PUZZLE SUITE · GAME THEORY NOTES

The Game Theory Behind Chess Solver

Two-player, zero-sum games of perfect information: chess and checkers

Chafic Charafeddine · chafic.me

A short game-theory primer

Game theory is the mathematics of decisions whose outcome depends on choices, chance, or other people. The ideas below come up throughout this page.

Perfect vs imperfect information

In a game of **perfect information**, every player sees everything that has happened so far, as in chess or Connect 4. In a game of **imperfect information**, something is hidden, such as an opponent's cards in poker. [1]

Zero-sum

A game is **zero-sum** when one player's gain is exactly the other's loss. Most two-player board games and heads-up poker are zero-sum. [2]

Minimax

Minimax means choosing the move whose *worst-case* result is best, on the assumption that the opponent always replies as well as possible. Von Neumann's 1928 minimax theorem showed that every finite two-player zero-sum game has a well-defined value when players are allowed to randomise. [3][4]

Nash equilibrium

A **Nash equilibrium** is a set of strategies in which no player can do better by changing only their own strategy. John Nash proved in 1950 that every finite game has at least one, possibly in mixed (randomised) strategies. In poker this is what "GTO" (game-theory optimal) refers to. [5][6]

Expected value (EV)

The **expected value** is the probability-weighted average outcome of a decision: what it would earn per play over many repetitions. When there is no opponent to outwit, the best play is usually just the play with the highest EV. [7]

Solved games

Following Allis, a game is **ultra-weakly solved** when its result under perfect play is known, **weakly solved** when a strategy that achieves that result from the start is known, and **strongly solved** when perfect play is known from every legal position. [8][9]

Chess

2 PLAYERS · ZERO-SUM · PERFECT INFORMATION

1 The game, in plain words

Chess is played by two players, White and Black, on an 8×8 board. Each side starts with sixteen pieces: a king, a queen, two rooks, two bishops, two knights and eight pawns, and each piece type moves in its own way. White moves first, and the players then alternate. You win by **checkmate**, which means attacking the opposing king so that it has no way to escape. A game can also end in a draw, for example by **stalemate** (the player to move has

no legal move but is not in check), by agreement, by threefold repetition of a position, or under the fifty-move rule.[10]

2 The game theory behind it

Chess is a **two-player, zero-sum game of perfect information** with no chance involved. Both players see the whole board, nothing is random, and one side's win is the other's loss. **Zermelo's theorem** (from Ernst Zermelo's 1913 work) says that in a finite game like this, exactly one of three things is true. White can force a win, Black can force a win, or both sides can force at least a draw.[11] So chess has a definite "correct" result. Nobody knows what it is, because **chess is unsolved**, not even ultra-weakly.[9]

The obstacle is size. In 1950, Claude Shannon gave a conservative estimate of about 10^{120} possible games, a figure now known as the Shannon number. That is far too many to enumerate.[12][13] Positions with very few pieces *are* solved. **Endgame tablebases** are complete databases built backwards from checkmates, and they give perfect play for positions with up to seven pieces.[14]

3 Finding the best play, and how the app does it

The basic procedure is **minimax**. Look ahead through the tree of possible moves, score the positions at the end of the look-ahead, then work back up the tree, assuming each side picks the move that is best for itself.[4] **Alpha-beta pruning** gives exactly the same answer while skipping branches that provably cannot change the result. With good **move ordering** (trying the most promising moves first), in the best case it can search about twice as deep in the same time.[15][16]

Because chess cannot be searched to the end, engines stop at some depth and use an **evaluation function**, an estimate of how good a position is. **Stockfish** is a free, open-source engine and one of the strongest ever built. It combines a heavily refined alpha-beta search with a neural-network evaluation called NNUE, which it has used since Stockfish 12 in 2020.[17][18] Its play is extremely strong, but it is not a proof. Its verdicts are search results, not solved values.

IN THE APP

Chess Solver's analysis relies on Stockfish-style engine search: alpha-beta minimax with an evaluation at the search horizon. Its suggested moves are the engine's best findings at the depth it searched. They are not proven results, because chess has not been solved.

Checkers

2 PLAYERS · ZERO-SUM · WEAKLY SOLVED

1 The game, in plain words

Checkers (English draughts) is played on the dark squares of an 8×8 board. Each player starts with twelve pieces. Ordinary pieces move one square diagonally forward. You capture by jumping diagonally over an opposing piece to an empty square, and capturing is **compulsory** when possible, including chains of several jumps. A piece that reaches the far row becomes a **king** and may also move backwards. You win if your opponent has no pieces left or no legal moves.[19]

2 The game theory behind it

Like chess, checkers is a two-player, zero-sum game of perfect information, so Zermelo's theorem says it has a definite value. In 2007, Jonathan Schaeffer's team at the University of Alberta published "Checkers is solved" in *Science*. With perfect play by both sides, the game is a **draw**. The proof built on their Chinook program and on computations that had been running almost continuously since 1989.[20][21]

The result is a **weak solution**. The team proved that each side can secure at least a draw from the standard starting position, but they did not compute perfect play for every legal position, which would be a strong solution. The proof combined forward search from the opening with huge endgame databases.[20][9]

Draw
value of checkers with perfect play (proved 2007)

Weakly
solved, from the standard start position

Unsolved
status of chess (value unknown)

3 Finding the best play, and how the app does it

A practical checkers engine uses the same toolkit as a chess engine, minus the proof machinery:

- **Alpha-beta search**: minimax that prunes branches which cannot affect the result.[15]
- **Iterative deepening**: search to depth 1, then 2, then 3, and so on until time runs out. A good move is always available, and each pass helps order the moves for the next.[22]
- **Transposition table**: a cache of positions already searched. Different move orders often reach the same position (a "transposition"), so the engine can reuse earlier work instead of searching it again.[23]

IN THE APP

The checkers engine uses alpha-beta search with iterative deepening and a transposition table. It does **not** use endgame tables. It plays strongly at the depth it reaches, but it does not reproduce the 2007 proof, so its moves are not guaranteed to be perfect.

At a glance

Game	Type	Solved?	Known value	App approach
Chess	2 players, zero-sum, perfect information, no chance	No	Unknown (Zermelo: win, loss or draw)	Stockfish-style alpha-beta engine search
Checkers	2 players, zero-sum, perfect information, no chance	Weakly (2007)[20]	Draw	Alpha-beta, iterative deepening, transposition table; no endgame tables

DETAILS-PAGE BLURB

Chess Solver analyses chess and checkers, two classic two-player, zero-sum games of perfect information. Zermelo's theorem says each has a definite result with perfect play. Checkers was proved a draw in 2007, but chess remains unsolved. The app uses alpha-beta minimax search, the same family of techniques behind engines like Stockfish.

References

1. Wikipedia. "Perfect information." https://en.wikipedia.org/wiki/Perfect_information
2. Wikipedia. "Zero-sum game." https://en.wikipedia.org/wiki/Zero-sum_game
3. von Neumann, J. (1928). "Zur Theorie der Gesellschaftsspiele." *Mathematische Annalen* 100, 295–320. https://doi.org/10.1007/BF01448847
4. Wikipedia. "Minimax." https://en.wikipedia.org/wiki/Minimax
5. Nash, J. F. (1950). "Equilibrium points in n-person games." *PNAS* 36(1), 48–49. https://doi.org/10.1073/pnas.36.1.48
6. Wikipedia. "Nash equilibrium." https://en.wikipedia.org/wiki/Nash_equilibrium
7. Wikipedia. "Expected value." https://en.wikipedia.org/wiki/Expected_value
8. Allis, L. V. (1994). *Searching for Solutions in Games and Artificial Intelligence*. PhD thesis, University of Limburg (Maastricht). https://project.dke.maastrichtuniversity.nl/games/files/phd/SearchingForSolutions.pdf
9. Wikipedia. "Solved game." https://en.wikipedia.org/wiki/Solved_game
10. Wikipedia. "Rules of chess." https://en.wikipedia.org/wiki/Rules_of_chess
11. Wikipedia. "Zermelo's theorem (game theory)." https://en.wikipedia.org/wiki/Zermelo%27s_theorem_(game_theory)
12. Shannon, C. E. (1950). "Programming a computer for playing chess." *Philosophical Magazine* 41(314), 256–275. https://doi.org/10.1080/14786445008521796
13. Wikipedia. "Shannon number." https://en.wikipedia.org/wiki/Shannon_number
14. Wikipedia. "Endgame tablebase." https://en.wikipedia.org/wiki/Endgame_tablebase
15. Knuth, D. E. & Moore, R. W. (1975). "An analysis of alpha-beta pruning." *Artificial Intelligence* 6(4), 293–326. https://doi.org/10.1016/0004-3702(75)90019-3
16. Chess Programming Wiki. "Alpha-Beta." https://www.chessprogramming.org/Alpha-Beta
17. Stockfish (official site). Free and open-source chess engine. https://stockfishchess.org/
18. Chess Programming Wiki. "Stockfish" (history, including NNUE evaluation from Stockfish 12, 2020). https://www.chessprogramming.org/Stockfish
19. Wikipedia. "English draughts." https://en.wikipedia.org/wiki/English_draughts
20. Schaeffer, J., Burch, N., Björnsson, Y., Kishimoto, A., Müller, M., Lake, R., Lu, P. & Sutphen, S. (2007). "Checkers is solved." *Science* 317(5844), 1518–1522. https://doi.org/10.1126/science.1144079
21. Wikipedia. "Chinook (computer program)." https://en.wikipedia.org/wiki/Chinook_(computer_program)
22. Chess Programming Wiki. "Iterative Deepening." https://www.chessprogramming.org/Iterative_Deepening
23. Chess Programming Wiki. "Transposition Table." https://www.chessprogramming.org/Transposition_Table