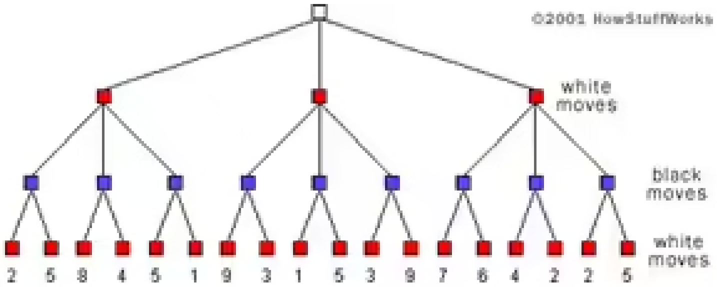

Figure 1: Three level diagram of Chess Ai

3 Three-Level Tree Diagram

The diagram shows a three-level tree that looks three moves ahead and has evaluated the value of the final board positions.

The computer is playing as the white player. The black player has moved and left the board position at the top of the tree. In this tree, white can make three possible moves. From each of those three possible moves, black can make three possible moves. From each of those nine board positions, white can make two possible moves. (In real life, the total number of moves from any position is 20 or so, but that would be hard to draw.)

To decide what to do, the computer looks at this tree and works upward from the bottom. Its calculations are set up so that it finds the best board positions from each of the possible positions black will take (it takes the maximum).

One level up, it assumes that black will choose the worst possible position for white (it takes the minimum).

Finally, it takes the maximum of the top three numbers: 7. That is the move the computer will make. Once black makes its move, the computer goes through this whole process again, generating a new tree and evaluating all of the board positions to figure out its next move.

This approach is called the minimax algorithm because it alternates between the maximums and minimums as it moves up the tree. By applying a technique called alpha beta pruning, the algorithm can run about twice as fast and requires a lot less memory. As you can see, this process is completely mechanical and involves no thought. It is simply a brute force calculation that applies an evaluation function to all possible board positions in a tree of a certain depth.

What is interesting is that this sort of technique works pretty well. On a fast-enough computer, the algorithm can look far enough ahead to play a very good game. If you add in learning techniques that modify the evaluation function based on past games, the machine can even improve over time.

The key thing to keep in mind, however, is that this is nothing like human thought. When we learn how human thinking works and create a computer that uses those techniques to play chess, we will really be onto something ...

4 The Power of Modern Chess Engines

At the heart of every AI-driven chess system is a powerful chess engine. This engine is the combination of search algorithms, evaluation functions, and, in more advanced cases, machine learning techniques. Chess engines have become the ultimate players in the game, consistently outperforming even the best human grandmasters. Whether based on brute-force calculations or adaptive learning, AI chess has changed the landscape of the classic game, becoming an indispensable tool for chess players seeking to analyze their games and improve their strategies.