

@TQT_UCSD



AI Time Machine Triton Quantitative Trading Game Theory





Game Theory!

The study of strategic decision-making where the outcome for each participant depends on the choices of others.

Key Concepts:

- **Players:** Decision-makers in the game.
- **Strategies:** Plans of action players may take.
- **Payoffs:** Outcomes resulting from the combination of strategies.

Quant Example:

- Two companies deciding on pricing strategies: if both lower prices, profits decrease; if both keep prices high, profits remain stable; if one lowers prices while the other doesn't, the one with lower prices gains market share.



Game Theory in Quant Finance?

In quantitative finance, game theory helps model and predict behaviors in competitive markets.

- **Trading Strategies:** Anticipating competitor moves
- **Market Making:** Setting bid-ask spreads considering potential trades
- **Auction Models:** Bidding strategies in IPOs or bond auctions

Understanding these strategic interactions is crucial for developing robust financial models.

Brain teasers



Brain Teaser #1:



Matching Pennies

Problem:

Two players simultaneously choose Heads or Tails. If both choices match, Player A wins; if not, Player B wins.

Question:

What strategy should each player adopt to maximize their chances of winning?

		Player #2	
		H (r)	T ($1-r$)
Player #1 P	H	-1, 1	1, -1
	T ($1-p$)	1, -1	-1, 1

[ANSWER] Brain Teaser #1:



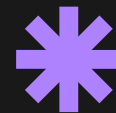
Answer:

Nash Equilibrium: Each player should randomly choose Heads or Tails with equal probability (50%). This mixed strategy ensures that neither player can predict the others move, leading to a fair game where the expected payoff is zero.

Explanation:

If one player becomes predictable, the other can exploit this pattern. Randomizing choices prevents exploitation.

Brain Teaser #2:



Two accomplices are arrested and interrogated separately. Each has the option to confess or remain silent. The outcomes are:

- Both confess: 5 years in prison each.
- One confesses, the other doesn't: the confessor goes free, the silent accomplice gets 10 years.
- Both remain silent: 1 year in prison each.

Outcome	Henry Cooperates	Henry Defects
Elizabeth Cooperates	(1,1)	(5,0)
Elizabeth Defects	(0,5)	(3,3)

Penalties for (Elizabeth, Henry)



Prisoners Dilemma

['pri-zən-ərs de-'le-mə]

A paradox in decision analysis in which two individuals acting in their own self-interests do not produce the optimal outcome.

 Investopedia

Question:

What is the rational choice for each prisoner?

[ANSWER] Brain Teaser #2:



Answer

Paradox: Rational self-interest leads both prisoners to confess, resulting in 5 years each. However, if both remained silent, they'd only get 1 year each.

Explanation:

This dilemma illustrates how individual rational decisions can lead to worse outcomes for all parties involved.

MinMax Algo.



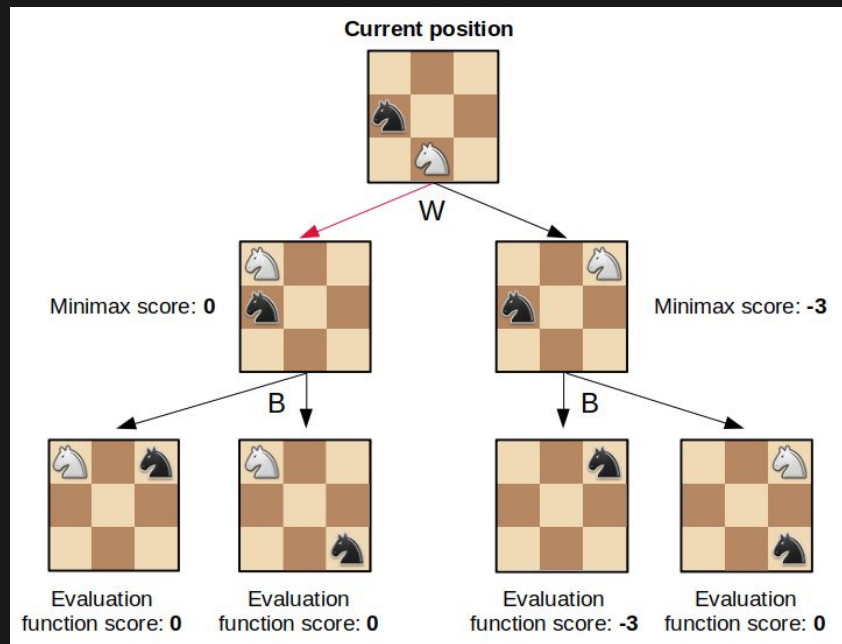
What Is a Minimax Bot?



An AI agent that uses the Minimax algorithm to make optimal decisions in competitive environments, usually in 2 person, turn based games

Great for:

- **Foundational Knowledge:** Understanding decision trees, recursion backtracking, & game state evaluation.
- **Practical Application:** Building AI that can play games like Tic-Tac-Toe or Chess.



Minimax Tic-Tac-Toe Bot



Steps:

1. **Game Representation:** Model the Tic-Tac-Toe board and rules.
2. **Recursive Evaluation:** Implement the Minimax algorithm to evaluate possible moves.
3. **Optimal Move Selection:** Choose the move with the best evaluated outcome.

Base Cases: Check for win, loss, or draw.

Recursive Cases: Simulate all possible moves and evaluate outcomes.

Optimization: Use pruning techniques to improve efficiency.

Result:

An AI that plays Tic-Tac-Toe optimally, never losing a game.

Pseudocode



Let's Build!

```
function minimax(position, depth, maximizingPlayer)
  if depth == 0 or game over in position
    return static evaluation of position

  if maximizingPlayer
    maxEval = -infinity
    for each child of position
      eval = minimax(child, depth - 1, false)
      maxEval = max(maxEval, eval)
    return maxEval

  else
    minEval = +infinity
    for each child of position
      eval = minimax(child, depth - 1, true)
      minEval = min(minEval, eval)
    return minEval
```

Optimized MinMax Algo

```
=====
[INFO] AI's Turn

AI Evaluation: 6.5
AI Best Move: e7e6
r . b q k b h r
p p p p . p p p
. . n . p . .
. . . . .
. . . . .
P P P . . P P P
R N R O K R N R
```



Alpha Beta Pruning



An optimized version of Minimax that “prunes” branches which cannot affect the final decision.

- Minimax is like checking every single possibility in a game before deciding.
 - Computationally expensive: **time complexity = $O(b^d)$**
- Alpha-Beta Pruning is like being smart and skipping paths you already know are bad because something better already exists.
 - More efficient: worst case $O(b^d)$, **best case $O(b^{d/2})$**

Where:

b = branching factor (number of moves per node aka number of legal moves)

d = depth of the tree

Alpha Beta Pruning



Very minor tweaks for alpha beta pruning:

- Add **Alpha** Parameter
- Add **Beta** Parameter
- Condition Checks + Update Min/Max values

```
function minimax(position, depth, alpha, beta, maximizingPlayer)
  if depth == 0 or game over in position
    return static evaluation of position

  if maximizingPlayer
    maxEval = -infinity
    for each child of position
      eval = minimax(child, depth - 1, alpha, beta, false)
      maxEval = max(maxEval, eval)
      alpha = max(alpha, eval)
      if beta <= alpha
        break
    return maxEval

  else
    minEval = +infinity
    for each child of position
      eval = minimax(child, depth - 1, alpha, beta, true)
      minEval = min(minEval, eval)
      beta = min(beta, eval)
      if beta <= alpha
        break
    return minEval
```

Quant: Alpha-Beta Pruning VS Minimax *

Portfolio Planning

- **Minimax**: Evaluate all outcomes for all portfolios.
- **Alpha Beta Pruning**: Skip poor-performing portfolios early.

Trading Bots

- **Minimax**: Simulate every market condition
- **Alpha Beta Pruning**: Ignore unprofitable strategies quickly.

Risk Models

- **Minimax**: Full scenario analysis
- **Alpha Beta Pruning**: Prune low-risk impact paths early.

PLEASEPLEASEPLEASEPLEASE





Thank You!

- Post Order Traversal -> Minimax -> Alpha Beta Pruning

Website to Study for Quant Interview questions (Including Brainteasers)

OpenQuant.co

<https://openquant.co/questions>