

	B chooses B1	B chooses B2	B chooses B3

same matrix with the signs reversed (i.e. if the choices are A1 and B1 then **B** pays 3 to **A**). Then, the minimax choice for **A** is A2 since the worst possible result is then having to pay 1, while the simple minimax choice for **B** is B2 since the worst possible result is then no payment. However, this solution is not stable, since if **B** believes **A** will choose A2 then **B** will choose B1 to gain 1; then if **A** believes **B** will choose B1 then **A** will choose A1 to gain 3; and then **B** will choose B2; and eventually both players will realize the difficulty of making a choice. So a more stable strategy is needed.

<b>A chooses A1</b>	+3	−2	+2
<b>A chooses A2</b>	−1	0	+4
<b>A chooses A3</b>	−4	−3	+1

Some choices are *dominated* by others and can be eliminated: **A** will not choose A3 since either A1 or A2 will produce a better result, no matter what **B** chooses; **B** will not choose B3 since some mixtures of B1 and B2 will produce a better result, no matter what **A** chooses.

**A** can avoid having to make an expected payment of more than 1/3 by choosing A1 with probability 1/6 and A2 with probability 5/6: The expected payoff for **A** would be  $3 \times (1/6) - 1 \times (5/6) = -1/3$  in case **B** chose B1 and  $-2 \times (1/6) + 0 \times (5/6) = -1/3$  in case **B** chose B2. Similarly, **B** can ensure an expected gain of at least 1/3, no matter what **A** chooses, by using a randomized strategy of choosing B1 with probability 1/3 and B2 with probability 2/3. These *mixed* minimax strategies are now stable and cannot be improved.

**Maximin** [\[edit\]](#)

Frequently, in game theory, **maximin** is distinct from minimax. Minimax is used in zero-sum games to denote minimizing the opponent's maximum payoff. In a *zero-sum game*, this is identical to minimizing one's own maximum loss, and to maximizing one's own minimum gain.

"Maximin" is a term commonly used for non-zero-sum games to describe the strategy which maximizes one's own minimum payoff. In non-zero-sum games, this is not generally the same as minimizing the opponent's maximum gain, nor the same as the *Nash equilibrium* strategy.

**In repeated games** [\[edit\]](#)

The minimax values are very important in the theory of *repeated games*. One of the central theorems in this theory, the *folk theorem*, relies on the minimax values.

**Combinatorial game theory** [\[edit\]](#)

In *combinatorial game theory*, there is a minimax algorithm for game solutions.

A *simple* version of the minimax *algorithm*, stated below, deals with games such as *tic-tac-toe*, where each player can win, lose, or draw. If player A *can* win in one move, his best move is that winning move. If player B knows that one move will lead to the situation where player A *can* win in one move, while another move will lead to the situation where player A *can*, at best, draw, then player B's best move is the one leading to a draw. Late in the game, it's easy to see what the "best" move is. The Minimax algorithm helps find the best move, by working backwards from the end of the game. At each step it assumes that player A is trying to **maximize** the chances of A winning, while on the next turn player B is trying to **minimize** the chances of A winning (i.e., to *maximize* B's own chances of winning).

**Minimax algorithm with alternate moves** [\[edit\]](#)

A *minimax algorithm*<sup>[6]</sup> is a recursive *algorithm* for choosing the next move in an n-player *game*, usually a two-player game. A value is associated with each position or state of the game. This value is computed by means of a *position evaluation function* and it indicates how good it would be for a player to reach that position. The player then makes the move that maximizes the minimum value of the position resulting from the opponent's possible following moves. If it is **A**'s turn to move, **A** gives a value to each of his legal moves.

A possible allocation method consists in assigning a certain win for **A** as +1 and for **B** as −1. This leads to *combinatorial game theory* as developed by *John Horton Conway*. An alternative is using a rule that if the result of a move is an immediate win for **A** it is assigned positive infinity and, if it is an immediate win for **B**, negative infinity. The value to **A** of any other move is the minimum of the values resulting from each of B's possible replies. For this reason, **A** is called the *maximizing player* and **B** is called the *minimizing player*, hence the name *minimax algorithm*. The above algorithm will assign a value of positive or negative infinity to any position since the value of every position will be the value of some final winning or losing position. Often this is generally only possible at the very end of complicated games such as *chess* or *go*, since it is not computationally feasible to look ahead as far as the completion of the game, except towards the end, and instead positions are given finite values as estimates of the degree of belief that they will lead to a win for one player or another.

This can be extended if we can supply a *heuristic* evaluation function which gives values to non-final game states without considering all possible following complete sequences. We can then limit the minimax algorithm to look only at a certain number of moves ahead. This number is called the "look-ahead", measured in "*plies*". For example, the chess computer *Deep Blue* (the first one to beat a reigning world champion, *Garry Kasparov* at that time) looked ahead at least 12 plies, then applied a heuristic evaluation function.<sup>[7]</sup>

The algorithm can be thought of as exploring the *nodes* of a *game tree*. The *effective branching factor* of the tree is the average number of *children* of each node (i.e., the average number of legal moves in a position). The number of nodes to be explored usually *increases exponentially* with the number of plies (it is less than exponential if evaluating *forced moves* or repeated positions). The number of nodes to be explored for the analysis of a game is therefore approximately the branching factor raised to the power of the number of plies. It is therefore *impractical* to completely analyze games such as chess using the minimax algorithm.

The performance of the naïve minimax algorithm may be improved dramatically, without affecting the result, by the use of *alpha-beta pruning*. Other heuristic pruning methods can also be used, but not all of them are guaranteed to give the same result as the un-pruned search.

A naïve minimax algorithm may be trivially modified to additionally return an entire *Principal Variation* along with a minimax score.

**Pseudocode** [\[edit\]](#)

The *pseudocode* for the depth limited minimax algorithm is given below.

```
function minimax(node, depth, maximizingPlayer)
  if depth = 0 or node is a terminal node
    return the heuristic value of node
  if maximizingPlayer
    bestValue := −∞
    for each child of node
      val := minimax(child, depth − 1, FALSE)
      bestValue := max(bestValue, val)
    return bestValue
  else
    bestValue := +∞
    for each child of node
      val := minimax(child, depth − 1, TRUE)
      bestValue := min(bestValue, val)
    return bestValue

(* Initial call for maximizing player *)
minimax(origin, depth, TRUE)
```

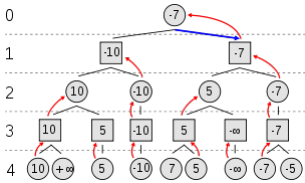
The minimax function returns a heuristic value for *leaf nodes* (terminal nodes and nodes at the maximum search depth). Non leaf nodes inherit their value, *bestValue*, from a descendant leaf node. The heuristic value is a score measuring the favor-ability of the node for the maximizing player. Hence nodes resulting in a favorable outcome (such as a win) for the maximizing player have higher scores than nodes more favorable for the minimizing player. For non terminal leaf nodes at the maximum search depth, an evaluation function estimates a heuristic value for the node. The quality of this estimate and the search depth determine the quality and accuracy of the final minimax result.

Minimax treats the two players (the maximizing player and the minimizing player) separately in its code. Based on the observation that  $\max(a, b) = -\min(-a, -b)$ , minimax may often be simplified into the *negamax* algorithm.

**Example** [\[edit\]](#)

Suppose the game being played only has a maximum of two possible moves per player each turn. The algorithm generates the *tree* on the right, where the circles represent the moves of the player running the algorithm (*maximizing player*), and squares represent the moves of the opponent (*minimizing player*). Because of the limitation of computation resources, as explained above, the tree is limited to a *look-ahead* of 4 moves.

The algorithm evaluates each *leaf node* using a heuristic evaluation function, obtaining the values shown. The moves where the *maximizing player* wins are assigned with positive infinity, while the moves that lead to a win of the *minimizing player* are assigned with negative infinity. At level 3, the algorithm will choose, for each node, the **smallest** of the *child node* values, and assign it to that same node (e.g. the node on the left will choose the minimum between "10" and "+∞", therefore assigning the value "10" to itself). The next step, in level 2, consists of choosing for each node the **largest** of the *child node* values. Once again, the values are assigned to each *parent node*. The algorithm continues evaluating the maximum and minimum values of the child nodes alternately until it reaches the *root node*, where it chooses the move with the largest value (represented in the figure with a blue arrow). This is the move that the player should make in order to *minimize* the *maximum* possible *loss*.



**Minimax for individual decisions** [\[edit\]](#)

Minimax in the face of uncertainty  [edit]

Minimax theory has been extended to decisions where there is no other player, but where the consequences of decisions depend on unknown facts. For example, deciding to prospect for minerals entails a cost which will be wasted if the minerals are not present, but will bring major rewards if they are. One approach is to treat this as a game against *nature* (see **move by nature**), and using a similar mindset as **Murphy's law** or **residentialism**, take an approach which minimizes the maximum expected loss, using the same techniques as in the two-person zero-sum games.

In addition, **expectiminimax trees** have been developed, for two-player games in which chance (for example, dice) is a factor.

Minimax criterion in statistical decision theory  [edit]

*Main article: Minimax estimator*

In classical statistical **decision theory**, we have an **estimator** 



δ


{\displaystyle \delta }

 that is used to estimate a **parameter** 



θ
∈
Θ


{\displaystyle \theta \in \Theta }

. We also assume a **risk function** 



R
(
θ
,
δ
)


{\displaystyle R(\theta ,\delta )}

, usually specified as the integral of a **loss function**. In this framework, 






δ
˜




{\displaystyle {\tilde {\delta }}}

 is called **minimax** if it satisfies

sup

θ




R
(
θ
,
δ
˜
)
=
inf

δ



sup

θ




R
(
θ
,
δ
)


{\displaystyle \sup \_{\theta }R(\theta ,{\tilde {\delta }})=\inf \_{\delta }\sup \_{\theta }R(\theta ,\delta )}

.

An alternative criterion in the decision theoretic framework is the **Bayes estimator** in the presence of a **prior distribution** 



Π


{\displaystyle \Pi }

. An estimator is Bayes if it minimizes the *average* risk

∫

Θ



R
(
θ
,
δ
)
d
Π
(
θ
)


{\displaystyle \int \_{\Theta }R(\theta ,\delta )d\Pi (\theta )}

.

Non-probabilistic decision theory  [edit]

A key feature of minimax decision making is being non-probabilistic: in contrast to decisions using **expected value** or **expected utility**, it makes no assumptions about the probabilities of various outcomes, just **scenario analysis** of what the possible outcomes are. It is thus **robust** to changes in the assumptions, as these other decision techniques are not. Various extensions of this non-probabilistic approach exist, notably **minimax regret** and **Info-gap decision theory**.

Further, minimax only requires **ordinal measurement** (that outcomes be compared and ranked), not *interval* measurements (that outcomes include "how much better or worse"), and returns ordinal data, using only the modeled outcomes: the conclusion of a minimax analysis is: "this strategy is minimax, as the worst case is (outcome), which is less bad than any other strategy". Compare to expected value analysis, whose conclusion is of the form: "this strategy yields E(X)=n." Minimax thus can be used on ordinal data, and can be more transparent.

Maximin in philosophy  [edit]

In philosophy, the term "maximin" is often used in the context of **John Rawls's** *A Theory of Justice*, where he refers to it (Rawls (1971, p. 152)) in the context of The Difference Principle. Rawls defined this principle as the rule which states that social and economic inequalities should be arranged so that "they are to be of the greatest benefit to the least-advantaged members of society".<sup>[]<sup>9</sup></sup>

See also  [edit]

- Alpha-beta pruning
- Claude Elwood Shannon
- Computer chess
- Expectiminimax
- Horizon effect
- Minimax Condorcet
- Monte Carlo tree search
- Minimax regret
- Negamax
- Negascout
- Sion's minimax theorem
- Transposition table

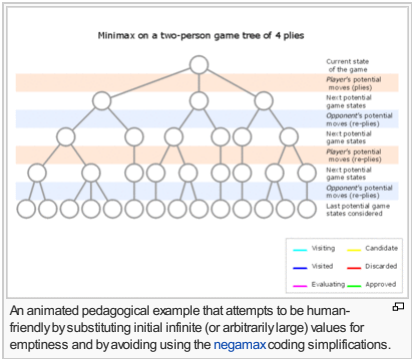
Notes  [edit]

- ↑ Provincial Healthcare Index 2013  (Bacchus Barua, Fraser Institute, January 2013 -see page 25-)
- ↑ *a b*, p. 126, ISBN 9789650614683  *Missing or empty |title= (help)*
- ↑ Osborne, Martin J., and **Ariel Rubinstein**. *A Course in Game Theory*. Cambridge, MA: MIT, 1994. Print.
- ↑ Von Neumann, J: *Zur Theorie der Gesellschaftsspiele* Math. Annalen. **100** (1928) 295-320
- ↑ John L Casti (1996). *Five golden rules: great theories of 20th-century mathematics – and why they matter*. New York: Wiley-Interscience. p. 19. ISBN 0-471-00261-5.
- ↑ Russell, Stuart J.; Norvig, Peter (2003), *Artificial Intelligence: A Modern Approach*  (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, pp. 163–171, ISBN 0-13-790395-2
- ↑ Hsu, Feng-hsiung (1999), "IBMs Deep Blue Chess Grandmaster Chips", *IEEE Micro* (Los Alamitos, CA, USA: IEEE Computer Society) **19** (2): 70–81, doi:10.1109/40.755469, "During the 1997 match, the software search extended the search to about 40 plies along the forcing lines, even though the nonextended search reached only about 12 plies."
- ↑ Arrow, "Some Ordinalist-Utilitarian Notes on Rawls's Theory of Justice, *Journal of Philosophy* 70, 9 (May 1973), pp. 245-263.
- ↑ Harsanyi, "Can the Maximin Principle Serve as a Basis for Morality? a Critique of John Rawls's Theory, *American Political Science Review* 69, 2 (June 1975), pp. 594-606.

External links  [edit]

- Hazewinkel, Michiel, ed. (2001), "Minimax principle", *Encyclopedia of Mathematics*, Springer, ISBN 978-1-55608-010-4
- A visualization applet
- "Maximin principle" from A Dictionary of Philosophical Terms and Names.
- Play a betting-and-bluffing game against a mixed minimax strategy
- [1] The Dictionary of Algorithms and Data Structures entry for minimax.
- Minimax (with or without alpha-beta pruning) algorithm visualization — game tree solving (Java Applet), for balance or off-balance trees.
- CLISP minimax game.  (in Spanish)
- Minimax Tutorial with a Numerical Solution Platform
- Java implementation used in a Checkers Game

<div><span>v</span> <span>t</span> <span>e</span></div>	<b>Topics in game theory</b> <span><span>[</span>hide<span>]</span></span>
<b>Definitions</b>	Normal-form game · Extensive-form game · Graphical game · Cooperative game · Succinct game · Information set · Hierarchy of beliefs · Preference
<b>Equilibrium concepts</b>	Nash equilibrium · Subgame perfection · Mertens-stable equilibrium · Bayesian-Nash · Perfect Bayesian · Trembling hand · Proper equilibrium · Epsilon-equilibrium · Correlated equilibrium · Sequential equilibrium · Quasi-perfect equilibrium · Evolutionarily stable strategy · Risk dominance · Core · Shapley value · Pareto efficiency · Quantal response equilibrium · Self-confirming equilibrium · Strong Nash equilibrium · Markov perfect equilibrium
<b>Strategies</b>	Dominant strategies · Pure strategy · Mixed strategy · Tit for tat · Grim trigger · Collusion · Backward induction · Forward induction · Markov strategy
<b>Classes of games</b>	Symmetric game · Perfect information · Simultaneous game · Sequential game · Repeated game · Signaling game · Cheap talk · Zero-sum game · Mechanism design · Bargaining problem · Stochastic game · <i>n</i> -player game · Large Poisson game · Nontransitive game · Global games · Strictly determined game
<b>Games</b>	Prisoner's dilemma · Traveler's dilemma · Coordination game · Chicken · Centipede game · Volunteer's dilemma · Dollar auction · Battle of the sexes · Stag hunt · Matching pennies · Ultimatum game · Rock-paper-scissors · Pirate game · Dictator game · Public goods game · Blotto games · War of attrition · El Farol Bar problem · Fair division · Fair cake-cutting · Cournot game · Deadlock · Diner's dilemma · Guess 2/3 of the average · Kuhn poker · Nash bargaining game · Screening game · Prisoners and hats puzzle · Trust game · Princess and monster game · Monty Hall problem · Rendezvous problem
<b>Theorems</b>	<b>Minimax theorem</b> · Nash's theorem · Purification theorem · Folk theorem · Revelation principle · Arrow's impossibility theorem
<b>Key figures</b>	Kenneth Arrow · Robert Aumann · Kenneth Binmore · Samuel Bowles · Melvin Dresher · Merrill M. Flood · Drew Fudenberg · Donald B. Gillies · John Harsanyi · Leonid Hurwicz · David K. Levine · Daniel Kahneman · David M. Kreps · Harold W. Kuhn · Eric Maskin · Jean-François Mertens · Paul Milgrom · Oskar Morgenstern · Hervé Moulin · Roger Myerson · John Nash · John von Neumann · Ariel Rubinstein · Thomas Schelling · Reinhard Selten · Herbert Simon · Lloyd Shapley · John Maynard Smith · Jean Tirole · Albert W. Tucker · Amos Tversky · William Vickrey · Robert B. Wilson · Peyton Young
<b>See also</b>	Tragedy of the commons · Tyranny of small decisions · All-pay auction · List of games in game theory · Confrontation analysis · List of game theorists · Combinatorial game theory · Alpha-beta pruning · Bertrand paradox · Bounded rationality · No-win situation · Coopetition



This page was last modified on 31 August 2015, at 03:55.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

