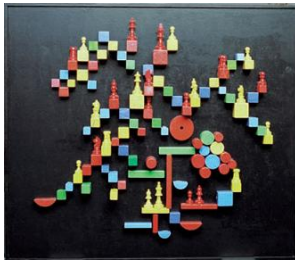


[Home](#) * [Search](#)



[Bernd Besser](#), Auf der Suche [\[1\]](#)

Because having a perfect [evaluation](#) is impossible, chess programs must rely on some type of **Search** in order to play reasonably.

Searching involves looking ahead at different [move](#) sequences and evaluating the [positions](#) after [making the moves](#). Formally, searching a two-player [zero-sum](#) board game with [perfect information](#) implies [traversing](#) and min-maxing a [tree-like data-structure](#) by various [search algorithms](#).

Table of Contents

[Shannon's Types](#)

[The Search Tree](#)

[Search Algorithms](#)

[Depth-First Search](#)

[Alpha-Beta Enhancements](#)

[Obligatory](#)

[Selectivity](#)

[Scout and Friends](#)

[Alpha-Beta goes Best-First](#)

[Best-First Search](#)

[Opponent Model](#)

[Parallel Search](#)

[Using Time](#)

[Related Issues](#)

[See also](#)

[Publications](#)

[1960 ...](#)

[1970 ...](#)

[1980 ...](#)

[1990 ...](#)

[2000 ...](#)

[2010 ...](#)

[2015 ...](#)

[Forum Posts](#)

[2000 ...](#)

[2005 ...](#)

[2010 ...](#)

[2015 ...](#)

[External Links](#)

[References](#)

Shannon's Types

[Claude Shannon](#) categorized searches into two types ^[2] :

- [Type A](#) - a [brute-force search](#) looking at every variation to a given [depth](#)
- [Type B](#) - a [selective search](#) looking at "important" branches only

Inspired by the experiments of [Adriaan de Groot](#) ^[3], Shannon and early programmers favored Type B strategy. Type B searches use some type of static heuristics in order to only look at branches that look important - with some risk to oversee some serious tactics not covered by the plausible move selector. Type B was most popular until the 1970's, when Type A programs had enough processing power and more efficient brute force algorithms to become stronger. Today most programs are closer to Type A, but have some characteristics of a Type B as mentioned in [selectivity](#).

The Search Tree

The [search tree](#) as subset of the [search space](#) is a [directed graph](#) of [nodes](#), the alternating white and black to move [chess positions](#) - and edges connecting two nodes, representing the [moves](#) of either side. The [root](#) of the search-tree is the position we like to evaluate to find the [best move](#). Because of [transpositions](#) due to different move sequences, nodes inside the tree may occur from various ancestors - even within a different number of moves. The search tree contains various cycles, since both sides may [repeat](#) a former position with the minimum of two reversible moves each, or four [plies](#). Cycles are usually eliminated and not searched twice, which results in searching a [directed acyclic graph](#) (DAG).

- [Search Space](#)
- [Search Tree](#)
- [Root](#)
- [Node](#)
- [Conspiracy Numbers](#)
- [Move List](#)
- [Principal Variation](#)
- [Graph History Interaction](#)
- [Path-Dependency](#)
- [Repetitions](#)
- [Transposition](#)
- [Score](#)

Search Algorithms

Most chess-programs use a variation of the [alpha-beta](#) algorithm to search the tree in a [depth-first](#) manner to attain an order of magnitude performance improvement over a pure [minimax](#) algorithm. Although [move ordering](#)

doesn't affect the performance of a pure mini-max search (as all branches and nodes are searched) — it becomes crucial for the performance of alpha beta search and enhancements such as [EVS](#), [NegaScout](#) and [MTD\(f\)](#). [Hans Berliner's](#) chess-program [HiTech](#) and [Ulrich Lorenz's](#) program

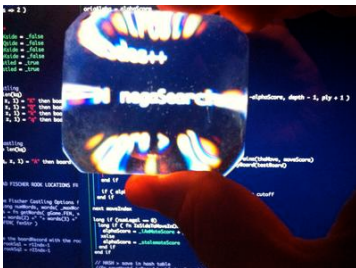
[P.ConNer's](#) used [best-first](#) approaches quite successfully.

Depth-First Search

[Depth-First](#) search starts at the root and explores as far as possible along each branch before backtracking. Memory requirements are moderate, since only one path from the root to one leaf is kept in memory. The

giga bytes of [RAM](#) in recent computers is utilized by a [transposition table](#). Minimax and Negamax are mentioned for educational reasons as the prototypes for the more advanced algorithms. They otherwise have no practical relevance in software, except traversing a minimax tree inside a [perft](#) framework for [testing](#) the [move generator](#). Depth-first algorithms are generally embedded inside an [iterative deepening](#) framework for [time control](#) and [move ordering](#) issues.

- [Minimax](#)
- [Negamax](#)
- [Alpha-Beta](#)



negaSearch function at the heart of every chess program

Alpha-Beta Enhancements

Obligatory

- [Transposition Table](#)
- [Iterative Deepening](#)
- [Aspiration Windows](#)

Selectivity

- [Quiescence Search](#)
- [Selectivity](#)
- [Mate Search](#)

Scout and Friends

- [Scout](#)
- [NegaScout](#)
- [Principal Variation Search](#)

Alpha-Beta goes Best-First

- [NegaC*](#)
- [MTD\(f\)](#)

- [Alpha-Beta Conspiracy Search](#)

Best-First Search

[Best-First](#) approaches build a search-tree by visiting the most promising nodes first.

They usually have huge memory requirements, since they keep an exponentially growing search space in memory.

- [B*](#) as used by [Hans Berliner's](#) chess-program [HiTech](#)
- [Conspiracy Number Search](#)
- [MCA \$\beta\$](#)
- [Monte-Carlo Tree Search](#)
- [Proof-number search](#)
- [SSS* and Dual*](#)
- [UCT](#)

Opponent Model

- [Opponent Model Search](#)

Parallel Search

- [Parallel Search](#)
- [Parallel Controlled Conspiracy Number Search](#) as used by [Ulf Lorenz's](#) program [P.ConNerS](#)

Using Time

- [Pondering](#)
- [Time Management](#)

Related Issues

- [Depth](#)
- [Horizon Effect](#)
- [Iterative Search](#)
- [Move Ordering](#)
- [Search Explosion](#)
- [Search Instability](#)
- [Search Pathology](#)
- [Search Statistics](#)
- [Search with Random Leaf Values](#)
- [Theorem-Proving and M & N procedure](#)

See also

- [Backtracking](#)
- [History of Computer Chess](#)
- [Knowledge Search versus Evaluation](#)

Publications

1960 ...

- [Daniel Edwards](#), [Timothy Hart](#) (1961). *The Alpha-Beta Heuristic*, AIM-030, reprint available from [DSpace](#) at [MIT](#). Retrieved on 2006-12-21.
- [Alexander Brudno](#) (1963). *Bounds and valuations for shortening the search of estimates*. Problemy Kibernetiki (10) 141–150 and Problems of Cybernetics (10) 225–241
- [James R. Slagle](#) (1963). *Game Trees, M & N Minimaxing, and the M & N alpha-beta procedure*. Artificial Intelligence Group Report 3, UCRL-4671, University of California
- [James R. Slagle](#), [Philip Bursky](#) (1968). [Experiments With a Multipurpose, Theorem-Proving Heuristic Program](#). *Journal of the ACM*, Vol. 15, No. 1
- [James R. Slagle](#), [John K. Dixon](#) (1969). [Experiments With Some Programs That Search Game Trees](#). *Journal of the ACM*, Vol. 16, No. 2, [pdf](#), [pdf](#)

1970 ...

- [James R. Slagle](#), [John K. Dixon](#) (1970). [Experiments with the M & N Tree-Searching Program](#). *Communications of the ACM*, Vol. 13, No. 3, pp. 147-154.
- [James R. Slagle](#), [Richard C. T. Lee](#) (1971). [Application of game tree searching techniques to sequential pattern recognition](#). *Communications of the ACM*, Vol. 14, No. 2
- [Larry Harris](#) (1973). *The bandwidth heuristic search*. [3. IJCAI 1973](#), [pdf](#)
- [Gerhard Wolf](#) (1973). [Implementation of a dynamic tree searching algorithm in a chess programme](#). *Proceedings of the ACM annual conference*
- [Larry Harris](#) (1974). *Heuristic Search under Conditions of Error*. Artificial Intelligence, Vol. 5, No. 3, pp. 217-234. ISSN 0004-3702. Also published (1977) under the title: *The heuristic search: An alternative to the alpha-beta minimax procedure*. [Chess Skill in Man and Machine](#) (ed. [Peter W. Frey](#)), pp. 157-166
- [Larry Harris](#) (1975) *The Heuristic Search And The Game Of Chess - A Study Of Quiescence, Sacrifices, And Plan Oriented Play*. [4. IJCAI 1975](#), 334-339. reprinted in [Computer Chess Compendium](#) by [David Levy](#) (Editor), pp. 136 - 142
- [Georgy Adelson-Velsky](#), [Vladimir Arlazarov](#), [Mikhail Donskoy](#) (1979). *Algorithms of adaptive search*. *Machine Intelligence 9* (eds. [Jean Hayes Michie](#), [Donald Michie](#) and L.I. Mikulich), pp. 373-384. Ellis Horwood, Chichester.
- [George Stockman](#) (1979). *A Minimax Algorithm Better than Alpha-Beta?* Artificial Intelligence, Vol. 12, No. 2, pp. 179-196. ISSN 0004-3702.

1980 ...

- [Judea Pearl](#) (1981). *Heuristic search theory: A survey of recent results*. [IJCAI-81](#), [pdf](#)
- [Judea Pearl](#) (1982). [The Solution for the Branching Factor of the Alpha-Beta Pruning Algorithm and its Optimality](#). [Communications of the ACM](#), Vol. 25, No. 8
- [Murray Campbell](#), [Tony Marsland](#) (1983). *A Comparison of Minimax Tree Search Algorithms*. [Artificial Intelligence](#), Vol. 20, No. 4, pp. 347-367. ISSN 0004-3702, [pdf](#)
- [Andrew L. Reibman](#), [Bruce W. Ballard](#) (1983). [Non-Minimax Search Strategies for Use against Fallible Opponents](#). Proceedings of [AAAI](#) 83
- [Nanda Srimani](#) (1985). *A New Algorithm (PS*) for Searching Game Trees*. Master's thesis, [University of Alberta](#)
- [Toshihide Ibaraki](#) (1986). *Generalization of Alpha-Beta and SSS* Search Procedures*. [Artificial Intelligence](#), Vol. 29, pp. 73-117. ISSN 0004-3702.
- [Tony Marsland](#), [Nanda Srimani](#) (1986). *Phased State Search*. [Fall Joint Computer Conference](#), [pdf](#)
- [Hermann Kaindl](#), [Helmut Horacek](#), [Marcus Wagner](#) (1986). *Selective Search versus Brute Force*. [ICCA Journal](#), Vol. 9, No. 3
- [Ronald L. Rivest](#) (1987). *Game Tree Searching by Min/Max Approximation*. [Artificial Intelligence](#) Vol. 34, 1, [pdf](#) 1995
- [Bruce Abramson](#) (1989). *Control Strategies for Two-Player Games*. [ACM Computing Surveys](#) 21(2): 137-161, [pdf](#)
- [Stuart Russell](#), [Eric Wefald](#) (1989). [On optimal game-tree search using rational metareasoning](#). In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, Detroit, MI: Morgan Kaufmann, [pdf](#)
- [Liwu Li](#) (1989). [Probabilistic Analysis of Search](#). Ph.D. thesis, [University of Alberta](#), advisor [Tony Marsland](#)

1990 ...

- [Victor Allis](#), [Maarten van der Meulen](#), [Jaap van den Herik](#) (1991). *Proof-Number Search*. Technical Reports in Computer Science, CS 91-01. Department of Computer Science, [University of Limburg](#)
- [Tony Marsland](#) (1992). *Computer Chess and Search*. Encyclopedia of Artificial Intelligence (2nd ed.) (ed. S.C. Shapiro) John Wiley & Sons, Inc. [pdf](#) ^[4] ^[5]
- [Eric B. Baum](#) (1992). *On Optimal Game Tree Propagation for Imperfect Players*. [AAAI-92](#), [pdf](#)
- [Claude G. Diderich](#) (1993). [A Bibliography on Minimax Trees](#). [ACM SIGACT News](#), Vol. 24, No. 4
- [Deniz Yuret](#) (1994). [The Principle of Pressure in Chess](#). TAINN 1994
- [Hans Berliner](#), [Chris McConnell](#) (1995). *B* Probability Based Search*. [Carnegie Mellon University](#) Computer Science research report, Pittsburgh, PA, [postscript](#)
- [Claude G. Diderich](#), [Marc Gengler](#) (1995). [A Survey on Minimax Trees and Associated Algorithms . Minimax and Its Applications](#). [Kluwer Academic Publishers](#)
- [Eric B. Baum](#), [Warren D. Smith](#) (1995). *Best Play for Imperfect Players and Game Tree Search*. Part 1 - Theory
- [Eric B. Baum](#), [Warren D. Smith](#) (1995). *Best Play for Imperfect Players and Game Tree Search*. with pseudocode appendix by [Charles Garrett](#), [ps](#)
- [Warren D. Smith](#), [Eric B. Baum](#), [Charles Garrett](#), [Rico Tudor](#) (1995). *Best Play for Imperfect*

Players and Game Tree Search. Part 2 - Experiments, [ps](#)

- [Andrew N. Walker](#) (1996). *Hybrid Heuristic Search*. [ICCA Journal](#), Vol. 19, No. 1
- [Ingo Althöfer](#) (1997). *On the k-best Mode in Computer Chess: Measuring the Similarity of Move Proposals*. [ICCA Journal](#), Vol. 20, No. 3
- [Eric B. Baum](#), [Warren D. Smith](#) (1997). *A Bayesian Approach to Relevance in Game Playing*. [Artificial Intelligence](#), Vol. 97, [CiteSeerX](#)
- [Donald E. Knuth](#) (1998). [The Art Of Computer Programming](#) Vol 3. *Sorting and Searching*, Second Edition, Addison-Wesley
- [Wim Pijls](#), [Arie de Bruin](#) (1998). [Game Tree Algorithms and Solution Trees](#). [CG 1998](#)

2000 ...

- [Paul E. Utgoff](#), [Richard P. Cochran](#) (2000). [A Least-Certainty Heuristic for Selective Search](#). [CG 2000](#), [pdf](#) » [LCF](#)
- [Thomas Thomsen](#) (2000). [Lambda-Search in Game Trees - with Application to Go](#). [CG 2000](#) also published in [ICGA Journal](#), Vol. 23, No. 4, winning the 2001 [ICGA Journal Award](#), [preprint as pdf](#) » [Lambda-Search](#)
- [Todd W. Neller](#) (2000). *Simulation-Based Search for Hybrid System Control and Analysis*. Ph.D. thesis, [Stanford University](#), advisor [Richard Fikes](#), [pdf](#)
- [Martin Müller](#) (2001, 2002). *Proof-Set Search*. Technical Report TR 01-09, [University of Alberta](#), [CG 2002](#), [CiteSeerX](#) ^[6]
- [Thomas Lincke](#) (2002). *Exploring the Computational Limits of Large Exhaustive Search Problems*. Ph.D thesis, [ETH Zurich](#), [pdf](#) » [Awari](#), [Repetitions](#) ^[7]
- [Steven Walczak](#) (2003). [Knowledge-Based Search in Competitive Domains](#). *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 3
- [David Rasmussen](#) (2004). *Parallel Chess Searching and Bitboards*. Masters thesis, [ps](#)
- [Yan Radovitsky](#), [Solomon Eyal Shimony](#) (2004). *Generalized Model for Rational Game Tree Search*. [pdf](#)
- [Markian Hlynka](#), [Jonathan Schaeffer](#) (2004). *Pre-Searching*. [ICGA Journal](#), Vol. 27, No. 4
- [Markian Hlynka](#), [Jonathan Schaeffer](#) (2005). [Automatic Generation of Search Engines](#). [Advances in Computer Games 11](#)
- [Ulf Lorenz](#) (2006). *A new Implementation of Error Analysis in Game Trees*. [ICGA Journal](#), Vol. 29, No. 2
- [Dmitry Batenkov](#) (2006). *Modern developments of Shannon's Chess*. [pdf](#)
- [Pim Nijssen](#) (2009). *Using Intelligent Search Techniques to Play the Game Khet*. Master's Thesis, [Maastricht University](#), [pdf](#) ^[8]
- [Claude G. Diderich](#), [Marc Gengler](#) (2009). [Minimax Game Tree Searching](#). [Encyclopedia of Optimization](#), [Springer](#)
- [David Silver](#) (2009). *Reinforcement Learning and Simulation-Based Search*. Ph.D. thesis, [University of Alberta](#), [pdf](#)

2010 ...

- [Junichi Hashimoto](#) (2011). *A Study on Game-Independent Heuristics in Game-Tree Search*. Ph.D. thesis, [JAIST](#)

- [Hung-Jui Chang](#), [Meng-Tsung Tsai](#), [Tsan-sheng Hsu](#) (2011). *Game Tree Search with Adaptive Resolution*. [Advances in Computer Games 13](#), [pdf](#)
- [Alexandru Godescu](#) (2011). [Information and search in computer chess](#). ^[9]
- [Pim Nijssen](#), [Mark Winands](#) (2012). *An Overview of Search Techniques in Multi-Player Games*. [ECAI CGW 2012](#)
- [Abdallah Saffidine](#), [Tristan Cazenave](#) (2012). *A General Multi-Agent Modal Logic K Framework for Game Tree Search*. [ECAI CGW 2012](#)
- [Akihiro Kishimoto](#), [Mark Winands](#), [Martin Müller](#), [Jahn-Takeshi Saito](#) (2012). *Game-Tree Search using Proof Numbers: The First Twenty Years*. [ICGA Journal](#), Vol. 35, No. 3
- [Kuo-Yuan Kao](#), [I-Chen Wu](#), [Yi-Chang Shan](#), [Shi-Jim Yen](#) (2012). *Selection Search for Mean and Temperature of Multi-branch Combinatorial Games*. [ICGA Journal](#), Vol. 35, No. 3
- [David Silver](#), [Richard Sutton](#), [Martin Mueller](#) (2013). *Temporal-Difference Search in Computer Go*. [Proceedings of the ICAPS-13 Workshop on Planning and Learning](#), [pdf](#)
- [Jeff Rollason](#) (2014). [Interest Search - Another way to do Minimax](#). [AI Factory](#), Summer 2014
- [Tom Holden](#) (2014). *Notes on an alternative approach to move choice in games such as Chess*. [pdf](#)

2015 ...

- [Jakub Pawlewicz](#), [Ryan Hayward](#) (2015). *Feature Strength and Parallelization of Sibling Conspiracy Number Search*. [Advances in Computer Games 14](#)
- [Mohd Nor Akmal Khalid](#), [Umi Kalsom Yusof](#), [Hiroyuki Iida](#), [Taichi Ishitobi](#) (2015). [Critical Position Identification in Games and Its Application to Speculative Play](#). [ICAART 2015](#)
- [Mohd Nor Akmal Khalid](#), [E. Mei Ang](#), [Umi Kalsom Yusof](#), [Hiroyuki Iida](#), [Taichi Ishitobi](#) (2015). [Identifying Critical Positions Based on Conspiracy Numbers](#). [Agents and Artificial Intelligence, ICAART 2015 - Revised Selected Papers](#)
- [Ting-Han Wei](#), [Chao-Chin Liang](#), [I-Chen Wu](#), [Lung-Pin Chen](#) (2015). *Software Development Framework for Job-Level Algorithms*. [ICGA Journal](#), Vol. 38, No. 3
- [Tobias Joppen](#), [Miriam Moneke](#), [Nils Schroder](#), [Christian Wirth](#), [Johannes Fürnkranz](#) (2017). [Informed Hybrid Game Tree Search for General Video Game Playing](#). [IEEE Transactions on Computational Intelligence and AI in Games](#), Vol. PP, No. 99

Forum Posts

2000 ...

- [About search algorithms and heuristics](#) by [José Carlos](#), [CCC](#), May 26, 2000
- [Search algorithms and effeciency](#) by [Kim Roper Jensen](#), [CCC](#), May 30, 2001
- [Search algorithms in chess programs](#) by [Russell Reagan](#), [CCC](#), December 12, 2001
- [Search algorithms](#) by [Renze Steenhuisen](#), [CCC](#), November 06, 2003
- [Game tree search algorithms](#) by [Russell Reagan](#), [CCC](#), January 20, 2004

2005 ...

- [Search questions](#) by [Sven Schüle](#), [Winboard Forum](#), July 17, 2007 » [Fail-Soft](#), [Principal Variation](#)

[Search](#)

- [Even more search questions](#) by [Sven Schüle](#), [Winboard Forum](#), July 17, 2007 » [Root](#), [Iterative Deepening](#)
- [Search or Evaluation?](#) by [Ed Schröder](#), [Hiarcs Forum](#), October 05, 2007 » [Search versus Evaluation](#), [Evaluation](#)
- [Re: Search or Evaluation?](#) by [Mark Uniacke](#), [Hiarcs Forum](#), October 14, 2007
- [Efficient algorithm for k-best mode?](#) by [Gijsbert Wiesenekker](#), [CCC](#), November 17, 2007

2010 ...

- [Scaling at 2x nodes \(or doubling time control\)](#), by [Kai Laskos](#), [CCC](#), July 23, 2013 » [Doubling TC](#), [Diminishing Returns](#), [Playing Strength](#), [Houdini](#)
- [Is search irrelevant when computing ahead of very big trees?](#) by [Fermin Serrano](#), [CCC](#), July 24, 2013 » [Knowledge](#)
- [Improve the search or the evaluation?](#) by [Jens Bæk Nielsen](#), [CCC](#), August 31, 2013 » [Evaluation](#), [Search versus Evaluation](#)
- [Slow Searchers?](#) by Michael Neish, [CCC](#), November 02, 2013
- [A new algorithm accounting for the uncertainty in eval funcs](#) by [Tom Holden](#), [CCC](#), November 12, 2014

2015 ...

- [Search algorithm in it's simplest forum](#) by [Mahmoud Uthman](#), [CCC](#), February 25, 2015 » [Alpha-Beta](#), [Quiescence Search](#)
- [Time assignment to children](#) by [Matthew Lai](#), [CCC](#), July 26, 2015
- [Some musings about search](#) by [Ed Schroder](#), [CCC](#), August 14, 2015 » [Automated Tuning](#)
- [Search](#) by [Laurie Tunnicliffe](#), [CCC](#), June 24, 2016
- [Searching using slow eval with tactical verification](#) by [Matthew Lai](#), [CCC](#), September 06, 2016
- [Root search](#) by [Laurie Tunnicliffe](#), [CCC](#), September 08, 2016 » [Root](#)
- [Doubling of time control](#) by [Andreas Strangmüller](#), [CCC](#), October 21, 2016 » [Doubling TC](#), [Diminishing Returns](#), [Playing Strength](#), [Komodo](#)
- [search efficiency](#) by [Marco Pampaloni](#), [CCC](#), June 23, 2017
- [comparing between search or evaluation](#) by [Uri Blass](#), [CCC](#), October 09, 2017 » [Evaluation](#)

External Links

- [Search algorithm from Wikipedia](#)
- [Combinatorial search from Wikipedia](#)
- [Depth-first search from Wikipedia](#)
- [Boost Graph Library: Depth-First Search](#)
- [Best-first search from Wikipedia](#)
- [A* search algorithm from Wikipedia](#)
- [Breadth-first search from Wikipedia](#)
- [Dijkstra's algorithm from Wikipedia](#)
- [Chapter 3 Solving Problems by Searching](#) from [C463 Artificial Intelligence Syllabus](#) by [Raymond](#)

[F. Wisman](#)

- [Tree Search](#) by [Paul Verhelst](#)
- [Move Searching and Alpha Beta](#) by [Adam Berent](#)
- [Lambda-search Java-code \(version 2.0\)](#) by [Thomas Thomsen](#)
- [Chess Programming Part IV: Basic Search](#) by [François-Dominic Laramée](#), [gamedev.net](#), August 2000
- [Chess Programming Part V: Advanced Search](#) by [François-Dominic Laramée](#), [gamedev.net](#), September 2000
- [Engine - Hispanic Chess Engines I The search function](#) by [Pedro Castro](#)
- [Miroslav Vitouš - Infinite Search](#) (1969), [YouTube](#) Video
[Jack DeJohette](#) ([Joe Chambers](#) track 6), [John McLaughlin](#), [Herbie Hancock](#), [Joe Henderson](#)

References

1. [^ Schach Bilder Welten - Bernd Besser - Galerie](#)
2. [^ Claude Shannon \(1949\). Programming a Computer for Playing Chess. pdf](#)
3. [^ Adriaan de Groot \(1946\). Het denken van den Schaker, een experimenteel-psychologische studie.](#) Ph.D. thesis, [University of Amsterdam](#); N.V. Noord-Hollandse Uitgevers Maatschappij, [Amsterdam](#). Translated with the help of [George Baylor](#), with additions (in 1965) as *Thought and Choice in Chess*. Mouton Publishers, The Hague. ISBN 90-279-7914-6. ([amazon](#))
4. [^ Excellent Computer-Chess Overview Paper Found!](#) by [Ernst A. Heinz](#), [rgcc](#), March 6, 1997
5. [^ Great article for people who wants to write a chess engine](#) by [Miguel A. Ballicora](#), [CCC](#), April 03, 2002
6. [^ Re: A new\(?\) technique to recognize draws](#) by [Dan Andersson](#), June 01, 2002
7. [^ Re: Aquarium IDEA, repetitions, and minimax over cycles](#) by [syzygy](#), [OpenChess Forum](#), September 22, 2012
8. [^ Khet \(game\) from Wikipedia](#)
9. [^ Information and search in computer chess \(Godescu\)](#) by [BB±](#), [OpenChess Forum](#), December 12, 2011
10. [^ A new algorithm accounting for the uncertainty in eval funcs](#) by [Tom Holden](#), [CCC](#), November 12, 2014

[Up one level](#)